وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR UNIVERSITY -ANNABA
UNIVERSITÉ BADJI MOKHTAR-ANNABA

جامعة باجي مختار-
عنابـــــة

Faculté : Technologie

Département : Informatique

THESE

Présentée en vue de l'obtention du diplôme de

Doctorat 3ème Cycle

Intitulé

Classification des Ressources Partagées dans un Système de Cloud basé sur un Système P2P

Domaine : Mathématiques et Informatique

Filière: Informatique.

Spécialité: STIC (**S**ciences et **T**echnologies de l'**I**nformation et de **C**ommunication).

Par

Mlle. Abir Achache

Devant le jury

Tolba Cherif	Université Badji Mokhtar-Annaba	Pr	Président
Toufik Sari	Université Badji Mokhtar-Annaba	Pr	Directeur de Thèse
Abdelhalim Baaziz	Université Badji Mokhtar-Annaba	MCB	Co-directeur
Azizi Nabiha	Université Badji Mokhtar-Annaba	Pr	Examinateur
Bey Anis	École supérieure des sciences de gestion- Annaba.	MCA	Examinateur

2022 / 2023

Remerciements

Tout d'abord, je tiens à remercier, le bon dieu, le tout puissant et miséricordieux, qui m'a donné la force, la foi et la patience d'accomplir ce travail.

En second lieu, je tiens à exprimer ma profonde gratitude à mon directeur de thèse ***Pr. Toufik Sari*** *et mon co-directeur de thèse* ***Dr. Abdelhalim Baaziz*** *pour leurs conseils judicieux et leur encadrement qualifié qui m'ont permis d'améliorer grandement la qualité de ce mémoire. Qu'elles trouvent ici l'expression de mon très grand respect et du plaisir que j'ai travaillé avec eux.*

J'exprime également toute ma reconnaissance au ***Pr. Cherif Tolba*** *pour avoir aimablement présidé le jury de cette thèse. En outre, je tiens à présenter mes sincères remerceiments au* ***Pr. Nabiha Azizi*** *et au* ***Dr. Anis Bey*** *de l'honneur qu'ils m'ont accordé de particier au jury de ma thèse.*

Par ailleurs, je présente mes remerciements sincères à tous les membres de laboratoire ***LabGED****. De même, je remercie vivement mes enseignants qui par leurs compétences et savoir-faire nous ont soutenu pendant toute la période de mes études avec patience et abnégation.*

Je ne pourrais pas oublier mes amis, je les remercie tous, pour leur écoute, leur disponibilité et surtout leur gentillesse ainsi que les bons moments passés ensemble. Je leur souhaite une bonne continuation.

Dédicace

Je dédie cette thèse

A MA TRES CHERE FAMILLE

Je remercie très sincèrement : Mes Grands-parents, Mes Parents, Mes Frères, Ma seul Sœur, Mon Neveu, Ma Niece, Mes Oncles, Mes Tantes, Mes Cousins et Mes Cousines, qui ont été une source constante de soutien et d'encouragement.

Je dédie cette thèse particulièrement

A MA TRES CHERE MERE

Source inépuisable de tendresse, de patience et de sacrifice. Ta prière et ta Bénédiction m'ont été d'un grand secours tout au long de ma vie.

Quoique je puisse dire et écrire, je ne pourrais exprimer ma grande affection et ma profonde reconnaissance. J'espère ne jamais te décevoir, ni trahir ta confiance et tes sacrifices.

Puisse Dieu tout puissant, te préserver et t'accorder santé, longue vie et Bonheur

Enfin, je veux me remercier...

Je veux me remercier d'avoir cru en moi,

Je veux me remercier de réaliser ce dur travail,

Je veux me remercier de n'avoir aucun jour de repos,

Je veux me remercier de m'avoir jamais abondonné,

Je veux me remercier d'avoir pu trouver la force de toujours me relever pour avancer,

Et je veux me remercier d'être simplement moi à tout moment.

ملخص

منذ ظهورها ، يُنظر إلى الحوسبة السحابية (بالانجليزية Cloud Computing) على أنها تقنية ثورية في قطاع تكنولوجيا المعلومات، والتي تمكنت من تغيير طريقة استخراج الحوسبة واستخدامها في بنية تحتية بعيدة تابعة لجهة خارجية. إنه الآن نموذج تقني عالمي للحوسبة الموجهة نحو الخدمة، حيث يتم تقديم موارد وحلول تكنولوجيا المعلومات كخدمات حسب الطلب. البرمجيات كخدمة (بالانجليزية Software as a Service أو SaaS) هو نموذج تقديم الخدمات السحابية الرائد، مما يمكّن مزودي الخدمة من تقديم برامج مستضافة ومطورة ومدارة للعملاء عبر الإنترنت. ومع ذلك ، فإن وجود العديد من الفرص التي توفرها السحابة ينطوي أيضًا على تكاليف كبيرة من حيث البنية التحتية. في هذا السياق، ظهر اتجاه التصميم الحالي للحوسبة السحابية، مما يسمح لخدمات تكنولوجيا المعلومات بالانتقال من خدمات الإنترنت التقليدية إلى الخدمات السحابية القائمة على تقنية نظير إلى نظير. أصبح اتحاد هذه التقنيات موضوعًا شاغلا وواحدًا من الكلمات الرائجة للجيل القادم في مجال الحوسبة. ومع ذلك، فإن السحابة القائمة على شبكة نظير إلى نظير هي بيئة واسعة النطاق وغير متجانسة وديناميكية للغاية، يعتمد أداؤها بشدة على قدرتها على الحفاظ على التوفر الدائم لخدمات SaaS التي أصبحت أحد العوامل الأكثر أهمية لنجاح الخدمات والتطبيقات المستندة إلى الإنترنت. تقدم هذه الرسالة مساهمتين تتناولان التحديات المتعلقة بالبنية التحتية وتوافر خدمات SaaS.

أولاً ، نقدم مبادرة لتوزيع نموذج SaaS استنادًا إلى شبكات نظير إلى نظير. الفكرة هي إنشاء سحابة بدون بنية تحتية لتقديم الخدمات للعملاء في شكل برنامج كخدمة. يتمثل الحل في استخدام شبكة هجينة من نظير إلى نظير كبنية أساسية للسحابة من أجل تزويد النظراء بخدمات SaaS التي يتم تقديمها في البداية بواسطة السحابة ولكن يتم استضافتها وتشغيلها بواسطة أقران في شبكة نظير إلى نظير. يمكن أن تكون هذه البنية حلاً موثوقًا به للمشاكل المتعلقة بالبنية التحتية.

ثانيا، نقوم بتمديد الحل الأول ونقترح نهجًا لتحسين توافر خدمات SaaS في النظام السحابي القائم على شبكة نظير إلى نظير. يعتمد النهج المقترح على آلية تصنيف هجينة جديدة غير خاضعة للإشراف تهدف إلى توفير بنية تحتية افتراضية ومثالية من أجل تنظيم أقران النظام في مجموعات متميزة ممثلة بالعقد الافتراضية التي تشكل معًا طبقة افتراضية. تسمح هذه الطبقة ليس فقط بتوزيع مقدمي الخدمات من الأقران ولكن أيضًا بتكوين مناطق مكثفة لكل خدمة ذات أهمية لمجموعة من الأقران المتجاورين، مما يحسن احتمالية توافر الخدمات في مناطق محددة. بالإضافة إلى ذلك، نقترح نموذجًا لقياس مدى توفر الخدمة يعتمد على استخدام الطبقة الافتراضية للنظام مع مراعاة الكيانات المختلفة على مستويات مختلفة. يبدو أن هذا النهج قادر على تلبية متطلبات جودة الخدمة من حيث التوافر والحفاظ على الأداء في مثل هذه البيئة الموزعة الديناميكية واسعة النطاق.

الكلمات الرئيسية : الحوسبة السحابية, نظير إلى نظير, البرمجيات كخدمة, خدمة, توافر, تصنيف غير خاضع للإشراف, عملية التجميع الهجين المتسلسل,C-Means ضبابي مع العتبة , التصنيف الهرمي التصاعدي, PeerSim.

Abstract

Since its emergence, Cloud Computing has been seen as a revolutionary technology in the IT sector that has successfully changed the way of abstracting and using computing via a remote third-party infrastructure. It has become a universal technology paradigm for service-oriented computing, in which IT resources and solutions are delivered as on-demand services. Software as a Service « SaaS » represents the leader of Cloud service delivery models, enabling service providers to deliver hosted, developed and managed software to customers over the Internet. The high potential opportunities offered by the Cloud also imply high costs in terms of infrastructure. In this context, a current trend in Cloud Computing conception has emerged, allowing IT services to move from traditional Internet services to Cloud services based on Peer-to-Peer technology. The union of these technologies is becoming a burning topic and one of the buzzwords of the next generation IT industry. However, a Cloud based on a Peer-to-Peer « P2P » network is a large-scale, heterogeneous and highly dynamic environment whose performance is highly dependent on its ability to maintain persistent availability of SaaS services which has become one of the most critical factors for the success of Internet-based services and applications. This dissertation proposes two contributions that address the infrastructure and availability challenges of SaaS services.

Firstly, we propose an initiation to distribute the SaaS model based on Peer-to-Peer networks. The idea is to create a Cloud without infrastructure for providing to clients services in Software as a Service form. The solution consists of using a hybrid Peer-to-Peer network as an infrastructure for our Cloud in order to provide SaaS services that are initially offered by the Cloud but hosted and executed by the Peer to Peer network peers. This architecture can be a reliable solution to infrastructure issues.

Secondly, we extend the first solution and propose an approach for improving SaaS service availability in order to meet service quality requirements and maintain performance in a P2P-Based Cloud Computing system. The proposed approach is mainly based on a new hybrid clustering mechanism that aims to provide a virtual and optimal infrastructure in order to organize the system peers into distinct clusters represented by virtual nodes forming together a virtual layer. This layer allows not only the distribution of peer providers but also the formation of condensed areas of each service of interest for a set of neighboring peers, which improve the availability probability of services in specific regions. In addition, we propose a service availability measurement model based on the use of the system's virtual layer taking into account different entities at different levels. This approach seems to be able to meet the Quality of Service « QoS »requirements in terms of availability and to maintain performance in such large-scale dynamic distributed environment.

Keywords: Cloud Computing, Peer-to-Peer, Software as a Service, Services, Availability, Clustering, Sequential Hybridization Clustering Process, Thresholded Fuzzy C-Means, Hierarchical Ascending Clustering, PeerSim.

Résumé

Depuis son émergence, le Cloud Computing est perçu comme une technologie révolutionnaire dans le secteur des technologies de l'information, qui a réussi à changer la façon dont l'informatique est abstraite et utilisée sur une infrastructure tierce distante. Il est désormais un paradigme technologique universel de l'informatique orientée services, dans lequel les ressources et les solutions informatiques sont délivrées sous forme de services à la demande. Le logiciel en tant que service (en anglais Software as a Service, abrégé SaaS) constitue le leader des modèles de prestation de services Cloud, permettant aux fournisseurs de services de fournir des logiciels hébergés, développés et gérés aux clients via l'Internet. Or, avoir autant d'opportunités offertes par le Cloud implique également des coûts importants en termes d'infrastructure. Dans ce contexte, une tendance de conception actuelle du Cloud Computing est apparue, permettant aux services informatiques de passer des services de l'Internet traditionnels aux services en Cloud basés sur la technologie Pair-à-Pair. L'union de ces technologies est en train de devenir un sujet brûlant et l'un des mots à la mode de l'industrie de la prochaine génération dans le domaine informatique. Cependant, un Cloud basé sur un réseau Pair-à-Pair est un environnement à grande échelle, hétérogène et hautement dynamique, dont la performance dépend fortement de sa capacité à maintenir une disponibilité permanente des services SaaS qui est devenue l'un des facteurs les plus critiques pour le succès des services et des applications basées sur l'Internet. Cette thèse présente deux contributions traitant des défis liés à l'infrastructure et à la disponibilité des services SaaS.

En premier lieu, nous proposons une initiation pour distribuer le modèle SaaS en s'appuyant sur les réseaux Pair-à-Pair. L'idée est de créer un Cloud sans infrastructure pour fournir aux clients des services sous forme de logiciel en tant que service. La solution consiste à utiliser un réseau Pair-à-Pair hybride comme infrastructure pour notre Cloud afin de fournir aux pairs des services SaaS qui sont initialement offerts par le Cloud mais hébergés et exécutés par des pairs du réseau Pair-à-Pair. Une telle architecture peut constituer une solution fiable aux problèmes liés à l'infrastructure.

Dans un second lieu, nous étendons la première solution et nous proposons une approche pour améliorer la disponibilité des services SaaS dans un système de Cloud basé sur un réseau Pair-à-Pair. L'approche proposée est basée sur un nouveau mécanisme de classification non supervisée hybride qui vise à fournir une infrastructure virtuelle et optimale afin d'organiser les pairs du système en clusters distincts représentés par des nœuds virtuels formant ensemble une couche virtuelle. Cette couche permet non seulement la distribution des pairs fournisseurs mais aussi la formation de zones condensées de chaque service d'intérêt pour un ensemble de pairs voisins, ce qui améliore la probabilité de disponibilité des services dans des régions spécifiques. En outre, nous proposons un modèle de mesure de la disponibilité des services qui se base sur l'utilisation de la couche virtuelle du système en tenant compte de différentes entités à différents niveaux. Cette approche semble capable de répondre aux exigences de la qualité de service en termes de disponibilité et de maintenir les performances dans un tel environnement distribué dynamique à grande échelle.

Mots Clés: Informatique en Nuage, Pair-à-Pair, Logiciel en tant que Service, Service, Disponibilité, Classification non Supervisée, Processus de Regroupement Hybride Séquentiel, C-Moyennes Floues à Seuil, Classification Ascendante Hierarchique, PeerSim.

Liste des Publications

Publications Internationales

- A.Achache, A.Baaziz and T.Sari, Hybrid Fuzzy Clustering to Improve Services Availability in P2P-based SaaS-Cloud, Multiagent and Grid Systems, vol. 17, no. 4, pp. 297-334, 2021.

Conférences Internationales

- A.Achache, A.Baaziz and T.Sari, A Hybrid P2P Network-Based Remote Treatment SaaS Cloud Computing, 5th International Conference on Green Computing and Engineering Technologies (ICGCET), Casablanca, Morocco, September 17-19, 2019.

- A.Achache, A.Baaziz, and T.Sari, The Impact of Data Mining and SaaS-Cloud Computing, 3rd International Conference on Data Science and Applications, Istanbul, Turkey, June 25 - 28, 2020.

Table des Matières

Remerciements i
Dédicace ii
ملخص iii
Abstract iv
Résumé v
Liste des Publications vi
Table des Matières vii
Liste des Figures xii
Liste des Tableaux xiv
Chapitre 1 : Introduction 1
1. Contexte et Problématique 1
2. Motivations et Objectifs 4
3. Contributions 5
4. Structure de la Thèse 6
Partie 1 : Etat de l'Art 8
Chapitre 2 : Réseau Pair-à-Pair 9
1. Introduction 10
2. Définition 10
3. Historique 11
4. Caractéristiques des Réseaux Pair-à-Pair 13
4.1. Décentralisation 13
4.2. Passage à l'échelle 14
4.3. Autonomie 14
4.4. Anonymat 14
4.5. Hétérogénéité 15
4.6. Dynamique 15
4.7. Auto-organisation 16
4.8. Transparence 16
4.9. Connectivité Ad Hoc 16
4.10. Partage de coût 17
5. Taxonomie des Réseaux Pair-à-Pair 17
5.1. Les Réseaux Pair-à-Pair Non Structurés 18

5.1.1. Les Réseaux Pair-à-Pair Centralisés 18
5.1.2. Les Réseaux Pair-à-Pair Décentralisés....20
5.1.3. Les Réseaux Pair-à-Pair Hybrides 21
5.2. Les Réseaux Pair-à-Pair Structurés....23
6. Application Pair-à-Pair....24
6.1. Le Calcul Distribué....24
6.2. Partage et Distribution du Contenu....25
6.3. Collaboration et Communication....26
6.4. Plateformes de Développement 26
7. Architecture Pair-à-Pair VS Client-Serveur 27
8. Défis des Réseaux Pair-à-Pair 29
8.1. Désabonnement 29
8.2. Free-Riding....29
8.3. Sécurité....30
8.4. Disponibilité 31
8.5. Découverte de Ressource 32
9. Conclusion....33
Chapitre 3 : Cloud Computing....34
1. Introduction 35
2. Historique....35
3. Définition 36
4. Technologies Connexes....37
4.1. Grid Computing....37
4.2. Informatique Utilitaire 38
4.3. Virtualisation 38
4.4. Architecture Orientée Services et Service Web....39
5. Composants Basiques de l'Architecture du Cloud Computing 41
5.1. Utilisateurs Finaux....41
5.2. API et Interfaces d'accès....41
5.3. Réseau....41
5.4. Centre de Données....42
5.5. Virtualisation....42
6. Caractéristiques du Cloud Computing 43
6.1. Libre Service à la Demande 43
6.2. Accès Large au Réseau 43
6.3. Multi-locataire....44

6.4. Mise en Commun des Ressources 44
6.5. Élasticité Rapide 44
6.6. Service Mesuré 45
6.7. Système de Tarification Utilitaire 45
7. Modèles du Service Cloud 45
7.1. Infrastructure en tant que Service 46
7.2. Plateforme en tant que Service 46
7.3. Logiciel en tant que Service 47
8. Modèles De Déploiement 48
8.1. Cloud Public 48
8.2. Cloud Privé 49
8.3. Cloud Communautaire 50
8.4. Cloud Hybride 51
9. Avantages du Cloud Computing 52
10. Défis de l'Adoption du Cloud Computing 54
10.1. Conception Centralisée 55
10.2. Sécurité et Confidentialité 57
10.3. Disponibilité 58
10.4. Interopérabilité et portabilité 59
10.5 Performance 60
11. Vers une Architecture Cloud Computing basé sur Pair-à-Pair 61
11.1. Présentation du Cloud Computing basé sur Pair-à-Pair 61
11.2. Pourquoi un Cloud Computing basé sur Pair-à-Pair 61
11.3. Cloud Computing basé sur Pair-à-Pair VS Volunteer Computing 62
12. Conclusion 63
Chapitre 4 : Disponibilité des Services dans le Cloud Computing 64
1. Introduction 65
2. Unités et Facteurs de Disponibilité 65
3. Définition de la Disponibilité 67
4. Taux de Disponibilité 69
5. Mesure de la Disponibilité du Service 69
6. Coût de l'Indisponibilité 71
7. Mécanismes de Disponibilité dans le Cloud Computing 72
7.1. Equilibrage de Charge 72
7.2. Redondance 73
7.3. Réplication des Données 74

7.4. Tolérance aux Pannes....75
8. Solutions pour la Disponibilité dans le Cloud Computing....76
9. Conclusion....81
Chapitre 5 : Clustering pour la Disponibilité dans le Cloud-SaaS....82
1. Introduction....83
2. Data Mining....83
2.1. Définition....83
2.2. Processus de Découverte de Connaissance dans une Base de Données....84
2.3. Stratégies et Techniques du Data Mining....86
2.4. Applications du Data Mining....88
3. Cloud Computing et Data Mining....88
3.1. Cloud Computing basé sur Data Mining....88
3.2. Data Mining basée sur le Cloud Computing....91
4. Clustering....93
4.1. Définition....93
4.2. Distance et Similarité....94
4.3. Approches de Clustering....96
4.4. Méthodes de Clustering....97
4.5. Techniques de Validation du Clustering....106
5. Clustering pour la Disponibilité dans le Cloud Computing....111
6. Conclusion....117
Partie 2 : Contribution....118
Chapitre 6: Un Système Cloud-SaaS Basé sur un Réseau P2P Hybride pour le Traitement des Services à Distance....119
1. Introduction....120
2. Solution Proposée....120
2.1. Modèle en Couches....121
2.1.1. Couche Cloud....122
2.1.2. Couche des Serveurs Cloud Partiels....122
2.1.3. Couche Clients....122
2.2. Description Détaillée....122
2.2.1. Inscription et Connexion....122
2.2.2. Hébergement des Services....123
2.2.3. Recherche d'un Service....124
2.2.4. Invocation de Service....126
3. Evaluation et Analyse des Résultats....127

3.1. Environnement Expérimental 127
3.2. Résultat et Discussion 130
4. Conclusion 132
Chapitre 7: Clustering Hybride pour Améliorer la Disponibilité des Services dans un Cloud-SaaS basé sur P2P 133
1. Introduction 134
2. Approche Proposée 134
2.1. Rappel du Modèle de Système 134
2.2. Méthodologie 135
2.2.1. Préparation des Données 137
2.2.2. Formulation de la couche virtuelle 138
2.2.3. Mesure de la Disponibilité 143
3. Evaluation et Analyse des Résultats 147
3.1. Montage Expérimental 147
3.2. Critères d'Evaluation 149
3.3. Résultats et Discussion 151
4. Discussion Comparative 158
5. Conclusion 160
Conclusion 163
Bibliographie 166

Liste des Figures

Figure 2. 1: Protocoles Pair-à-Pair Populaire. 13
Figure 2. 2: Taxonomie des Réseaux Pair-à-Pair....18
Figure 2. 3: Architecture Pair-à-Pair Centralisé. 19
Figure 2. 4: Architecture Pair-à-Pair Décentralisé. 20
Figure 2. 5: Architecture Pair-à-Pair Hybride. 22
Figure 2. 6: Taxonomie des applications Pair-à-Pair (Milogicic et al. 2002)....24
Figure 2. 7: Modèle Client-Serveur VS modèle Pair-à-Pair....28
Figure 3. 1: Modèles des Services Cloud. 45
Figure 3. 2: Cloud Public. 49
Figure 3. 3: Cloud Privé. 50
Figure 3. 4: Cloud Communautaire. 51
Figure 3. 5: Cloud Hybride. 52
Figure 4. 1: Les Principales Causes des Interruptions non Planifiées. 66
Figure 4. 2: Les Principales Causes des Interruptions Planifiées. 67
Figure 5. 1: Processus de Découverte de Connaissances dans les Bases de Données. 85
Figure 5. 2: Objectifs et Stratégies du Processus d'Exploration des Données. 86
Figure 5. 3: Étapes du Processus d'Exploration de Données SaaS. 90
Figure 5. 4: Les Niveaux des Services du DMC....92
Figure 6. 1: Modèle en Couches du Système Proposé. 121
Figure 6. 2: Architecture Globale du Système. 121
Figure 6. 3: Etape d'Hébergement. 123
Figure 6. 4: Procédure d'Hébergement. 124
Figure 6. 5: Etape de Recherche. 125
Figure 6. 6: Procédure de Recherche. 125
Figure 6. 7: Etape d'Invocation. 127
Figure 6. 8: Procédure d'Invocation. 127
Figure 6. 9: Nombre de Nœuds Ressources. 130
Figure 6. 10: Recherche Réussie Vs. Recherche Echouée. 130
Figure 6. 11: Résultat Réussi Vs. Résultat Echoué. 131
Figure 6. 12: Pourcentage du Résultat Réussi Vs. Pourcentage du Résultat Echoué. 132
Figure 7. 1: Schéma Global de l'Approche Proposée. 136

Figure 7. 2: Matrice de Vecteurs des Profils des Clients. 137
Figure 7. 3: Algorithme de Classification Ascendante Hiérarchique avec le critère de Ward. 139
Figure 7. 4: Algorithme Thresholded Fuzzy C-Means modifié. 141
Figure 7. 5: Cycle de Disponibilité. 147
Figure 7. 6: Valeurs Moyennes des Indices de Validité Interne du Mécanisme de Clustering Hybride Proposé, de Fuzzy C-Means et de la Classification Ascendante Hiérarchique. 151
Figure 7. 7: Pourcentage de Recherche Réussie Vs Pourcentage de Recherche Echouée. 153
Figure 7. 8: Pourcentage de Résultat Réussi Vs Pourcentage de Résultat Echoué. 155
Figure 7. 9: Pourcentage de la Disponibilité Moyenne des Services. 156

Liste des Tableaux

Tableau 4.1: Taux de Disponibilité et Temps d'Arrêt du Service par (GR2841).....69

Tableau 4. 2: Résumé des principaux articles sur l'amélioration de la disponibilité dans le Cloud via des mécanismes de réplication, de planification et de tolérance aux pannes. 79

Tableau 5. 1: Aperçu des principaux travaux fondés sur les techniques de Clustering pour améliorer la disponibilité dans le Cloud Computing. 113

Tableau 5. 2: Comparaison des algorithmes de Data Mining implémentés dans SaaS.....116

Tableau 6. 1: Paramètres de Configuration. 128

Tableau 7. 1: Paramètres de Configuration. 148

Tableau 7. 2: Validation du Fuzzy C-Means, Hierarchical Ascending Classification, mécanisme de Clustering Hybride Semi-Flou Hierarchical Ascending Classification-Thersholded Fuzzy C-Means, appliquée à l'ensemble de données des profils des clients. 152

Tableau 7. 3: Valeurs moyennes, minimales et maximales des courbes de la figure 7.9.....157

Chapitre 1 : Introduction

Sommaire

1. Contexte et Problématique
2. Motivations et Objectifs
3. Contributions
4. Structure de la thèse

1. Contexte et Problématique

Au cours de la dernière décennie, nous avons assisté à une explosion de l'utilisation de l'Internet pour les applications distribuées, associée au développement des technologies de l'information. C'est une sorte de révolution qui devient de plus en plus importante et surtout plus rapide. Cette évolution rapide se traduit par l'émergence de plusieurs technologies parmi lesquelles le Cloud Computing. Ce dernier a été considéré comme le pilier de la future génération de l'Internet et l'un des concepts les plus révolutionnaires des technologies de l'information du 21ème siècle qui a concrétisé la promesse de l'informatique utilitaire (Syed et Baig 2013) (Zhang et al. 2010). Il est devenu un sujet de recherche majeur et une technologie indispensable qui pénètre de plus en plus dans tous les domaines informatiques commerciaux et scientifiques.

Le Cloud Computing est un paradigme informatique permettant un accès illimité à des ressources matérielles et logicielles (y a compris les réseaux, les serveurs, le stockage, les applications et les données) partagées via Internet. Les ressources informatiques peuvent être approvisionnées, libérées et consommées à la demande et de manière dynamique par les utilisateurs (Jo et Han 2018) (Kaur 2015) (Sandanayake et Jayangani 2018) (Zhang et al. 2010). Par ailleurs, le Cloud Computing offre des modèles commerciaux aux utilisateurs qui peuvent bénéficier de l'infrastructure fournie en adoptant une approche orientée vers les services. Ceux-ci sont proposés selon trois modèles: infrastructure en tant que service, plateforme en tant que service et logiciel en tant que service. Par conséquent, le Cloud Computing a favorisé l'émergence de nouveaux services et a fondamentalement changé la façon dont les ces derniers sont construits, fournis et utilisés, entraînant une évolution rapide du marché (Syed et Baig 2013) (Zhang et al. 2010).

Le logiciel en tant que service (en anglais Software as a Service, abrégé SaaS) est considéré comme un modèle de service informatique prometteur et leader des modèles de service Cloud. Il consiste à fournir des applications logicielles hébergées, développées et générées aux utilisateurs via Internet (Kaur 2015) (Sandanayake et Jayangani 2018). Ces services sont fournis via un paiement à l'utilisation ou gratuitement, et sont accessibles de n'importe où dès qu'il y a une connexion Internet, ce qui permet aux utilisateurs de se débarrasser des contraintes d'installation et d'exécution locale des applications (Syed et Baig 2013) (Zhang et al. 2010). Selon un rapport récent, la société américaine de conseil et de recherche dans le domaine des technologies avancées « Gartner » estime que les services SaaS bénéficieront d'une croissance significative des revenus et que les fournisseurs de logiciels passeront du « Cloud first » au « Cloud only ». Cela signifie que « la consommation de logiciels sous licence continuera à baisser, tandis que les modèles de consommation des services Cloud et du SaaS via le paiement à l'utilisation continueront à croître ». Les services Cloud ont gagné en popularité en raison

de leurs avantages fournis, tels que la flexibilité, l'évolutivité, le déploiement rapide, l'économie de coûts, la mobilité, la productivité, etc. Avoir autant d'opportunités fournies par le Cloud coûte aussi beaucoup en termes d'infrastructure. La majorité des systèmes basés sur cette technologie s'appuient sur une infrastructure structurée centralisée. Ils sont hébergés dans de grands centres de données et gérés de manière centralisée. En revanche, un point de défaillance unique pour le Cloud est représenté par les centres de données qui peuvent être facilement affectés par des phénomènes et des facteurs humains, techniques, climatiques et environnementaux, ce qui cause leur instabilité, voire leur blocage carrément. De plus, des goulots d'étranglement et une congestion du réseau peuvent survenir lorsque les demandes de service augmentent, ce qui entraîne une charge énorme sur les serveurs du Cloud (Zhang et al. 2010). Par ailleurs, les centres de données basés sur le Cloud, appartenant généralement à des organisations importantes comme Google et Microsoft, sont coûteux à construire. En outre, une forte consommation énergétique est nécessaire pour un fonctionnement adéquat des ressources matérielles constituant ces centres de données. Ceci implique des dépenses élevées et engendre, de surcroit, des conséquences négatives sur l'environnement car il en résulte la production de gaz à effet de serre qui contribuent à ce problème environnemental. L'emplacement géographique des Clouds centralisés est un autre inconvénient. Cet emplacement peut être le meilleur pour les propriétaires mais non pour les clients. Par conséquent, des recherches récentes ont étudié une stratégie différente de conception du Cloud Computing sans installation centralisée, en utilisant les technologies Pair-à-Pair (Kisembe et Jeberson 2017). Les systèmes Pair-à-Pair (en anglais Peer-To-Peer, abrégé P2P) sont apparus comme un modèle alternatif attrayant et prometteur, offrant une amélioration significative de la conception des systèmes distribués à grande échelle et de l'évolution des architectures Internet. Les systèmes Pair-à-Pair sont des systèmes distribués, composés d'un ensemble de nœuds hétérogènes, autonomes, dynamiques et interconnectés, capables de s'auto-organiser en topologies de réseau afin de partager une partie de leurs propres ressources physiques telles que l'espace disque, le processeur réseau ou la bande passante, ainsi que des ressources logiques telles que des services ou différentes formes de connaissances. Les participants aux réseaux P2P jouent un rôle similaire, chaque nœud pouvant agir à la fois comme serveur et comme client et étant directement accessible par les autres nœuds sans passer par des entités intermédiaires. En fonction du degré de centralisation, les réseaux P2P peuvent être divisés en : 1) réseau P2P non structuré centralisé 2) réseau P2P non structuré hybride, 3) réseau P2P non structuré décentralisé (purs), 4) réseau P2P structuré avec une table de hachage distribuée (DHT) (Chaudhary et Surolia, 2015) (Filali et al. 2011) (Maurya et al. 2016) (Pourebrahimi et al. 2005) (Steinmetz et Wehrle 2005).

L'union de ces technologies est en train de devenir à la mode dans le domaine de l'industrie. En effet, l'utilisation de ces services a gagné en popularité en raison de leur mobilité, faible coût, évolutivité, flexibilité, etc. Comme toute technologie qui nécessite une évolution constante, elle doit porter une attention particulière à ses différents défis. Dans le cadre d'une telle architecture Cloud Computing basée sur le Pair-à-Pair (en anglais Cloud Computing based on Peer-To-Peer, abrégé CC-P2P), la capacité du système à exécuter sa tâche dépend de la fiabilité de l'environnement P2P, qui dépend de l'architecture physique et logique du réseau et des nœuds composant le système. En d'autres termes, l'exécution des tâches est attribuée aux nœuds en fonction des besoins des clients. Par conséquent, la conception de ces environnements doit tenir compte des caractéristiques physiques des pairs sur lesquels le système fonctionne, notamment la vitesse du processeur, la taille de la mémoire, la capacité du disque, etc., ainsi que de leurs caractéristiques logiques qui contrôlent leurs comportements dynamiques, tels que l'inactivité, l'inertie, le désabonnement, le free-riding, etc. En effet, les performances du système sont influencées par ces caractéristiques. Toute raison de défaillance matérielle ou logicielle rend les pairs du réseau inactifs et inaccessibles pour répondre aux besoins des clients, empêchant les fournisseurs de services de fournir correctement les services demandés. Les

fournisseurs de services Cloud basés sur le P2P doivent atteindre des niveaux de service identiques, voire meilleurs, que les services Cloud traditionnels afin de garantir la qualité de leurs services, de maintenir la fidélité des clients et de maximiser leurs gains financiers durables (Ahuja et Mani 2012) (Dhote et Deshpande 2016). L'importance de satisfaire une disponibilité cohérente et permanente des services est récemment devenue l'un des facteurs les plus critiques pour le succès des applications et des services basés sur l'Internet, surtout s'agissant des services fournis par le biais des environnements CC-P2P. Traditionnellement, la disponibilité était limitée aux installations locales de ressources matérielles et logicielles que les entreprises et les consommateurs déployaient et entretenaient. Avec les progrès des services Cloud, on assiste à une migration considérable de ces ressources vers le Cloud où leur disponibilité fait référence à la probabilité de disponibilité d'un système, d'un réseau, du matériel et des logiciels qui fournissent collectivement ces services pendant leur utilisation. En outre, la disponibilité des services dans un système CC-P2P est définie comme étant la propriété d'être accessible et fonctionnel à la demande d'une entité autorisée et reflète la capacité du système à répondre aux demandes des clients, qui est également liée à la probabilité de la disponibilité temporelle et fonctionnelle des pairs fournisseurs (Ahuja et Mani 2012) (Karimi et al. 2008). Par conséquent, le fait d'assurer la disponibilité du service dans de tels systèmes orientés services est plus qu'une simple garantie de la disponibilité du serveur. Il s'agit également de s'assurer que l'infrastructure de communication entre les fournisseurs et les clients est capable d'assurer de manière flexible et efficace les fonctionnalités requises des services, leur disponibilité et d'autres caractéristiques qui peuvent être couvertes par le terme de qualité de service en nuage (en anglais Quality of Cloud Service, abrégé QoCS), notamment le temps de réponse, les performances, la vitesse, l'évolutivité et la confiance. Bien que diverses techniques aient été développées pour assurer l'accessibilité et améliorer la disponibilité des pairs et des services, elles empêchent les fournisseurs de services d'assurer un haut niveau de disponibilité et de continuité opérationnelle des systèmes de prestation de services à grande échelle sans affecter leurs performances. Lesdits fournisseurs souffrent toujours de problèmes liés à leur fonctionnement de base, qui repose principalement sur la nécessité d'une connaissance globale des réseaux et de leurs exploitations complexes stockées et calculées pendant leur traitement. Outre le volume de données dynamiques analysées dans des environnements distribués à grande échelle, ces techniques produisent généralement d'énormes quantités de données circulant dans le réseau, ce qui entraîne une augmentation inutile des efforts et des coûts en termes de gestion complexe et de stockage cohérent des données. De plus, les délais de traitement et d'exécution dans les grands réseaux, l'augmentation du temps d'inactivité des pairs et la limitation de la bande passante du réseau créent des goulots d'étranglement qui sont très sensibles à l'accès simultané aux données dans l'environnement distribué (Wei et al. 2010). Dans la mesure où la disponibilité durable, l'accessibilité et la gestion efficace des données et des services dans les environnements distribués ont une importance vitale pour les clients et les fournisseurs de services, la question de l'amélioration de la disponibilité des services dans les systèmes à concentration logicielle doit être réexaminée. On attend des fournisseurs de services qu'ils fournissent des infrastructures beaucoup plus fiables et robustes tout en assurant un environnement hautement évolutif, des communications performantes, une disponibilité adéquate des services et des capacités de fourniture de services afin d'améliorer l'expérience de leurs clients qui ne cessent de croître. Par conséquent, l'adoption de technologies matures et puissantes dans ces environnements est devenue nécessaire pour analyser les défis actuels, y répondre efficacement et maintenir des performances convenables.

2. Motivations et Objectifs

Un « terrain d'entente » intéressant a vu le jour, combinant les avantages des technologies du Cloud Computing et du Pair-à-Pair. Il est rapidement devenu le sujet le plus brûlant dans le domaine de l'informatique. Une architecture Cloud basée sur un système Pair-à-Pair est constituée principalement de la participation volontaire de pairs capables de s'auto-assembler et de s'autogérer afin de fournir une infrastructure Cloud distribuée et évolutive. L'infrastructure résultante permet l'accès et l'approvisionnement des services à des coûts réduits ou nuls en exploitant les pairs autant que possible. Cela signifie que les pairs, en installant les logiciels appropriés sur leurs machines locales, peuvent participer volontairement au CC-P2P et se fournir mutuellement des ressources en cas de besoin. Ceci contribue à réduire la latence du réseau, notamment pour les applications interactives. En diminuant les coûts d'utilisation et de déploiement, le CC-P2P offre, de plus, les avantages des commodités informatiques du Cloud Computing combinés à la flexibilité et l'évolutivité d'un système P2P. Cet état de fait en fait une solution viable pour l'informatique à très faible coût. Par conséquent, la philosophie avantageuse de ce mariage technologique a permis son adoption par les services informatiques utilisés actuellement, ce qui a été représenté par une vaste gamme de recherches, d'applications et d'études scientifiques. Néanmoins, tous les efforts existants portent essentiellement sur la proposition de diverses solutions de stockage dans les Clouds basées sur les réseaux P2P (Jo et Han 2018) (Kavalionak et Montresor 2012). De plus, bien que l'approche P2P soit utilisée avec succès depuis une vingtaine d'années, les recherches qui portent sur le développement de SaaS basés sur cette approche sont encore rares. D'autre part, même de nos jours, la plupart des systèmes P2P traite du simple partage de contenu, ce qui est essentiellement différent de la fonctionnalité de partage de services.

> En considérant les motivations énoncées ci-dessus, le premier objectif de notre travail est de développer une initiation dans le domaine de la recherche pour distribuer le modèle SaaS en se basant sur les réseaux P2P. En d'autres termes, nous avons pour objectif de proposer une solution Cloud qui utilise un réseau P2P hybride comme infrastructure permettant aux pairs d'utiliser et d'exécuter des services SaaS offerts hébergés sur d'autres pairs. En effet, un CC-P2P ne constitue pas une innovation en soi, mais l'idée de l'utiliser pour créer un réseau de fourniture de logiciels en tant que service pour le traitement à distance est une nouveauté à exploiter.

Le système P2P, en fournissant au Cloud Computing une infrastructure sous-jacente qui hérite de ses caractéristiques fondamentales, constitue un back-end pour lui. De ce fait, la valeur de l'infrastructure du Cloud résultante serait proportionnelle à la nature et au nombre des ressources et des pairs qui y contribuent. Le P2P doit relever un certain nombre de défis afin de devenir une solution d'infrastructure essentielle pour les applications Internet omniprésentes, surtout le Cloud Computing. C'est ce qui lui permettra de satisfaire, à la demande du Cloud Computing, les caractéristiques essentielles de l'approvisionnement en ressources. Dans cette optique, la garantie de la disponibilité des services à la demande est une exigence majeure qui doit être maintenue afin de construire un environnement plus complet et plus fiable.

Atteindre une disponibilité adéquate dans un environnement Cloud basé sur le P2P est une tâche complexe, principalement en raison des attentes élevées des utilisateurs finaux qui ne comprennent généralement pas le fait que cette exigence soit un défi pour les architectes de systèmes. Il est désormais indispensable de recourir à des technologies avancées et puissantes afin de pouvoir répondre efficacement à ces défis tout en maintenant un niveau de performance approprié. Dans ce contexte, l'exploration de données (en anglais Data Mining, abrégé DM) est apparue comme un domaine informatique relativement nouveau et interdisciplinaire, consistant à découvrir de nouveaux

modèles potentiellement valides et utiles à partir d'une énorme quantité de données en utilisant plusieurs méthodes à l'intersection de l'intelligence artificielle, de l'apprentissage automatique, des statistiques et des systèmes de bases de données qui ont été proposés et ont évolué au fil du temps. Elle permet d'extraire des informations structurées et des connaissances utiles à la prise de décisions à partir de sources de données web non structurées ou semi-structurées afin d'identifier les tendances, d'établir des relations entre les données et de prédire le comportement futur, ce qui contribue à réduire les coûts de stockage personnel et d'infrastructure ainsi qu'à fournir des services efficaces et sécurisés à leurs utilisateurs (Bandela et al. 2015) (Syed et Baig 2013) (Vani et Priya 2015) (Zhang et al. 2010). La classification non supervisée (en anglais Clustering) est considéré comme l'une des catégories de Data Mining les plus puissantes et les plus utilisées. L'intégration de ses techniques dans de tels environnements est devenue très importante. Il est devenu évident que l'avenir devrait être témoin de la puissance réelle de grandes quantités de données et de leur traitement efficace sur des plateformes distribuées par l'utilisation du Clustering.

Au regard de ces motivations, notre thèse vise en second lieu à étendre le premier objectif en envisageant une solution efficace pour résoudre le problème de disponibilité des services dans notre « système Cloud-SaaS basé sur un réseau Pair-à-Pair hybride pour le traitement des services à distance ». Nous visons à proposer et intégrer une approche basée sur des algorithmes de Clustering pour améliorer la disponibilité et garantir la persistance des pairs et des services qui les détiennent en surmontant les problèmes de défaillances du système et de dynamique des pairs.

3. Contributions

Les principales contributions de cette thèse sont résumées comme suit:

i. *Un système Cloud-SaaS basé sur un réseau P2P hybride pour le traitement des services à distance*

Nous avons proposé une nouvelle architecture de Cloud basée sur un réseau Pair-à-Pair hybride (en anglais Hybrid Peer-To-Peer, abrégé H-P2P) pour fournir aux clients des logiciels (pas nécessairement des applications web) en tant que service à distance en utilisant la technologie Remote Method Invocation (RMI). L'architecture proposée permet un Cloud Computing distribué où l'infrastructure du Cloud est construite sur un réseau H-P2P, permettant aux pairs d'utiliser et d'exécuter des services SaaS qui sont initialement fournis par le Cloud mais hébergés et exécutés par les pairs eux-mêmes. Le prototype réalisé et les résultats de la simulation (avec PeerSim) sont encourageants et prouvent qu'une architecture Cloud répartie et dynamique basée sur un réseau Pair-à-Pair peut constituer une solution fiable aux problèmes liés à l'infrastructure.

ii. *Clustering Hybride pour Améliorer la Disponibilité des Services dans un Cloud-SaaS basé sur P2P*

Nous avons proposé une approche pour améliorer la disponibilité des services SaaS afin de répondre aux exigences de qualité de service et de maintenir les performances dans un environnement dynamique « Cloud-SaaS basé sur un réseau Pair-à-Pair hybride pour le traitement des services à distance ». Notre approche est principalement basée sur un nouveau mécanisme hybride de classification non superviée qui vise à fournir une infrastructure virtuelle et optimale afin d'organiser les pairs du système en clusters distincts représentés par des nœuds virtuels formant ensemble une couche virtuelle. Cette couche permet non seulement la distribution des pairs utilisateurs et fournisseurs mais aussi la formation de zones condensées de chaque service d'intérêt pour un ensemble

de pairs voisins, ce qui améliore la probabilité de disponibilité des services dans des régions spécifiques. En outre, un modèle de mesure de la disponibilité des services a été proposé sur la base de l'utilisation de la couche virtuelle du système en tenant compte de diverses entités à différents niveaux. Pour obtenir les meilleures performances dans un système de fourniture de services, le modèle utilise différents types de disponibilitéqui existent dans la littérature (Dunn et al. 2005) (Karimi et al. 2008 (b)) (Netes 2018) et qui sont affinés pour inclure des considérations spéciales dans leurs calculs en raison de leur utilisation particulière. Dans cette contribution:

- Nous avons formé les données de profil des clients du système à partir d'un suivi et d'une analyse dynamique de leurs comportements et de leurs interactions entre eux et avec les services offerts par le système. Les données de profil collectées auprès de chaque client sont représentées sous forme de vecteurs de profil avec plusieurs attributs, l'ensemble de tous ces vecteurs générant une matrice de vecteurs de profil de client.
- Nous avons proposé un mécanisme hybride de classification non supervisé en combinant dans un processus séquentiel l'algorithme de classification ascendante hiérarchique avec l'algorithme C-Moyennes Floues à Seuil pour améliorer la qualité du regroupement automatique semi-flou des clients en fonction des similarités de leurs vecteurs de profil. Le mécanisme de clustering fournit une représentation efficace pour former la couche virtuelle du système, ce qui permet d'obtenir une infrastructure virtuelle optimale qui contribue grandement à améliorer la disponibilité du service.
- Nous avons considéré la couche virtuelle comme étant le noyau pour estimer la disponibilité des services du système et des pairs qui les détiennent. Compte tenu de l'absence de méthodes formelles pour mesurer la disponibilité des services dans les réseaux P2P sur lesquels l'architecture du Cloud est construite, plusieurs métriques de disponibilité sont redéfinies et reformulées pour être utilisées à différents niveaux dans le système de fourniture des services du Cloud-SaaS basé sur le P2P.
- Nous avons mis en œuvre un cadre expérimental de notre approche et l'avons adapté au simulateur PeerSim. Un certain nombre d'expériences préliminaires est mené pour explorer la performance du mécanisme de clustering proposé sur les données de profil. En outre, les performances de l'approche proposée sont évaluées à l'aide d'un ensemble de simulations comparées à celles réalisées sur le système de base proposé dans cette thèse en se basant sur divers critères de performance, notamment la disponibilité moyenne du service. Les résultats expérimentaux montrent que l'approche proposée améliore la probabilité de disponibilité des services SaaS et la fiabilité du système de CC-P2P. Elle répond principalement à la nature à grande échelle des systèmes distribués et permet de faire le meilleur compromis pour maintenir la qualité de service en termes de disponibilité, de performance et de coût.

4. Structure de la Thèse

Après avoir présenté, dans ce premier chapitre introductif, le contexte et les questions de recherche auxquelles cette thèse tente de répondre, leurs motivations et objectifs ainsi que les principales contributions, cette thèse s'articule autour des 7 chapitres suivants:

Le chapitre 2 représente un état de l'art sur les réseaux Pair-à-Pair. Ces derniers seront définis. Un aperçu historique ayant permis leur apparition sera donné, de même que leurs caractéristiques et que leurs architectures fondamentales. Ensuite, nous comparons l'architecture Pair-à-Pair avec l'architecture Client-Serveur. A la fin de ce chapitre, nous donnons une vue d'ensemble des domaines d'application contemporains des réseaux Pair-à-Pair. Nous faisons également part de leurs principaux problèmes et défis.

Le chapitre 3 apporte une étude approfondie sur le Cloud Computing. Il englobe, pour se faire, une vue d'ensemble de son historique, sa définition et les technologies connexes auxquelles il se compare. Suite à quoi, nous présentons les composants fondamentaux de l'architecture du Cloud Computing, ses caractéristiques, ses modèles de service et ses modèles de déploiement. Ensuite, nous montrons les avantages du Cloud Computing et ses défis majeurs en termes de son adoption. Nous mettons en lumière, enfin, l'architecture de Cloud basée sur les réseaux Pair-à-Pair et à laquelle nous nous intéressons dans cette thèse.

Le chapitre 4 est consacré à un état de l'art concernant la disponibilité des services dans le Cloud. Nous y présentons les principales unités et facteurs d'interruption des services Cloud. Nous introduisons, ensuite, plusieurs définitions ainsi que les taux standards de disponibilité des services dans les environnements Cloud Computing. Sur un autre volet, nous expliquons les mesures et les méthodes fondamentales pour le calcul la disponibilité des services, suite à quoi nous abordons les coûts de leur indisponibilité. Puis, nous donnons un aperçu des principaux mécanismes proposés dans la littérature et inhérents à l'amélioration de la disponibilité, et ce, en synthétisant leurs avantages et leurs faiblesses notamment. Enfin, nous présenterons les principales solutions et travaux de recherche permettant de faire face au problème de la disponibilité dans le Cloud Computing.

Le chapitre 5 est dédié à une étude sur les avancées de l'utilisation du Data Mining, principalement sa célèbre technique «Clustering», en particulier le modèle SaaS dans le Cloud Computing. D'abord, nous faisons un tour d'horizon des concepts de base du Data Mining. Ensuite, nous expliquons la manière dont le Data Mining et le Cloud Computing peuvent collaborer. Dans cette sillage, nous élaborons une vue globale du processus d'intégration du Data Mining dans le modèle SaaS du Cloud Computing pour améliorer la qualité de ses services. Deuxièmement, nous énumérons les concepts fondamentaux du Clustering, en abordant ce volet par sa définition et ses principales mesures de distance et de similarité. Puis, nous abordons les approches et les algorithmes de clustering adoptés dans le cadre de notre travail de recherche, suite à quoi nous discutons les métriques d'évaluation les plus répandues dans ce domaine. Pour clôturer ce chapitre, nous présentons des solutions et travaux de recherche importants concernant l'utilisation des algorithmes de Clustering pour la disponibilité des services dans le Cloud Computing, particulièrement dans le modèle SaaS.

Le chapitre 6 englobe la première contribution de cette thèse, représentée par un « Système Cloud-SaaS basé sur un réseau Pair-à-Pair hybride pour le traitement de services à distance ». Nous l'entamons par une description compréhensible et détaillée du système proposé, dont son architecture et son fonctionnement. A la fin du chapitre, nous décrivons le cadre expérimental et nous présentons les résultats de simulation.

Le chapitre 7 comporte la deuxième contribution de ce manuscrit, consistant en une approche basée sur un mécanisme de clustering hybride semi-flou. Ledit mécanisme a pour objectif l'amélioration de la disponibilité des services dans un « système Cloud-SaaS basé sur un réseau Pair-à-Pair hybride pour le traitement de services à distance », et ce, en palliant les problèmes de défaillances du système et de dynamique des pairs. Nous fournissons d'abord une description approfondie de l'approche proposée, ayant pour but principal l'amélioration de la disponibilité des services. Ensuite, nous décrivons le cadre expérimental et présentons les résultats obtenus. Enfin, nous présentons une étude dans laquelle nous comparons l'approche proposée aux principaux travaux existants.

Le chapitre 8 présente une conclusion sur les travaux réalisés dans cette thèse et donne une vue d'ensemble sur les perspectives et les orientations futures de la recherche.

Partie 1 : Etat de l'Art

Chapitre 2 : Réseau Pair-à-Pair

Sommaire

1. Introduction
2. Définition
3. Historique
4. Caractéristiques des Réseaux Pair-à-Pair
   - 4.1. Décentralisation
   - 4.2. Passage à l'échelle
   - 4.3. Anonymat
   - 4.4. Autonomie
   - 4.5. Dynamique
   - 4.6. Hétérogénéité
   - 4.7. Auto-Organisation
   - 4.8. Transparence
   - 4.9. Connectivité Ad Hoc
   - 4.10. Partage de Cout
5. Taxonomie des Réseaux Pair-à-Pair
   - 5.1. Les Réseaux Pair-à-Pair Non Structurés
     - 5.1.1. Les Réseaux Pair-à-Pair Centralisés
     - 5.1.2. Les Réseaux Pair-à-Pair Décentralisés
     - 5.1.3. Les Réseaux Pair-à-Pair Hybrides
   - 5.2. Les Réseaux Pair-à-Pair Structurés
6. Applications Pair-à-Pair
   - 6.1. Le Calcul Distribué
   - 6.2. Partage du Contenu
   - 6.3. Collaboration et Communication
   - 6.4. Plateformes de Développement
7. Architectures Pair-à-Pair Vs Client-Serveur
8. Défis des Réseaux Pair-à-Pair
   - 8.1. Désabonnement
   - 8.2. Free-Riding
   - 8.3. Sécurité
   - 8.4. Disponibilité
   - 8.5. Découverte de Ressource
9. Conclusion

1. Introduction

En tant que technologie jeune très émergente, le réseau Pair-à-Pair a suscité une grande attention dans le monde entier, allant des utilisateurs irréguliers de l'Internet aux entrepreneurs capitalisés. De nos jours, les innovations des réseaux Pair-à-Pair offrent également une approche de recherche très intéressante pour les communautés scientifiques. La technologie Pair-à-Pair se place comme une alternative au système Client-Serveur, offrant une architecture de réseau qui permet aux pairs participants de collaborer spontanément en utilisant des systèmes d'information et de communication appropriés afin de partager leurs propres ressources sans qu'une coordination centrale soit nécessaire. En adoptant une politique égalitaire, les pairs du réseau ont des responsabilités équivalentes de manière à ce que chacun d'entre eux peut agirà la fois comme client et serveur en fonction de ses besoins (Begredj et Madaoui 2013) (Kumar et SubbaRao 2019) (Schoder et al. 2005). En effet, les réseaux Pair-à-Pair sont désormais reconnus comme étant le modèle standard de distribution de l'information grâce à leurs propriétés avantageuses telles que la décentralisation, l'auto-organisation, l'évolutivité, la tolérance aux pannes, etc. Par conséquent, ils sont fortement sollicités dans la conception de grandes applications distribuées relevant de divers domaines, dont principalement le partage de contenu, le calcul distribué, la collaboration ainsi que les plateformes de développement.

Ce chapitre est consacré à l'étude des réseaux Pair-à-Pair. Tout d'abord, nous commençons par définir le concept de Pair-à-Pair, puis nous présentons l'historique qui a mené à son émergence. Ensuite, nous décrivons les caractéristiques des réseaux Pair-à-Pair ainsi que leurs diverses architectures. Nous abordons par ailleurs la différence entre l'architecture Pair-à-Pair et l'architecture Client-Serveur. À la fin, nous présentons une vue d'ensemble des domaines d'application des réseaux Pair-à-Pair ainsi que leurs principaux défis.

2. Définition

Le Pair-à-Pair (en anglais Peer-To-Peer, abrégé P2P) définit un modèle de réseau informatique qui offre une opportunité importante pour qu'un grand nombre (centaines de milliers ou millions) de nœuds coopèrent les uns avec les autres afin de partager leurs ressources via un réseau largement distribué (i.e. l'Internet). Étant un système de distribution coopératif, le réseau P2P permet l'agrégation et l'exploitation des ressources et des services distribués et fournis par les nœuds participants afin de remplir les fonctions essentielles du réseau de manière décentralisée. Au sein d'un tel réseau, les nœuds participants sont donc reliés directement les uns aux autres par des liens logiques, et ils partagent leurs ressources et bénéficient des ressources des autres nœuds du système (Bacache-Beauvallet et Cagé 2016) (Bruda et al. 2012).

Notre étude bibliographique a permis de relever plusieurs définitions des réseaux Pair-à-Pair, proposées principalement par la communauté scientifique. Pair-à-Pair désigne pour certains un type d'interconnexion, pour d'autres, il s'agit d'une technologie, mais tout le monde s'accorde à dire que le P2P permet le partage décentralisé des données (Juste et Axel 2014). Des tentatives pour décrire les réseaux P2P de façon plus étendue les définissent simplement comme étant l'opposé des architectures Client-Serveur (Singh 2001).

- **Définition 1** (Oram 2001)**:** Oram donne une définition de base du terme «Peer-to-Peer»:

  > *"a Peer-to-Peer system is a self-organizing system of equal, autonomous entities (peers) which aims for the shared usage of distributed resources in a networked environment avoiding central services."*

- **Définition 2** (Aberer 2002)**:**

  "Peer-to-peer computing or networking is a distributed application architecture that partitions tasks or workloads between peers. Peers are equally privileged, equipotent participants in the application".

- **Définition 3** (Nguyen 2011)**:**

  "The term peer-to-peer refers to a class of systems and applications that use distributed resources to perform a function in a decentralized manner".

En faisant réunir ces définitions, nous pouvons fournir une description plus complète de la technologie P2P. Le système P2P est un système informatique distribué, composé d'un ensemble de nœuds hétérogènes, autonomes, dynamiques et interconnectés, capables de s'auto-organiser en topologies de réseau afin de partager une partie de leurs propres ressources physiques telles que l'espace disque, le processeur réseau ou la bande passante ; ainsi que des ressources logiques telles que des services ou différentes formes de connaissances. Dans un tel système, toutes ces ressources partagées sont fournies uniquement par les pairs participants au réseau. Dans ce contexte, un pair est défini comme une entité constituée simplement en une application fonctionnant sur une machine, qui peut être un ordinateur personnel, un ordinateur de poche ou un téléphone mobile. Le concept innovant dans ce modèle de communication est l'attribution des rôles équivalentsaux pairs participantsqui agissent comme des « Servent » dans le réseau. Le mot artificiel « Servent » est issu de la combinaison de la première syllabe du terme serveur ("Serv-") et de la deuxième syllabe du terme client (*-ent") (Schollmeier 2001) (Steinmetz et Wehrle 2005). Ainsi, il représente la capacité des nœuds d'un réseau P2P d'agir à la fois comme des *clients* lorsqu'ils consomment des ressources disponibles, et comme des *serveurs* lorsqu'ils offrent des ressources, et est directement accessible par les autres nœuds sans passer par des entités intermédiaires. Le P2P modélise l'échange direct des ressources en faisant communiquer directement les pairs entre eux sans existence d'une autorité centralisée, ce qui rend sa conception particulièrement distincte des architectures traditionnelles.

3. Historique

- **Avant le World Wide Web**

La conception originale de l'Internet à la fin des années 1960 était basée sur un système Pair-à-Pair. Ce dernier, connu auparavant sous l'appellation de «hôte-à-hôte», était le premier à établir une communication d'égal à égal entre deux pairs. L'apparition de ce concept a commencé avec l'établissement de l'ARPAnet (Advanced Research Projects Agency Network). L'objectif principal de ce réseau était de partager des ressources informatiquesà travers différents centres de recherche américains. En adoptant une philosophie égalitaire loin de celle du Client-Serveur, les concepteurs cherchaient à interconnecter des réseaux universitaires existants et de les considérer comme égaux (les traiter de la même manière). Une décennie plus tard, un système en réseau P2P de forums, appelé «UseNet», a été inventé et lancé publiquement à l'université de Caroline du Nord à Chapel Hill et l'université Duke. UseNet a été le premier à fonctionner sans serveur central ni administrateur.Avant la fin des années 1980, l'utilisation de mesures de sécurité telles le pare-feu était presque inutile lors de la communication directe entre machines sur le réseau, car seuls les chercheurs collaboraient dans différents projets (Juste et Axel 2014) (Sungkar et Kogoya 2020).

- **La révolution des systèmes Pair-à-Pair**

Vers les années 1990, le réseau internet a été popularisé mondialement grâce au grand nombre d'applications développées, comme le WWW, la messagerie électronique et le streaming. Les particuliers se sont empressés de rejoindre la communauté Internet, ce qui a permis l'adhésion de

millions d'utilisateurs commerciaux et de clients. En effet, les internautes cherchent de plus en plus à utiliser leurs ordinateurs connectés pour des besoins qui dépassent le fait de demander temporairement du contenu à des serveurs Web ou de courrier électronique. Par conséquent, leschercheurs et les concepteurs de réseaux ont complété leurs systèmes qui exploitaient l'approche traditionnelle de partage et de distribution de ressources Client-Serveur par la nouvelle alternative, à savoir les réseaux P2P.

La vague de popularité du P2P a commencé dans en 1999 avec l'introduction de l'application Napster qui a radicalement changé les mécanismes de partage de fichiers. Napster a été développée par Shawn Fanning pour le partage et l'échange de fichiers multimédias. Les utilisateurs de Napster recherchaient des chansons ou des artistes par le biais d'un serveur centralisé de recherche/indexation qui indexait les titres sur le disque dur de chaque ordinateur connecté au réseau. Bien qu'il s'agisse d'un réseau P2P, la partie de recherche de ressources reste celle du client-serveur. Cette application permettait aux utilisateurs d'exploiter leurs ordinateurs non seulement pour consommer et télécharger du contenu, mais aussi pour offrir et livrer du contenu à d'autres utilisateurs participants sur Internet. Cependant, la présence d'une administration centralisée a représenté le point de défaillance unique dans le réseau Napster qui pouvait ainsi être facilement ciblé. De ce fait, Napster a été condamné et fermé en raison d'un procès intenté par de grandes maisons de disques portant sur des problèmes de droits d'auteur et de contenu illégal partagé sur le réseau (Schollmeier 2001) (Steinmetz et Wehrle 2005).

La fermeture de Napster n'a pas arrêté la croissance des applications P2P. En mars 2000, l'application Gnutella a été publiée en tant que projet open source par la société Nullsoft. Gnutella est un protocole informatique distribué pour la recherche et de transfert de fichiers Pair-à-Pair. Dans Gnutella, les pairs participants agissent à la fois comme des serveurs et des clients, et sont nommés des «Servents». Chacun d'entre eux est connecté à un ensemble de voisins. En éliminant toute entité centrale, les fonctionnalités d'échange et d'approvisionnement des fichiers ainsi que de la recherche et l'acheminement du contenu sont complètement distribuées. Au lieu de rechercher des ressources dans un serveur centralisé,le pair demandeur interroge tous ses voisins en leur envoyant un message de recherche. Les pairsvoisins font de même avec leurs propres voisins. Le message de recherche est associé avec un champ TTL (Time To Live) pour comptabiliser le nombre de retransmissions restantes. La recherche est arrêtée lorsque la ressource est trouvée ou le TTL atteint une valeur nulle. Si la recherche est aboutie, le pair demandeur et le pair fournisseur se mettent en connexion directe afin d'échanger la ressource voulue.Ce concept est connu sous le nom d'inondation de requête (en anglais, Flooding) (Stutzbach et Rejaie 2005). Via ce mécanisme, il est difficile de suivre le comportement des pairs et, par conséquent, prouver des activités illégales. Cependant, le processus d'inondation entraîne des frais généraux en termes de bande passante. Une variante de l'approche d'inondation consiste à choisir certains nœuds comme des super-nœudsqui sont caractérisés par une grande capacité et une haute disponibilité. Ces super-nœuds sont répartis sur le réseau et sont chargés de maintenir l'index des ressources dans leurs domaines respectifs et de répondre aux recherches de façon appropriée. De plus, les super-nœuds peuvent interroger d'autres super-nœuds et permettre à d'autres super-nœuds de les interroger à leur tour. Cette approche peut, dans une large mesure, éviter les problèmes de bande passante, mais elle ne peut pas éliminer complètement les problèmes inhérents aux inondations. Ces réseaux Peer-to-Peer avec une deuxième hiérarchie de routage dynamique sont appelés réseaux Peer-to-Peer de deuxième génération. En effet, la nature distribuée de Gnutella a été reprise par une communauté de développement et de recherche en plein essor. Elle a également conduit le spectacle vers d'autres protocoles populaires incluant : Direct Connect, KaZaA, iMesh, Freenet, BitTorrent et eDonkey ; comme le montre la figure 2.1. Avec l'émergence de ces réseaux, il est devenu plus facile de partager des fichiers à travers le monde (Jaideep et Battula 2016).

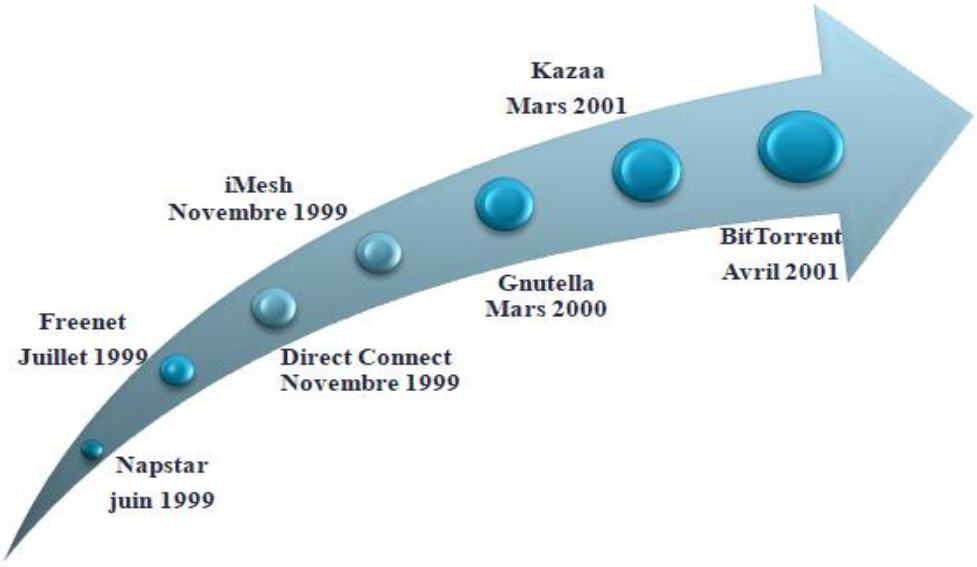


Figure 2. 1: Protocoles Pair-à-Pair Populaire.

La technologie P2P n'est pas seulement utilisée pour le partage de fichiers multimédia. Ce paradigme de mise en réseau a été adopté par la communauté des chercheurs en bioinformatique. Un service P2P appelé Chinook4 a été développé pour faciliter l'échange de techniques d'analyse. La technologie est également utilisée dans d'autres domaines, notamment les réseaux téléphoniques IP comme Skype et les réseaux de télévision comme PPLive. Dans un tel réseau, l'augmentation de la fiabilité est une préoccupation majeure qui nécessite l'adaptation des schémas de routage pour réduire le taux de réplication et garantir une communication adéquate entre les utilisateurs. Pour répondre à ces lacunes, les réseaux et protocoles Pair-à-Pair de troisième génération sont apparus. Ils se caractérisent principalement par l'utilisation d'un algorithme de routage proactif basé sur des tables de hachage distribuées (DHT). La DHT contient l'indexation des ressources et est située dans les pairs du réseau, ce qui aide à rechercher efficacement des ressources. Elle a été introduite dans des projets comme Tapisserie, Pâtisserie, Kademlia et Chord (Benoit 2006) (Jaideep et Battula 2016).

Bien que les systèmes Pair-à-Pair soient apparus très tôt en tant que technologie de réseau prometteuse, leur développement rapide pour devenir une application majeure de l'Internet est dû aux applications qui ont pu émerger selon ce nouveau modèle de réseau.

4. Caractéristiques des Réseaux Pair-à-Pair

Les systèmes P2P possèdent des caractéristiques avantageuses par rapport aux autres systèmes distribués. Pour cette raison, ils sont en plein essor depuis plusieurs années et leur popularité ne cesse de croître. Même s'il n'y a pas, à notre connaissance, un système P2P qui possède toutes les caractéristiques suivantes, chaque système P2P essaie de posséder un grand nombre de ces caractéristiques.

4.1. Décentralisation

Les réseaux P2P remettent en question le concept Client-Serveur dans lequel le stockage et le traitement des données ne se font que sur des serveurs centralisés et l'accès au contenu est assuré par des protocoles de demande-réponse. La conception égalitaire des réseaux P2P a reposé sur le principe de décentralisation permettant que les ressources et l'administration du réseau soient gérées localement sur chaque machine participante. Le comportement d'un tel réseau est déterminé

par les actions collectives des nœuds où chacun d'entre ces derniers est responsable de la gestion de ses propres ressources et où aucun pair ne possède une vue globale du réseau. L'architecture de ces réseaux peut fonctionner sans avoir besoin d'une administration centralisée, ce qui permet d'éviter les goulots d'étranglement et le point de défaillance unique, d'augmenter ainsi la résistance du système face aux pannes et aux défaillances (Begredj et Madaoui 2013) (Jaideep et Battula 2016)(Milogicic et al. 2002) (Nguyen 2011).

4.2. Passage à l'échelle

L'absence d'une autorité centraliséeforte se répercute immédiatement sur l'amélioration du passage à l'échelle des réseaux Pair-à-Pair. Les architectures P2P se présentent actuellement comme une solution viable pour permettre le passage à l'échelle de ses applications. En effet, la description du passage à l'échelle dans un tel système en réseau se fait par rapport à la taille du réseau, c'est-à-dire au nombre de nœuds et d'arcs du graphe constituant la topologie du réseau (Nguyen 2011). Jourjon et D. El Baz ont proposé une définition du principe de passage à l'échelle pour un système informatique basé sur un réseau Pair-à-Pair (Jourjon et El Baz 2005).

> *"The scalability of a P2P network designed for global computing is its capacity to maintain its efficiency when peers join or leave the system."*

Il s'agit d'une condition indispensable à l'exploitation de systèmes P2P qui doivent être en mesure de faire coopérer un grand nombre de nœuds (jusqu'à des milliers ou des millions) pour partager leurs ressources tout en maintenant une bonne performance des systèmes. L'évolutivité d'un tel système fait référence à sa capacité à continuer à bien fonctionner lors d'une croissance ou diminution imprévisible du nombre de nœuds participants afin de répondre aux besoins de ses utilisateurs. Les architectures P2P s'appuient sur des méthodes et des mécanismes bien adaptés avec un environnement dans lequel il y a un nombre important de messages à échanger entre un grand nombre de nœuds partageant un grand volume de ressources via un réseau largement distribué et dynamique (Al King 2010) (Begredj et Madaoui 2013) (Milogicic et al. 2002) (Nguyen 2011) (Tlili 2011).

4.3. Autonomie

Cette caractéristique assure le fait que la participation et le fonctionnement des pairs dans le réseau P2P sont déterminés localement et ne se reposent sur aucune autorité ou fournisseur de services centralisé. L'autonomie des nœuds leur permet d'être indépendants dans la gestion de leurs ressources d'une façon libre. Chaque nœud décide quand et quelle partie de ses ressources il faut partager avec d'autres entités dans le réseau P2P. Il peut également se connecter et/ou se déconnecter à n'importe quel moment (Al King 2010) (Bender 2007) (Jaideep et Battula 2016) (Schoder et al. 2005).

4.4. Anonymat

La notion d'anonymat est étroitement liée à celle de l'autonomie. Dans les réseaux centralisés, la garantie de l'anonymat est difficile en raison de la présence d'un serveur qui est généralement capable d'identifier les nœuds connectés et de dévoiler leurs ressources ainsi que leurs opérations effectuées dans le système. En revanche, la protection de la vie privée et de l'anonymat des utilisateurs est désormais un aspect indispensable au regard de l'évolution des technologies et des applications Internet, y compris les systèmes Pair-à-Pair. Dans les systèmes Pair-à-Pair, l'anonymat est défini comme étant le degré pour lequel le système tient compte des entités et des opérations non identifiées. Cela signifie que l'anonymat doit permettre aux pairs d'utiliser les services fournis

par les systèmessans révéler leur identité ni celle de tout ce qui est en relation avec leurs opérations exécutées. La communication entre deux pairs au sein du système Pair-à-Pair doit être en mesure d'assurer l'anonymat du pair expéditeur, l'anonymat du pair récepteur et l'anonymat mutuel qui cache l'identité de l'expéditeur et du récepteur ainsi que des autres pairs durant l'échange des messages. Par ailleurs, elle doit également préserver l'anonymat des ressources partagées dans le réseau et des requêtes initiées par les pairs qui, à priori, ne doivent pas être suivies. La technologie P2P vise à permettre aux gens d'utiliser des systèmes sans se soucier des ramifications légales ou autres, tout en garantissant que la censure des utilisateurs et du contenu ne soit pas possible (Begredj et Madaoui 2013) (Milogicic et al. 2002). Par conséquent, divers mécanismes d'anonymat ont émergé et ont été intégrés dans les applications P2P, notamment Freenet, FreeHaven et Publiusmettre, qui ont pour rôle principal d'assurer la persistance et la disponibilité dans un environnementsoumis à la censure. De plus, des infrastructures P2P ont vu le jour, telles que Crowds, Morphmix et Tarzan, dont le but consiste à fournir des services d'anonymat aux applications P2P (Doyen 2005).

4.5. Hétérogénéité

Le réseau P2P peut rassembler un ensemble de pairs qui ne sont pas nécessairement homogènes. Les nœuds participants au réseau peuvent posséder des caractéristiques différentes sur les plans : (1) matériel avec, à titred'exemple, des téléphones mobiles, des stations de travail ou un cluster de machines ; (2) logiciel au sein des langages de programmation et des systèmes d'exploitations ; et (3) de la communication avec l'utilisation de diverses technologies. En outre, la capacité d'autonomie des pairs favorise la disponibilité d'une collection de ressources hétérogènes dans le réseau (Jaideep et Battula 2016) (Kumar et SubbaRao 2019).

4.6. Dynamique

Le réseau P2P est constitué d'un ensemble de pairs qui participent volontairement en partageant une partie de leurs ressources. L'une des propriétés importantes dans un tel réseau est la participation dynamique des pairs, également appelée le désabonnement (churn) des pairs participants qui est une propriété inhérente aux réseaux P2P. Le désabonnement est défini comme un changement dans l'ensemble de nœuds constituant le réseau en raison des adhésions, des départs ou des défaillances de nœuds. En raison de leur autonomie, les nœuds peuvent rejoindre le réseau un par un ou le quitter librement et gracieusement à tout moment sans affecter son fonctionnement. Pour ce faire, ilsinforment leurs voisinspour restructurer la topologie du réseau et conserver les ressources. En revanche, un grand nombre de nœuds peut rejoindre le réseau et/ou le quitter silencieusement, simultanément et fréquemment. Cela revient d'abord au comportement des pairs participant au réseau P2P. En effet, ces pairs sont exécutés à des fins professionnelles ou personnelles sur des machines qui peuvent se connecter et se déconnecter de manière autonome et imprévisible. Par ailleurs, le fonctionnement de ces machines est susceptible d'être affecté par des défaillances matérielles ou logicielles qui rendent les pairs indisponibles dans le réseau. En outre, les pairs participants sont naturellement dispersés dans différentes zones géographiques. Ils peuvent parfois se trouver dans des emplacements physiques qui leur permettent de se connecter au réseau, et parfois non, notamment en raison de problèmes d'électricité et d'Internet (Doyen 2005) (Trifa et Khemakhem 2014). De ce fait, des ressources existantes peuvent disparaitre et de nouvellesressources peuvent être ajoutées auréseau. Par conséquent, la dynamique des pairs permet de créer un réseau hautement dynamique avec des topologies complexes, capables de gérer un grand nombre de ressources fortement variables (Jaideep et Battula 2016) (Kumar et SubbaRao 2019) (Tlili 2011) (Yeferny 2014).

4.7. Auto-organisation

Une autre caractéristique attrayante des systèmes P2P est la possibilité d'auto-organisation. Un système P2P est un environnement dynamique capable de reproduire sa structure en fonction des changements dynamiques et imprévisiblesde la topologie du réseau. Les changements peuvent concerner la constitution et le nombre de pairs ainsi que les relations au sein du réseau.Ils se traduisent par l'ajout, la suppression ou la modification d'un pair, de ses ressources et données ou de ses relations ou connexions avec d'autres pairs. En outre, la participation d'un nouveau nœud à un système P2P ne nécessite pas une infrastructure couteuse. De tels systèmes possèdent des pairs d'amorçage qui offrent aux nouveaux nœuds la possibilité de se joindre. Par conséquent, les systèmes P2P sont auto-organisés et composés d'un ensemble de nœuds qui est capable de se reconfigurer afin de réaliser une fonction globale de manière décentralisée, et ce, quels que soient les allers et retours des pairs et la disponibilité des ressources dans le réseau (Kumar et SubbaRao 2019) (Milogicic et al. 2002) (Steinmetz et Wehrle 2005) (Tlili 2011).

4.8. Transparence

Les systèmes Pair-à-Pair présentent la propriété de transparence qui s'applique à plusieurs points dans le réseau. D'après Jourjon (Jourjon et El Baz 2005):

> *"The transparency can be defined as the property to make undistinguished local or remote access to all parts of the task and data set needed for computation."*

La définition ci-dessus signifie que cette caractéristique permet à chaque pair connecté au réseaud'avoir accès à l'ensemble des composants nécessaire pour le calcul, quelque soit l'état du réseau. Ces composants incluent principalement les pairs et les ressources/données. Les systèmes Pair-à-Pair font une abstraction des différences de nature des pairs en termes d'hétérogénéité des caractéristiques (matériel, logiciel, communication) et leurs emplacements dans le réseau. Ceci revient à appliquer la transparence sur le routage effectué dans la mesure d'échange entre les pairs. Les voisins dans la topologie virtuelle ne sont pas forcément des voisins dans la topologie physique. Ils peuvent être situés dans des espaces physiques et sur des réseaux différents. En ce qui concerne les ressources/ données, les systèmes Pair-à-Pair rendentleur accès et leur localisation transparents. Lorsqu'un pair accède à une ressource, il ne sait pas si cette dernière est locale ou distante. Dans le cas où la ressource est distante, il n'a pas de connaissance dupair qui l'héberge (Nguyen 2011) (Tlili 2011). Au fil du temps, d'autres formes de transparence ont été incluses, telles que la transparence de la défaillance, la mobilité, la mise à l'échelle, etc (Milogicic et al. 2002).

4.9. Connectivité Ad Hoc

Le réseau ad hoc est composé de dispositifs individuels qui communiquent directement entre eux plutôt que de passer par un point d'accès centralisé tel qu'un routeur. La communication ad hoc permet au système de communiquer sans avoir besoin d'une infrastructure préexistante, à l'exception des ordinateurs qui forment le réseau. Le terme ad hoc sous-entend un environnement où les membres vont et viennent en fonction de leur emplacement physique ou de leurs intérêts du moment.

Les réseaux P2P présentent une nature ad hoc. Ils ont pour objectif de faire communiquer directement les différents pairs participants au sein des différents types de réseaux. Ces pairs peuvent s'organiser en groupes ad hoc pour communiquer, collaborer et partager leurs ressources entre eux afin de mener à bien les tâches à accomplir. Les réseaux P2P ne reposent pas sur une infrastructure établie, mais plutôt sur l'interconnexion d'un ensemble de nœuds hétérogènes qui

partagent volontairement leurs ressources (Begredj et Madaoui 2013) (Milogicic et al. 2002). En outre, la nature Ad Hoc des réseaux P2P est liée à leur dynamique. Les réseaux P2P sont caractérisés par une connectivité intermittente des pairs qui les composent, c'est-à-dire qu'ils peuvent rejoindre et quitter le réseau (de manière normale ou anormale) à tout moment, généralement de manière imprévisible.

4.10. Partage de coût

Les réseaux P2P sont une forme de réseau informatique distribué constitué d'un nombre de machines indépendantes qui contribuent aux ressources du système P2P pour assurer son fonctionnement en partageant volontairement une partie de leurs ressources. L'organisation dans un tel réseau repose sur l'ensemble des pairs sans recourir à des serveurs d'administration centralisés. Ces machinesreprésentent un fort potentiel en bordure de l'Internet, et peuvent être des ordinateurspersonnels, desordinateurs de poche, des téléphonesmobiles, etc., qui appartiennent à des propriétaires différents (par exemple des particuliers, des universités, des entreprises). Ces propriétaires sont responsables de tout ce qui est achat, mise en œuvre et maintenance. Par conséquent, les systèmes P2P permettent une utilisation maximale de la puissance du réseau, en éliminant ou en répartissant les coûts d'infrastructure sur tous les pairs. Ils réduisent considérablement les coûts d'acquisition des systèmes et des contenus ainsi que les coûts de maintenance. Outre l'aspect financier, cela permet aux fournisseurs de services de concentrer leurs actions sur l'aspect logiciel de l'infrastructure qu'ils offrent. De plus, les dépenses liées au fonctionnement des infrastructures et des systèmes de communication sont allégées grâce à l'utilisation des existantes et à la réduction des coûts d'administration et d'utilisation (Al King 2010) (Doyen 2005) (Milogicic et al. 2002) (Schoder et al. 2005) (Tlili 2011).

5. Taxonomie des Réseaux Pair-à-Pair

Les réseaux Pair-à-Pair constituent une architecture distribuée, dans laquelle les participants mettent une partie de leurs ressources informatiques à la disposition des autres membres du réseau et servent simultanément de serveurs et de clients. En effet, les échanges dans une telle architecture peuvent avoir lieu directement entre les pairs participants au sein d'un système global, stable et de taille variable. Bien que la décentralisation soit une caractéristique particulière des réseaux Pair-à-Pair, il peut subsister certaines unités centrales, permettant par exemple de localiser un contenu spécifique (Bacache-Beauvallet et Cagé 2016).

Actuellement, il existe de nombreux réseaux P2P dont chacun possède des propriétés différentes. Il n'existe pas de critère unique de leur catégorisation. En adoptant la classification présentée dans (Steinmetz et Wehrle 2005), les réseaux P2P peuvent être développés sous différentes architectures, à savoir : 1) non structurée centralisée, 2) non structurée décentralisée, 3) non structurée hybride et, 4) structurée. La figure 2.2 présente brièvement quelques points saillants de chaque architecture.

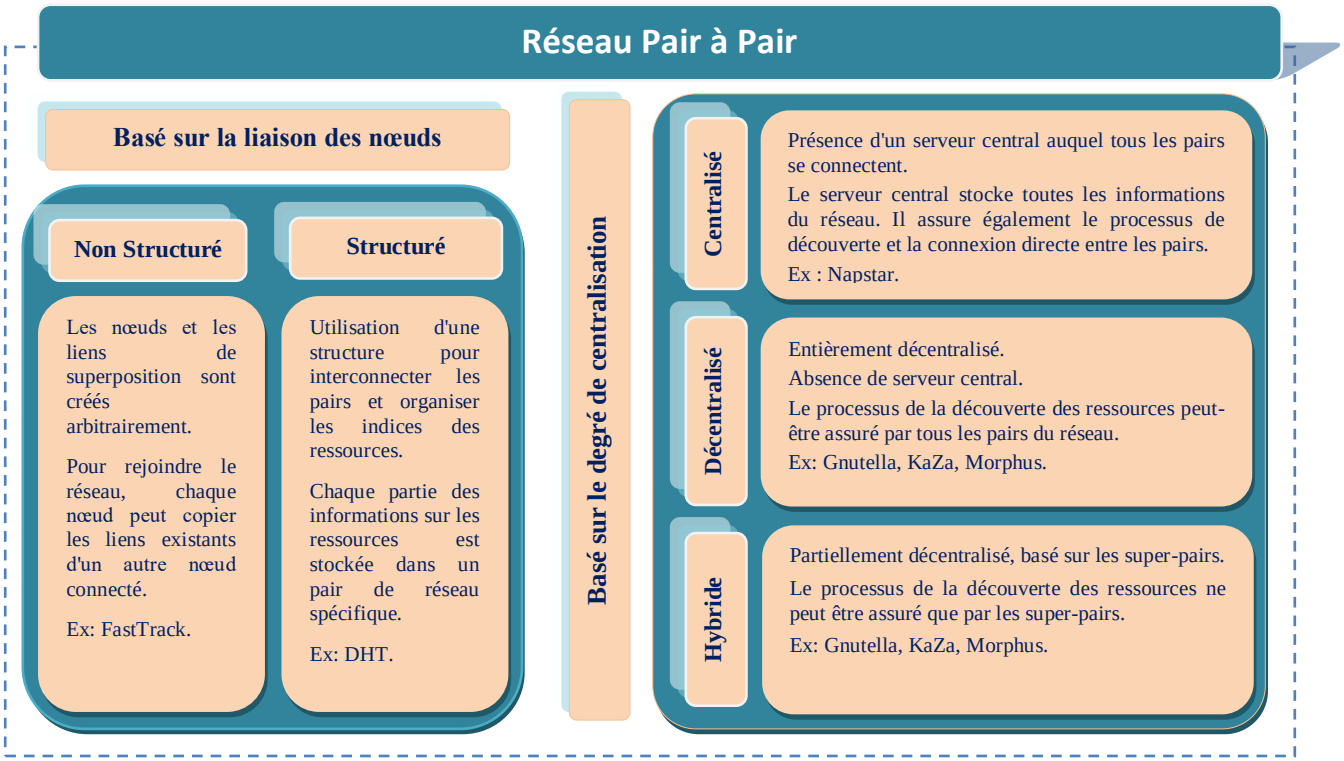


Figure 2. 2: Taxonomie des Réseaux Pair-à-Pair.

Dans les approches non structurées, les réseaux P2P sont établis de manière non-déterministe et sans aucune restriction sur le placement des ressources/données dans la topologie du réseau. Les réseaux P2P non structurés n'obéissent à aucune structure spécifique et le contenu stocké sur un nœud donné et son adresse IP ne sont pas liés. Dans de tels réseaux, les pairs se découvrent mutuellement et établissent des connexions entre eux arbitrairement, de manière plate ou hiérarchique (Steinmetz et Wehrle 2005). Par conséquent, chaque nœud a une vision limitée du réseau, ce qui fait qu'il peut connaître ses voisins dans le réseau sans être au courantdes données qu'ils stockent. Par ailleurs, la localisation des données n'est pas connue à priori lors des processus de recherche (Park et al. 2010) (Tlili 2011). En vue de trouver des pairs possédant une ressource particulière ou un ensemble de ressources désirées, les réseaux P2P non structurés s'appuient sur plusieurs techniques telles que l'inondation. Dépendamment du niveau de décentralisation, les réseaux informatiques Pair-à-Pair non structurés se scindent en trois modèles d'architecture, à savoir les systèmes P2P centralisés, les systèmes P2P décentralisés (ou purs) et les systèmes P2P hybrides.

5.1.1. Les Réseaux Pair-à-Pair Centralisés

La première génération du réseau Pair-à-Pair a commencé avec le concept de centralisation qui s'inspire fortement du modèle traditionnel client-serveur. Cette topologie de réseau se repose sur la disponibilité d'un serveur central qui se charge de gérer les ressources et les bases des données des pairs qui se connectent à lui (Peter 2002). Autrement dit, le serveur maintient les informations sur les pairs connectés (adresse IP, numéro du port, bande passante disponible, etc.) ainsi que les métadonnées décrivant les ressources partagées par les pairs du réseau (nom ressource, etc.).

Dans cette architecture, chaque pair du réseau maintient une connexion au serveur centrale auquel il peut émettre des demandes de ressource correspondant aux mots clés indiqués dans la demande. Chaque pair arrivant doit contacter le serveur pour l'informer de son adresse IP actuelle et le notifier activement des informations sur toutes les ressources qu'il est prêt à partager. Les informations

recueillies auprès des pairs seront ensuite utilisées par le serveur pour créer dynamiquement une base de données centralisée, qui associe les noms des ressources à une liste d'adresses IP sous la forme de pair <clé-ressource, adresse-nœud>. Le serveur central exploite cette base de données lors de la résolution des requêtes de recherche en identifiant les nœuds et stockant les ressources voulues. Si la requête peut être résolue par le serveur de recherche centralisé, elle renvoie les coordonnées d'accès des pairs (principalement des adresses IP et des ports) offrant la ressource qui est décrite avec les mêmes mots clés que ceux indiqués dans la demande (Guetmi 2016) (Steinmetz et Wehrle 2005). Après la connexion au serveur et la découverte du nœud qui possède la ressource voulue, la communication et l'échange (l'interaction de bout en bout) de données se fait de façon directe entre le pair demandeur et le pair fournisseur (Kurose et Ross 2003). Par conséquent, il convient de noter que le serveur ne conserve pas les ressources disponibles dans le réseau mais se charge uniquement de la gestion des partages, des recherches et des insertions d'informations. Le concept de base de ce type de réseau est montré dans la figure 2.3. Dans la figure, le pair D initie une demande de(s) ressource(s) (par exemple, des fichiers) au serveur central. Ce dernier trouve que le pair C peut répondre à la demande du pair D et envoie l'ID de nœud C au pair D. A ce stade, le pair D établit un lien de communication direct avec le pair C et obtient la ressource voulue.

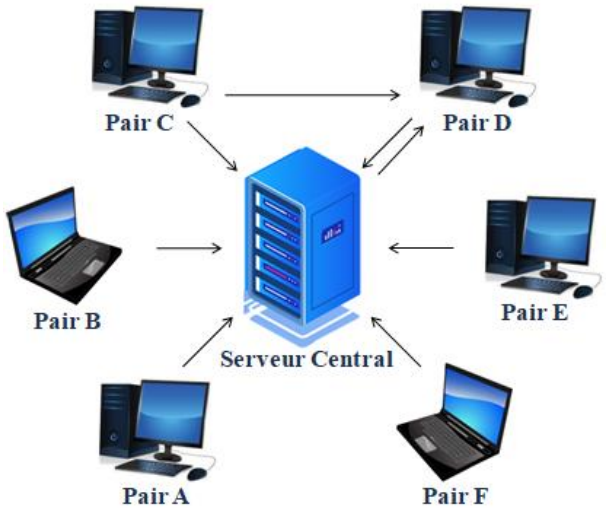


Figure 2. 3: Architecture Pair-à-Pair Centralisé.

Ce concept a été largement utilisé et est devenu particulièrement connu en raison de Napster, offrant des échanges gratuits de fichiers audio/vidéo (Steinmetz et Wehrle 2005). Par la suite, d'autres applications sont apparues, comme Audiogalaxy et WinMX.

Le grand intérêt de cette architecture réside essentiellement dans la simplicité et la facilité de son déploiement ainsi que dans le confort et l'efficacité de la découverte des informations. Elle permet de fournir des recherches complètes, rapides et garanties. Cependant, le principal inconvénient de cette conception est que le serveur central constitue un point de défaillance unique de tout le réseau, ce qui le rend susceptible de rencontrer certaines failles de sécurité et particulièrement vulnérable à la censure et aux attaques malveillantes. Par conséquent, la défaillance du serveur entraîne l'effondrement de l'ensemble du réseau et empêche son fonctionnement.En outre, le serveur représente un goulot d'étranglement qui affecte l'évolutivité de cette approche, en raison des limitations de la taille de la base de données et de sa capacité à répondre aux requêtes (Kumar et SubbaRao 2019) (Park et al. 2010) (Pourebrahimi et al. 2005) (Steinmetz et Wehrle 2005) (Wang et Li 2003).

5.1.2. Les Réseaux Pair-à-Pair Décentralisés

Un système P2P décentralisé « ou pur » correspond à un réseau superposé de nature plate où tous les pairs du réseau agissent sur un pied d'égalité.Ce modèle d'architecture ne dispose pas de serveur d'index centralisé pour gérer le réseau, ni de pairs privilégiés ayant une fonction d'infrastructure particulière. Les pairs d'un tel réseau remplissent tous les mêmes tâches et ont des responsabilités équivalentes faisant que chacun d'entre eux peut se comporter en tant que client et serveur en même temps (Guetmi 2016). Ils sont souvent appelés SERVENT (**SERV**eur+cli**ENT**) et sont caractérisés par un degré d'autonomie élevé. Dans ce type d'architecture, il n'y a pas d'hiérarchie et les nœuds ne sont pas obligés d'être structurés selon une forme géométrique prédéfinie. Pour rejoindre le réseau, chaque nœud établit une connexion directe avec un ou plusieurs nœuds appelés voisins, formant une topologie logique du P2P qui est souvent un maillage aléatoire et non structuré. Par ailleurs, chaque nœud maintient sa propre base de données et il n'est pas nécessaire qu'il ait connaissance des ressources disponibles chez ses voisins (Al King 2010). Les requêtes de recherche sont distribuées entre les pairs du réseau en se basant principalement sur un mécanisme d'inondation. Lorsqu'un pair s'intéresse à une ressource spécifique, il envoie une requête à ses voisins dans le réseau superposé, qui la retransmettent ensuite à tous leurs pairs voisins. Ce mécanisme de recherche permet aux requêtes d'être exécutées saut par saut à travers le réseau jusqu'à ce qu'elles réussissent, échouent ou atteignent le temps maximum accordé (Time-To-Live). En cas de succès, les pairs partageant la ressource souhaitée répondront au pair demandeur qui sélectionnera ensuite un pair parmi eux et initiera avec lui une session de communication et d'échange de la ressource désirée par le biais d'une connexion directe (Park et al. 2010). Gnutella, Freenet, Chord, et CAN sont des exemples de tels systèmes (Pourebrahimi et al. 2005). La figure 2.4 présente l'architecture purement décentralisée.

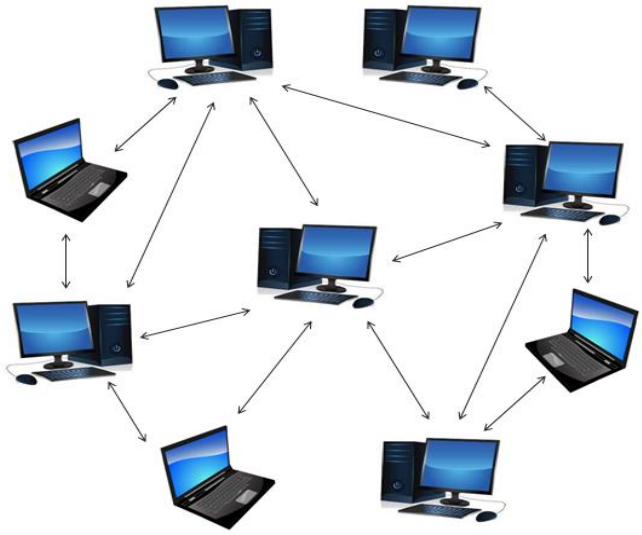


Figure 2. 4: Architecture Pair-à-Pair Décentralisé.

Les réseaux Pair-à-Pair purs ont permis l'introduction d'un concept de communication avantageux en éliminant la notion de la centralisation. Ce nouveau type de réseaux P2P suscite une nette préférence de la part des utilisateurs grâce au total de l'anonymat qu'il procure. En outre, il impose de très faibles exigences aux nœuds individuels et peut accueillir des nœuds de puissance variable qui sont très autonomes.De plus, l'absence de point central de défaillance fait que le réseau est intrinsèquement tolérant aux pannes, de sorte que la perte d'un pair, voire de plusieurs pairs, n'affecte pas les performances du système de manière significative. Cependant, les réseaux Pair-à-Pair décentralisés sont souvent critiqués pour leur évolutivité limitée. En effet, le nombre de messages générés pour la construction des liens de voisinage et pour la recherche des ressources peut croître linéairement avec la

taille du réseau. Cela entraine une surcharge du réseau et une augmentation du trafic réseau. En outre, les requêtes de recherches ne sont pas garanties dans de tels réseaux qui présentent une découverte de ressources lente, surtout si les ressources sont rares. De plus, le comportement du réseau est difficile à prévoir du fait de l'absence d'une vision globale au niveau du réseau. Par conséquent, il n'y a aucune garantie sur la qualité des services offerts dans les réseaux décentralisés (Park et al. 2010) (Pourebrahimi et al. 2005) (Wang et Li 2003).

5.1.3. Les Réseaux Pair-à-Pair Hybrides

Certains chercheurs affirment qu'en environnement P2P, l'hypothèse es rôles équivalents des nœuds constitue une forte contrainte. Dans la pratique, les nœuds d'un réseau P2P sont caractérisés par une forte disparité en termes de capacités de stockage et de traitement. Il est donc préférable d'attribuer aux nœuds des rôles qui conviennent à leurs capacités et à leurs puissances. Par conséquent, une architecture hybride issue de la combinaison du modèle P2P centralisé et du modèle P2Pdécentralisé a été proposée afin de tirer parti de leurs avantages. Les réseaux Pair-à-Pair hybrides sont caractérisés par l'introduction d'une autre couche hiérarchique dynamique. La particularité dans ces systèmes réside dans le fait que certains des pairs du réseau, appelés « Super-pairs » (ou Super-nœuds, en anglais Super-peer), assument un rôle plus important que le reste des pairs. Dans une telle topologie, les Super-pairs sont spécialement désignés en fonction de leurs espaces disque, leurs bandes passantes et leurs puissances de calcul élevées afin de pouvoir effectuer des fonctions utiles au système, telles que l'indexation, l'évaluation de requêtes, le contrôle d'accès, la gestion de métadonnées et le rôle de relais dans l'acheminement des requêtes. Lorsqu'un pair rejoint le réseau, il est attribué à un super-pair auquel il doit notifier les ressources qu'il va partager.Similairement à la conception centralisée, chaque Super-pair joue le rôle d'un serveur, mais uniquement pour les pairs ordinaires qui lui sont connectés. Autrement dit, chaque serveur présente un nœud central pour un cluster qui comporte un ensemble de pairs ordinaires organisés en P2P.Les Super-pairs maintiennent une base de données contenant les index pour les informations sur les ressources partagées par les pairs qui lui sont attribués, ce qui rend la recherche des ressources plus facile et efficace. Les requêtes de recherche initiées par les pairs du réseau sont donc envoyées/acheminées vers leurs super-pairs et non à d'autres pairs. De façon analogue à la conception décentralisée, les Super-pairs inondent les requêtes de recherche dans le réseau Super-pair superposé au nom de leurs pairs.Les super-pairs répondent par la suite aux pairs qui ont émis les demandes en leur fournissant des listes de pairs disposant des ressources désirées (Park et al. 2010) (Schollmeier2001). La figure 2.5 représente l'architecture Pair-à-Pair hybride « partiellement centralisée ».

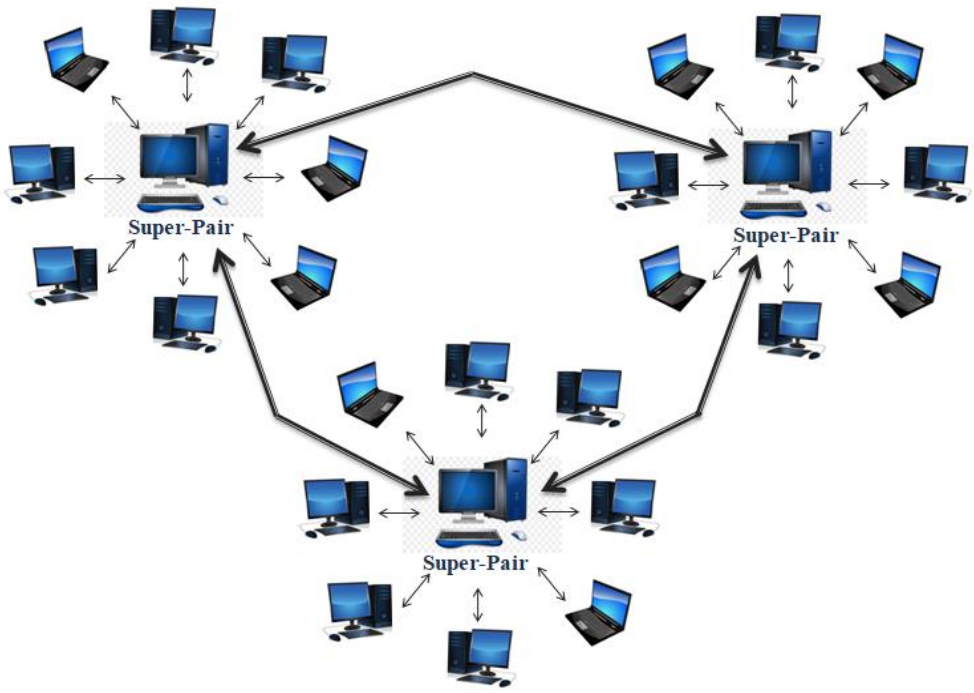


Figure 2. 5: Architecture Pair-à-Pair Hybride.

Les réseaux hybrides utilisent plusieurs super-pairs afin de faciliter la gestion des ressources partagées dans le système. En fonction de la désignation des super-pairs, nous distinguons deux types de réseaux hybrides, à savoir statique et dynamique. Dans les réseaux P2P hybrides statiques, les super-pairs sont manuellement désignés dès le début. Comparativement aux autres types de systèmes P2P, ceux-là sont moins tolérants aux pannes. Les clients d'un super-pair qui tombe en panne deviennent isolés du reste du système et privés de ses services.Suivant cette lacune, les réseaux hybrides dynamiques sont introduits dans les systèmes dans lesquelsles pairs ordinaires peuvent changer de rôle au fil du temps et devenir des super-pairs qui prennent part à la coordination de la structure du réseau P2P. Dans ces systèmes, l'élection de nouveau super-pair s'effectue automatiquement si les pairs disposent d'une disponibilité, d'une bande passante et d'une puissance de traitement suffisante. De ce fait, il est possible que les super-pairs soient dynamiquement remplacés en présence d'une panne. Le réseau continuera ainsi à fonctionner étant donné que les nœuds qui sont connectés à ces super-pairs peuvent ouvrir une nouvelle connexion avec d'autres. Les pairs existants deviennent eux-mêmes des super-pairs afind'éviter tout risque dans le cas où un grand nombre ou tous les super-pairs disparaissent (Pourebrahimi et al. 2005). Une telle architecture de réseau est utilisée par exemple par Gnutella 6, Morpheus et Kazaa.

Actuellement, les systèmes Pair-à-Pair hybrides offrent de meilleures performances par rapport aux réseaux purement décentralisés, car les fonctions du réseau ne sont plus dédiées à un seul serveur mais sont plutôt réparties entre les super-pairs. Par ailleurs, chaque super-pair maintient une base de données des pairs de son cluster relativement faible. Par conséquent, la structure des modèles P2P hybrides permet de réduire le nombre de connexions et la charge de travail de chaque serveur, et d'éviter ainsi les problèmes de surconsommation de bande passante. En outre, elle permet d'éliminer les points de défaillance unique et réduire le temps de découverte et le trafic de messages échangés entre les nœuds. En revanche, le déploiement et la gestion de ces réseaux sont nettement plus complexes et imposent une maintenance non-triviale du réseau. En outre, ces systèmes héritent des limites habituelles des réseaux centralisés. Le fait d'avoir des super-pairs ayant plus de responsabilités

que les pairs ordinaires, peut conduire à un goulot et entraîner des problèmes d'anonymat, de disponibilité et de passage à l'échelle (Park et al. 2010) (Pourebrahimi et al. 2005).

5.2. Les Réseaux Pair-à-Pair Structurés

Contrairement aux réseaux Pair-à-Pair non structurés, une architecture de réseau informatique Pair-à-Pair a été apparue ens'appuyant sur une politique de conception différente qui permet d'établir un lien entre le contenu stocké et l'adresse IP d'un nœud. Les pairs et les ressources d'un tel réseau sont organisés sous une forme géométrique prédéfinie (anneau, arbre, étoile, etc.) selon des critères et des algorithmes spécifiques. En outre, les pairs maintiennent des informations sur leurs voisins ainsi que sur les ressources (le contenu partagé) qui les possèdent. Cela conduit à des superpositions avec des topologies et des propriétés spécifiques appelées «Réseaux Pair-à-Pair structurés» (Al King 2010) (Steinmetz et Wehrle 2005).

Les réseaux Pair-à-Pair structurés sont généralement construits à l'aide des techniques de tables de hachage distribuées. Une table de hachage distribuée (en anglais Distributed Hash Table, abrégé DHT) gère les ressources en les répartissant sur les nœuds du réseau. Cela peut être réalisé en appliquant une fonction de hachage distribuée sur les ressources et les pairs du réseau. Dans de tels réseaux, chaque pair a une clé obtenue en appliquant une fonction de hachage sur son adresse IP, et chaque ressource partagée a une clé obtenue en appliquant une fonction de hachage sur son nom. Ensuite, les indexes des ressources sont distribués sur les pairs d'une manière spécifique de manière à ce que chaque pair de la DHT devient responsable de la portion d'index dont les clés sont numériquement proches de la clé de ce pair.En d'autres termes, les ressources sont placées dans la table de hachage distribuée dont la valeur de hachage est la plus proche de l'identifiant du pair, ce qui permet d'établir un lien entre les ressources et les pairs. De cette façon, la recherche d'une ressource revient alors à chercher le pair dont la clé est numériquement proche de la clé de la ressource recherchée. De plus, les nœuds possèdent des connaissances locales de l'ensemble du système distribué, qui sont souvent utilisées pour guider le routage des requêtes dans le système. Ces réseaux suivent donc un protocole global pour déterminer le pair responsable d'une telle ressource et dirige ensuite la recherche vers lui. Par conséquent, les requêtes de recherches peuvent être acheminées efficacement vers les pairs qui possèdent les ressources voulues, même si ces ressources sont extrêmement rares (Kumar et SubbaRao 2019) (Steinmetz et Wehrle 2005).

Plusieurs systèmes basés sur la DHT ont été proposés dans la littérature. Ils s'appuient sur des stratégies de routage et des schémas d'organisation différents, tels queChord, Pastry, Tapestry, CAN *Content Adressable Network*, Kademlia (Park et al. 2010) (Wang et Li 2003).

Contrairement aux réseaux P2P non structurés, la recherche des ressources dans les réseaux structurés est plus précise et plus rapide, permettant de diminuer le nombre de messages échangés de la restriction du degré d'autonomie des nœuds. Par conséquent, ces réseaux offrent une découverte de ressources complète et efficace ainsi qu'un trafic réseau réduit. Cependant, ils souffrent des coûts de stockage élevés en termes d'espace, de gestion et de maintenance, dus à l'utilisation des tables de routage. En effet, ces réseaux nécessitent un espace de stockage supplémentaire qui augmente linéairement avec le nombre de pairs et la quantité de ressources, entraînant une surcharge dans les réseaux. De plus, il est difficile de gérer et de maintenir les tables de routage lorsqu'un nœud rejoint ou quitte le réseau à un rythme élevé (Kumar et SubbaRao 2019) (Tlili 2011).

6. Application Pair-à-Pair

La communication Pair-à-Pair joue un rôle dominant dans les réseaux contemporains. Ce nouveau modèle de réseau ne s'est pas répandu en tant que principe, mais plutôt par ses applications qui ont permis son évolution et également sa prolifération dans de nombreux domaines. Une classification reconnue (Milogicic et al. 2002) scinde les domaines d'applications P2P en quatre catégories (Figure 2.6).

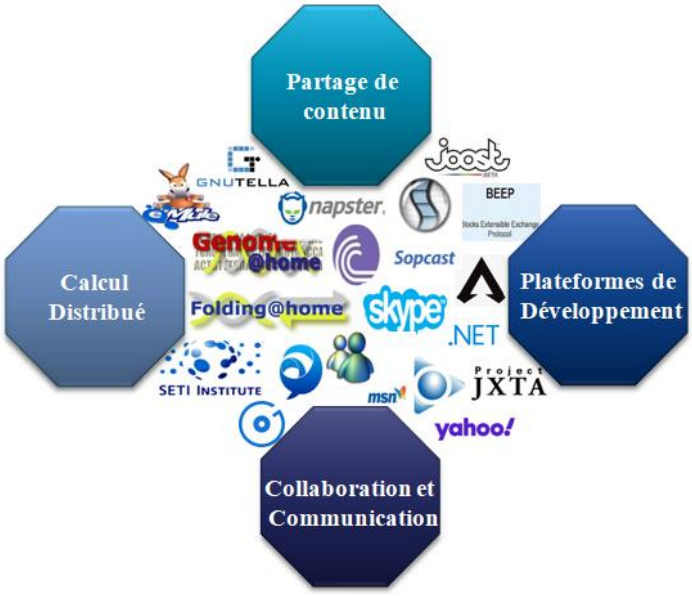


Figure 2. 6: Taxonomie des applications Pair-à-Pair (Milogicic et al. 2002).

6.1. Le Calcul Distribué

Le calcul distribué est l'un des premiers domaines applicatifs des technologies P2P. Les applications de calcul distribué offrent la possibilité aux internautes (machines connectées à Internet) de mettre à disposition une partie de leurs ressources informatiques (la puissance de calcul, la bande passante du réseau et l'espace disque, etc.) en vue d'accroitre le potentiel réseau. L'idée générale derrière ces applications est que les cycles d'inactivité de n'importe quelle machine connectée au réseau peuvent être utilisés pour résoudre les problèmes des autres ordinateurs nécessitant des calculs supplémentaires.

Les applications de calcul distribué permettent de diviser le problème de calcul qui sera à résoudre en petites portions indépendantes. Le traitement de chacune des parties de calcul est effectué sur un ordinateur connecté au réseau, en exploitant leurs ressources inutilisées. Chacun des ordinateurs possède un logiciel client qui profite des périodes d'inactivité dans la machine pour effectuer un calcul demandé par le serveur central. Ce dernier est chargé de distribuer les taches de travail aux ordinateurs sur le réseau et de collecter les résultats auprès de ces ordinateurs une fois le calcul terminé (Milogicic et al. 2002).

Plusieurs projets de calcul distribué ont été développés dans différents domaines. L'un des projets les plus populaires est SETI@home (Search for Extraterrestrial Intelligence) en astrophysique, qui vise la détection de la présence de civilisations extraterrestres avancées dans d'autres systèmes solaires par l'analyse de spectre électromagnétique. D'autres solutions ont été développées pour les organismes de recherche biologique et pharmaceutiques, telles que Genome@home et Folding@home, dédiés aux problèmes complexes de séquençage du génome et des protéines

(Amad et al. 2012) (Maurya et al. 2016) (Pourebrahimi et al. 2005). En marge de ces projets, le calcul distribué se développe notamment sous la forme du Grid Computing. Ce dernier correspond à une technologie informatique qui fournit une grande capacité de calcul au système distribué en utilisant des ressources largement réparties géographiquement et qui appartiennent à plusieurs organisations ayant des politiques différentes (Sungkar et Kogoya 2020). Ce même principe est utilisé également pour concevoir et maintenir des systèmes de stockage massif de données. Ces systèmes s'appuient sur la coopération des pairs à mettre à disposition leurs espaces de disques inutilisés aux autres pairs du réseau afin de sauvegarder leurs données d'une manière transparente et sécurisée (Begredj et Madaoui 2013) (Kumar et SubbaRao 2019).

6.2. Partage et Distribution du Contenu

Le partage de contenu est ledomaine où la technologie Pair-à-Pair a connu le plus de succès. Les applications de partage de contenus se concentrent sur le stockage et la récupération d'informations auprès de divers pairs du réseau. Elles consistent à échanger des contenus entre différents utilisateurs connectés simultanément à l'Internet (Amad et al. 2012) (Maurya et al. 2016) (Pourebrahimi et al. 2005). Ainsi, divers types d'applications Pair-à-Pair pour le partage de contenu ont été développés au fil du temps, dont le grand intérêt scientifique s'est principalement porté sur l'échange de fichiers, le streaming multimédia et la voix sur IP. L'ère du *partage de fichier Pair-à-Pair* (en anglais Peer-to-Peer File Sharing) a commencé avec le système d'échange de fichiers musicaux fortement célèbre, à savoir Napster, et s'est poursuivie avec des applications beaucoup plus puissantes qui ont introduit le P2P dans le courant dominant. Avec une variété de philosophies et de modes de fonctionnement, un bon nombred'applications departage de fichiers ont vu le jour, dont les plus populaires sont Gnutella, Fasttrack, eDonkey et BitTorrent. Par ailleurs, les évolutions de la technologie P2P ont permis son adaptation au *streaming multimédia* de type « store-and-forward », qui est un mode de transfert favorable surtout pour les fichiers multimédias qui sont volumineux. Le streaming P2P est inspiré du principe de partage de fichiers. Il permet à un ensemble d'utilisateurs finaux de partager des vidéos en coopérant et en échangeant les ressources réseau disponibles. Les utilisateurs finaux, une fois une vidéo réceptionnée, peuvent simultanément la visionner et retransmettre à leurs voisins sur le réseau les paquets déjà reçus. Ce concept technologique a révolutionné le domaine de la diffusion en ligne et a donné naissance à un certain nombre de systèmes de streaming P2P, tels que PPlive, PPStream, Sopcast et Joost (Alhaisoni et Liotta 2009). Joost est l'un des premiers systèmes streaming P2P permettant de fournir des services vidéo à la demande (en anglais Vidéo on Demande, abrégé VoD) complets et de haute qualité en utilisant les technologies P2P TV. Joost est un système Pair-à-Pair hybride. Il s'appuie à la fois sur des serveurs centraux pour fournir les contenus vidéo et sur les machines des utilisateurs. À tout moment, les utilisateurs de Joost peuvent avoir des vidéos cryptées dans leur ordinateur et les retransmettent automatiquement aux autres utilisateurs Joost qui les demandent. En outre, les systèmes Pair-à-Pair ont été utilisés pour le *partage de la voix sur IP* (en anglais Peer-to-Peer Voice over Internet Protocol, abrégé P2P VoIP), qui désigne l'utilisation conjointe des techniques de voix sur le réseau IP (ou téléphonie sur Internet) sur un réseau pair-à-pair. En d'autres termes, cette méthode permet de communiquer par la voix en utilisant le réseau Internet Pair-à-Pair, c'est-à-dire sans l'aide d'un serveur central (P2P VoIP, 2018). Skype est le client VoIP P2P le plus populaire. Il offre à ses utilisateurs la possibilité de communiquer entre eux via Internet en passant des appels, en envoyant des messages textes et en échangeant des fichiers. Il fournit également des informations sur la disponibilité de membres. Skype est une application qui se compte sur un fonctionnement totalement décentralisé. Néanmoins, il possède un seul élément central pour permettre l'authentification des utilisateurs (Guha et al. 2006).

6.3. Collaboration et Communication

Les applications P2P collaboratives visent à permettre aux communautés d'usagers situées à des endroits différents de collaborer et communiquer en temps réel, sans aucune nécessité d'une autorité centrale pour collecter et relayer les informations. La nature distribuée des systèmes P2P permet de fournir des environnements de collaboration performants où un espace virtuel est créé et mis à la disposition d'un groupe d'individus pour permettre un travail interactif. Ces applications sont variées et peuvent être utilisées dans des environnements professionnels, éducatifs et domestiques. En ce qui concerne la communication, le P2P offre une facilité de messagerie instantanée et des communications sans serveur (Kumar et SubbaRao 2019). Plusieurs applications sont très populaires auprès d'utilisateurs d'une grande variété, dont nous citons Yahoo !, AIM et Jabber. Le projet Jabber est un effort open source inventé par Jeremie Miller, visant à créer une architecture de messagerie instantanée ouverte tout en assurant une compatibilité et une communication transparente avec d'autres systèmes de messagerie instantanée populaires, tels que Yahoo, AOL et MSN. Ce projet apporte une valeur ajoutée aux utilisateurs de systèmes de messagerie instantanée en leur permettant d'échanger des messages et de présenter des informations avec d'autres utilisateurs, sans tenir compte du réseau propriétaire de messagerie instantanée qu'ils utilisent (Miller 2001). Jabber est également utilisé dans le domaine du partage et de la collaboration en quasi-temps-réel et d'échange multimédia via Jingle, dont la VoIP (téléphonie sur Internet), la visioconférence et l'échange de fichiers sont des exemples d'applications (Extensible Messaging and Presence Protocol, 2022).

Un autre aspect de la collaborationconsiste en des applications partagées permettant de travailler de manière commune sur un projet distribué. Groove, Magi et PowerPoint distribué sont des exemples de tels systèmes. Groove est un logiciel avec des capacités P2P qui a été acquis par Microsoft en avril 2005. Il fournit une variété d'applications pour la communication, le partage de contenu (fichiers, images et données de contact) et la collaboration (calendrier, regroupement, édition et dessin en collaboration et navigation Web collaborative) (Edwards 2002).

Des applications destinées au plus grand public se tournent aujourd'hui vers le P2P et ses solutions. L'une des applications les plus aptes à se répandre est celle des jeux en ligne. DOOM est une série de jeux vidéo de type First Person Shooter (abrégé *FPS*), développé par id Software et dans lequel de multiples joueurs peuvent collaborer ou s'affronter dans un environnement virtuel. DOOM utilise une structure P2P dans laquelle la machine de chaque joueur envoie des mises à jour de l'environnement virtuel (comme le mouvement du joueur) aux autres machines (Amad et al. 2012) (Milogicic et al. 2002) (Pourebrahimi et al. 2005).

6.4. Plateformes de Développement

Dans la mesure du manque de normes de conception des applications P2P, la majorité des logiciels P2P existants a été développée d'une manière spécifique sans référence à des standards propres au P2P. La normalisation des réseaux P2P est nécessaire pour permettre l'interopérabilité des applications. À cet égard, les plateformes P2P ont été implémentées en vue de fournir une infrastructure générique servant de base au développement d'applications P2P. Les plateformes P2P constituent un support pour les systèmes distribuées tout en assurant les principales fonctionnalités P2P, telles que la gestion des pairs, la communication entre les pairs dans le réseau, l'attribution de noms, la découverte des ressources, la communication, la sécurité et l'agrégation de ressources (Milogicic et al. 2002).

Diverses plateformes célèbres sont en concurrence pour devenir la future plateforme P2P. JXTA est la première tentative pour développer des normes pour le P2P. JXTA est une plateforme P2P Open Source, développée et lancée par Sun MicroSystems en 2001. Il permet de fournir une infrastructure informatique et de programmation réseau polyvalente. Il crée un système P2P en identifiant un petit ensemble de fonctions de base, nécessaires pour prendre en charge les applications P2P, et en les fournissant en tant que modules de base pour des fonctions de niveau supérieur. L'architecture de JXTA est divisée en trois couches où elle met en œuvre le modèle OSI : 1) JXTA core fournit un support de base pour le fonctionnement des applications et l'exécution des services Pair-à-Pair. Elle met en œuvre un ensemble de primitives permettant la découverte, le transport, la création et la suppression de groupes de pairs ainsi la gestion de leurs dynamiques. 2) La couche de services utilise les fonctionnalités de la couche précédente pour mettre en œuvre des services tels que la recherche et l'indexation, le partage de fichiers, la traduction de protocoles, l'authentification, etc. 3) La couche application permet de mettre en œuvre les applications intégrées à JXTA (Amad et al. 2012) (Maurya et al. 2016) (Pourebrahimi et al. 2005) (Schoder et al. 2005).

D'autres entreprises ont essayé de mettre en œuvre leurs propres normes, telle que NET proposé par la société Microsoft, BEEP (Blocks Extensible Exchange Protocol) (Rose 2001) et APEX (Rose et Klyne 2002) proposés par l'IETF (*Internet Engineering Task Force)*.S'agissant des propositions faites par des instituts universitaires, deux propositions principales de plateforme de développement P2P y figurent : Anthill (Babaoglu et al. 2002) et Shine (Yoshida et al. 2003).

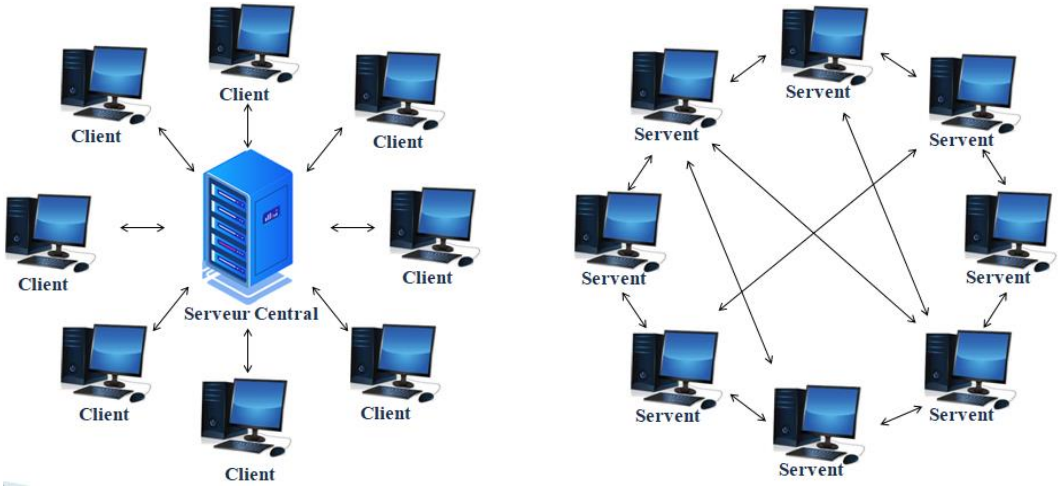
L'avènement du modèle P2P a donné lieu au développement croissant des applications et du partage de fichiers aux applications en temps réel. De nouveaux horizons et idées révolutionnaires pour les applications déployées font encore l'objet de recherches. Les gens veulent utiliser le P2P dans de nombreuses applications différentes, notamment le commerce électronique, l'éducation, le travail collaboratif, les applications mobiles, les réseaux sociaux, les moteurs de recherche, le développement d'applications blockchain, ...etc.

7. Architecture Pair-à-Pair VS Client-Serveur

Traditionnellement, l'échange de services entre ordinateurs est fondé sur le modèle Client-Serveur, qui est un réseau distribué constitué essentiellement de deux types de nœuds: un nœud performant (puissant) appelé « serveur » et plusieurs nœuds périphériquesmoins performants (de puissance inférieure) appelés« clients » (figure 2.7 (a)). Dans le modèle Client-Serveur, les ressources telles que le stockage ou la capacité de calcul peuvent être partagées entre le client et le serveur. Le serveur est l'unité centrale de contrôle et d'enregistrement et le seul fournisseur de ressources (contenus et services). Il est en général doté de capacités supérieures à celle des clients en termes de puissance de calcul, de ressources mémoires, etc (Steinmetz et Wehrle 2005). Cependant, les clients constituent les nœuds ordinaires du réseau qui se connectent au serveur afin d'accéder aux contenus ou exécuter les services offerts par ce dernier, sans partager aucune de leurs propres ressources (Schollmeier 2001).

Les systèmes Pair-à-Pair se posent comme une solution de rechange à l'architecture Client-Serveur, offrant un modèle informatique contemporain et moderne dans lequel il existe une approche plus hiérarchique de la mise en réseau. Le modèle P2P utilise un réseau composé d'un grand nombre de nœuds distribués, hétérogènes, autonomes et hautement dynamiques qui partagent une partie de leurs propres ressources informatiques, telles que la capacité de stockage, la puissance de calcul, la bande passante et le contenu. Dans cette architecture, il n'y a pas de distinction de capacités entre les différents pairsdu réseau. Les participants au réseau P2P peuvent agir en tant que serveur et client en

même temps, d'où l'appellation de « servent » ou simplement « pair » (figure 2.7 (b)). En d'autres termes, le pair peut lancer des demandes des ressources qu'il désire à d'autres pairs et, en même temps, répondre aux demandes qu'il reçoit de la part d'autres pairs sur le réseau en leur fournissant les ressources voulues. Ce type de réseau permet aux nœuds de communiquer directement entre eux sans passer par des entités intermédiaires ou un serveur central (Pourebrahimi et al. 2005) (Wang et Li 2003).



(a) Modèle Client-Serveur (b) Modèle Pair-à-Pair

Figure 2. 7: Modèle Client-Serveur VS modèle Pair-à-Pair.

À la différence des réseaux Client-Serveur, les réseaux Peer-to-Peer ne se reposent pas sur une infrastructure spécifique pour assurer les services de transport. En fait, ces réseaux forment des "structures superposées" axées sur l'allocation et la distribution de ressources par le biais des connexions directes. Une des principales différences entre les réseaux Client-Serveur et les réseaux Peer-to-Peer est que les premiers possèdent une seule entité centrale, qui conserve et fournit toutes les ressources et tous les services offerts (Schollmeier 2001). Dans une telle conception, les clients ne partagent aucune de leurs ressources informatiques. Les réseaux Peer-to-Peer sont fortement décentralisés, localisent une ressource désirée chez certains pairs participants et fournissent leurs adresses IP correspondantesau pair demandeur. De plus, le(s) serveur(s) dans les réseaux P2P centralisés ne conserve(nt) que les méta-informations relatives au contenu partagé au lieu de stocker le contenu lui-même. Par conséquent, ces systèmes permettent d'éviter la charge élevée qui résulte généralement d'un serveur central et de son réseau, ce qui entraîne une répartition plus uniforme de la charge sur le réseau physique sous-jacent (Steinmetz et Wehrle 2005).

En termes d'infrastructure, les réseaux P2P sont relativement moins chers et plus faciles à déployer que les réseaux Client-Serveur. Le modèle P2P diffère du modèle Client-Serveur où toute communication s'arrête si le serveur est en panne. Il est robuste et a une bonne capacité en termes de tolérance aux pannes car il n'y a pas de point de défaillance unique. Cela signifie que le désabonnement (normal en cas de déconnexion, anormal en cas de panne) d'un ou plusieurs nœuds n'a pas d'effet significatif sur le reste du réseau P2P, ce qui assure sa fiabilité et sa disponibilité. Les réseaux P2P offrentune architecture évolutive, ce qui signifie que la capacité totale etles performances du système s'améliorent au fur et à mesure qu'un nombre de pairs est ajouté au réseau. Cependant, avec une approche client-serveur, les performances du serveur se détériorent à mesure que le nombre de clients demandant des services au serveur augmente (Bashmal et al. 2017).

8. Défis des Réseaux Pair-à-Pair

Bien que les technologies P2P offrent un large éventail d'avantages grâce à leurs applications, elles sont confrontées à certains défis et problèmes qui doivent être relevés. Certains d'entre eux sont les suivants:

8.1. Désabonnement

Les systèmes P2P sont devenus, avec le développement de nouvelles architectures de réseau, une alternative efficace aux architectures client-serveur traditionnelles. A cet effet, les pairs participent au réseau superposé. Ils agissent comme client et serveur à la fois. Ceci signifie que les pairs sont tenus de fournir aux autres pairs, à la demande, leurs parts de ressources qu'ils auront été chargés de maintenir. Notons que les P2P sont basés sur une structure de réseau hautement dynamique dans laquelle les pairs peuvent quitter ou rejoindre le système à des moments arbitraires. Ce prossus, appelé *désabonnement* (en anglais, *churn*), est l'un des plus grands défis auxquels sont confrontés les réseaux P2P pour la plupart d'entre eux. Le *désabonnement* représente le changement dans l'ensemble de nœuds dans les réseaux à cause des arrivées et des départs anormaux (en cas de défaillances) ou normaux (en cas de déconnexion) de nœuds. Les nœuds, dans les conditions normales, rejoignent le réseau un par ou le quittent de façon gracieuse en informant leurs voisins afin que la topologie de la superposition soit restructurée. Cependant, s'il y a une défaillance d'un nœud, un grand pourcentage de nœuds quittent simultanément, silencieusement et fréquemment le réseau. Un environnement de réseau imprévisible est ainsi provoqué. Ce qui engendre une augmentation considérable des coûts et une dégradation des performances et de la qualité de service de tels systèmes caractérisés par une nature décentralisée et une forte dynamique. En effet, certains aspects qualitatifs pertinents des réseaux P2P peuvent être affectés de manière significative par un taux de désabonnement élevé, notamment leur stabilité, leur efficacité et leur évolutivité. L'aspect le moins problématique du désabonnement est généralement celui de l'impact de l'arrivée de nouveaux pairs, qui entraine des défaillances temporaires principalement, comme des incohérences ou des ressources pouvant se trouver de manière temporaire à un mauvais endroit dans la superposition. Cependant, des dommages irréparables peuvent être entrainés par le départ des pairs du système, par exemple la perte de ressources stockées ou de l'ensemble de la structure de superposition, les échecs de routage ou encore des vues incohérentes des pairs sur la superposition (Bashmal et al. 2017) (Trifa et Khemakhem 2014). Ainsi, afin de maintenir un fonctionnement correct et efficace et de garantir la disponibilité des ressources, il est essentiel que les systèmes P2P intègrent dans leurs architectures un degré adéquat de robustesse contre le désabonnement.

8.2. Free-Riding

Dans le partage des ressources et la communication, les réseaux Pair-à-Pair jouent un rôle important. Ils visent à permettre aux pairs de collaborer entre eux directement, la plupart du temps sans aucune coordination, ni contrainte, ni contrôle sur la participation des pairs individuels. Toutefois, un défi notable inhérent au comportement des pairs des réseaux, dénommé « *free-riding* », entrave le succès des services de tels systèmes ainsi que leurs performances. Il s'agit d'un un problème sérieux dans les réseaux P2P, tant pour les utilisateurs que pour les fournisseurs de ressources. Etant des pairs égoïstes, les free-riders agissent dans leurs propres intérêts. Ils profitent des ressources des autres sans offrir les leurs en retour. Ceci signifie qu'ils ne contribuent pas et ne collaborent pas avec les ressources du réseau alors qu'ils consomment celles fournies par d'autres pairs. De ce fait, une petite fraction de pairs seulement offre la majorité des contenus disponibles. Ceci compromet l'attribut d'équité des réseaux P2P de manière considérable. Un déséquilibrage de charge est créé lorsqu'il y a une

abondance élevée de free-riders dans le réseau, ce qui permet de punir les pairs qui contribuent activement au réseau et qui se retrouvent ainsi forcées à sur-utiliser leurs ressources. La plupart des demandes est dirigée, dans une telle situation, vers ces pairs surchargés. Une congestion du réseau est donc provoquée, engendrant une augmentation considérable de vulnérabilité et du temps de réponse. Par ailleurs, une dégradation des performances, affectant gravement la capacité des systèmes P2P à atteindre la collaboration et le partage, qui sont leur objectif global, est entraînée par ce phénomène. Ces problèmes font que la conception de mécanismes et de mesures rigoureux soit nécessaire. C'est ce qui permettra de gérer efficacement les free-riders dans les systèmes P2P et de lutter contre le free-riding (Alotibi et al. 2019) (Kamvar et al. 2003).

8.3. Sécurité

La sécurité est, jusqu'à présent, l'un des sujets les plus redoutables, particulièrement dans les réseaux distribués. Le P2P est souvent défini comme étant un type d'informatique distribuée dans lequel des nœuds d'influence égale, pour échanger des ressources dans un environnement autonome, dynamique et anonyme, communiquent directement entre eux. Les questions liées à la sécurité restent un défi majeur pouvant nuire au succès des systèmes P2P, bien que ces derniers aient bénéficié ces dernières années d'une énorme popularité. Les systèmes P2P, de par leur nature inhérente, sont susceptibles d'être mis en danger par leurs propres nœuds dans le cas où un nombre suffisant d'entre eux décide d'adopter un comportement malveillant. Par conséquent, les systèmes P2P se sont transformés en un terrain fertile pour les attaques. Ces dernières sont capables, potentiellement, de mettre en péril la confidentialité, la disponibilité, l'authenticité et l'intégrité du contenu, du système, des données ou des pairs. Les plus importantes attaques contre la sécurité des systèmes P2P sont, entre autres, l'attaque de l'homme au milieu, l'attaque déni de service distribué, l'attaque par empoisonnement du réseau, l'attaque par propagation des vers et/ou virus, l'attaque Eclipse et l'attaque Sybil.

L'*attaque de l'homme au milieu* (en anglais Man-In-The-middle Attack, abrégé MITM) constitue un cauchemar pour le monde du réseau. Le centre de l'attaquant se présente entre deux nœuds dans ce type d'attaque, afin de pouvoir manipuler les communications ou les espionner par le biais de l'envoi de fausses informations, de l'usurpation d'identité, de l'insertion ou de la rediffusion de messages précédents dans le flux de données, etc. (Gangan 2015). Il n'y a aucune garantie, au sein des réseaux P2P, que les utilisateurs envoient les ressources et les fichiers, ce qui contribue grandement à la propagation des vers malveillants et des virus. L'*attaque par déni de service distribué* (en anglais Distributed Denial of Service Attack, abrégé DDoS) représente les tentatives de paralyser et d'empêcher les pairs légitimes du réseau d'accéder ou de fournir un service ou une ressource, en en œuvrant de manière à ce que la cible ne puisse répondre ou soumettre aucune demande, ce qui la rend inutile. Le réseau P2P pourrait subir de graves dommages, même si ce type d'attaque est moins efficace sur les réseaux P2P, avec un nombre suffisant d'attaquants qui ciblent la plupart des utilisateurs (Washbourne 2015). « *Empoisonnement du réseau* », qui est une autre attaque ciblant les réseaux P2P, consiste à injecter des données inutiles (poison) dans le système. Un attaquant peut injecter de grandes quantités de données inutiles de type clé-valeur dans l'index (Li 2008), vu que les réseaux P2P, sous ses différentes architectures, sont tenus de mettre en œuvre un service de consultation (Li 2008). L'attaque par propagation des virus et des vers (en anglais, Virus and Worm Propagation) consiste en la diffusion des programmes malveillants susceptibles de se répliquer et se propagersur d'autres nœuds dans le réseau de manière automatique. Les risques de sécurité sont plus importants car la propagation des vers malveillants et des virus présente une rapidité élevée dans les réseaux P2P (Shredeh 2016). L'*attaque Eclipse*est autre attaque courante dans un réseau P2P. Un attaquant, dans une attaque Eclipse, contrôle une grande partie des voisins d'un bon nœud. L'attaquant peut ainsi manipuler le réseau superposéet contrôler la plupart des pairs honnêtes. Des pairs

malveillants peuvent, dans cette situation, travailler ensemble afin de tromper le bon nœud en en écrivant leurs adresses dans la table des voisins de ce dernier (López-Fuentes et al. 2012). De ce fait, de nouveauxconcepts, stratégies, politiques et technologies de sécuritésont nécessaires pour permettrel'interaction ainsi que le traitement distribué sécurisé dans les systèmes P2P.A cet effet, la recherche qui concerne la confiance et la sécuritédans les systèmes P2P doit être fondée sur l'expertisede la communauté sociologique et celle de la communauté de l'informatique (Amad et al. 2012). Par ailleurs, lors de l'*attaque Sybil* dans les réseaux P2P, un attaquant crée plusieurs fausses identités pour agir comme des nœuds accusatoires et infiltre le réseau entier. L'attaquant peut librement faire agir les pairs du réseau à sa guise en prenant ce dernier complètementen charge, et ce, une fois ces identitésacceptées au sein du réseau. Ceci peut rendre le réseau inutilisable par les autres participants. C'est donc potentiellement dangereux.

8.4. Disponibilité

La plus grande partie de la popularité des systèmes P2P provient de leur indépendance de tout contrôle centralisé et de toute infrastructure dédiée. Cependant, les systèmes P2P sont exposés par ces mêmes propriétésà des défis qui doivent être relevés pour que leur succès durable soit assuré. Le fait de répondre aux exigences en matière de disponibilité permanente dans le réseau constitue l'un des principaux défis.

Afin de fournir efficacement les services sur les systèmes P2P, il est essentiel de garantir la disponibilité desressources et des nœuds dont le service dépend de manière persistante.

Les systèmes P2P ne disposent pas d'une infrastructure dédiée. Ils dépendent des ressources qu'offrent les pairs participants, ces derniers mettant en œuvre à la foisdes fonctionnalitésde serveur et de client.Pour donner accès aux ressources apportées par chaque nœud d'un système P2P, ce nœud doit être capable d'accepter les messages des autres nœuds et de communiquer avec eux.Les systèmes P2P, en tant que systèmes distribués, peuvent subirdes conditions et des perturbations étendues difficilesà gérer qui les rendent indisponibles pendant d'importants intervalles de temps.La prévision ou la garantiede la disponibilité des ressources requises à un moment donné peut s'avérer difficile dans de tels systèmes, et ce, en raison de la naturedynamique et autonomedes membres de la communauté. En effet, les pairs des réseaux P2P peuvent quitter ou rejoindre à tout moment le réseauappelé « *désabonnement* », en fonction de leurs intérêts et de manière parfois spontanée et imprévisible.Cela engendre une disponibilité intermittentedes pairs et de leurs ressources informatiques, ce qui peut causer une interruption dans l'exécution des services, voire faire planter tout le système si le taux de désabonnement estélevé. Pareillement, les systèmes P2P à faible nombre de participantsqui se connectentaux réseaux de manière périodique ne peuvent pas garantir la durabilité et la disponibilitédes ressources et des services.Cette présence dynamique des pairsse répercute de manière directe sur la disponibilité des ressources et a un impact significatif surla qualité de l'expérience (en anglais Quality of Experience, abrégé QoE) des clients, la qualité des services (en anglais Quality of Service, abrégé QoS), les performances et la fiabilitédes systèmes P2P. Les participants à un système P2P sont censés, par ailleurs, contribuer aux ressources pour le bien commun de tous les pairs. Dans certaines situations très répandues en revanche, les ressources peuvent être disponiblesmais les pairs préfèrent qu'ils ne contribuent pas au contenu du système. Les ressources disponibles pour un système P2P sont réduites par la présence d'un grand nombre de ces pairs, appelés « free-riders », qui peut détériorer la qualité du service que le système est en mesure de fournir à ses utilisateurs. Par ailleurs, certains des pairs des systèmes P2P peuvent exposer ces derniers à des risques de sécurité pour nuire à leursperformances et à leurs fonctionnalités. Ces pairs malveillants tentent de cibler puis d'attaquer les pairs victimes de même que leurs ressourcesen provoquant leurs défaillances et donc leur indisponibilité. Des efforts considérables ont été consacrésà la recherche et au développement de

solutions technologiques dans ce contexte. Il s'agit de solutions se présentant sous diverses formes d'attaques dont le but est de réduire la disponibilité du contenu utilisable dans les réseaux P2P (Rodrigues et Druschel 2010) (Schoder et al. 2005) (Tlili 2011). Afin d'assurer les ressources offertes à une communauté et une bonne disponibilité des pairs, les infrastructures P2P doivent mettre en place des mécanismes de disponibilité pouvant pallier ces problèmes. Ces mécanismes doivent garantir le fait que les ressources et les pairs puissent être conservésperpétuellement, indépendamment des facteurs qui affectent l'activité du système, de manière à ce que les utilisateurs autorisés soient aptes à accéder aux informations requises lorsqu'ils en ont besoin. Ceci signifie que les systèmes P2P doivent être capable d'atteindre, de manière simultanée, la résilience face aux différentes lacuneset difficultés et les niveaux de disponibilité souhaités.

8.5. Découverte de Ressource

Un réseau Pair-à-Pair est un système distribué composé d'un nombre de pairs qui mettent leurs ressources à la disposition des autres pairs sur le même réseau et qui communiquent directement entre eux sans avoir besoin d'un serveur central. En tant qu'environnements dynamiques à grande échelle, ces systèmes fournissent une grande quantité de ressources informatiques hétérogènes, provenant de différentes sourcespour le partage de ressources et le calcul distribué (Bashmal et al. 2017). En dépit de leurs concepts avantageux, la découverte rapide et efficace de ressources appropriées dans les réseaux P2P reste l'un des principaux défis. La découverte de ressources dans de tels environnements en réseau fait référence au processus qui permet la recherche et la localisation d'une ou plusieurs ressources particulières existantes dans le réseau. En termes plus simples, il s'agit d'un mécanisme qui exige que chaque nœud demandeur puisse explorer et obtenir les ressources souhaitées dans le réseau en identifiant les nœuds qui les détiennent (Arunachalam et Vinayakumar 2021).

Dans les réseaux Pair-à-Pair, il n'existe pas d'index centralisé pour rechercher des informations sur l'emplacement des pairs et des ressources, et chaque nœud du réseau ne connaît que les ressources qu'il héberge actuellement. En effet, aucun des pairs constituant le réseau n'a une connaissance globale du réseau, ce qui fait de la recherche et de la récupération des ressources l'un des problèmes les plus difficiles, en particulier pour les grands systèmes contenant des milliers ou des millions de nœuds. Par ailleurs, le nœud sur lequel la ressource sera découverte n'est pas connu à priori et la topologie du réseau P2P varie dynamiquement au fur et à mesure que des nœuds rejoignent et quittent le réseau. En plus de leurs disponibilités dynamiques, la nature non dédiée des nœuds entraîne un taux élevé de renouvellement des ressources informatiques dans les systèmes Pair-à-Pair. Les pairs peuvent être empêchés de trouver les ressources nécessaires à cause de telles perturbations, chose qui peut menacer la stabilité et la capacité de surviede l'ensemble du système.Ces systèmes, en tant qu'environnement P2P non supervisé à grande échelle, peuvent facilement échouer et souffrir d'une anarchie générale engendrée par des problèmes inhérents au comportement malveillant des utilisateurs sans aucun contrôle minimum. Cela conduit, de plus, à un trafic de recherche élevé, à une congestion et à un faible équilibrage de la charge dans le réseau, ce qui cause une augmentation de la latence de recherche et une faible performance du réseau (Navimipour et Milani 2015) (Gao et al. 2012). Par conséquent, la mise en œuvre d'une méthode sophistiquée pour chercher et localiser efficacement une ressource requise ne sera pas facile. Une approche qualifiée de découverte de ressources devrait être capable de s'adapter aux conditions du réseau qui changent et trouver rapidement de bonnes solutions, tout en maintenant les performances du système. Pour ce faire, elle doit être en mesure de faire le compromis entre plusieurs critères de qualité, tels que le taux du trafic, la latence de découverte, le taux de succès et d'échec des requêtes, l'équilibrage de charge, la consommation de la bande passante, etc (Bashmal et al. 2017) (Ouaarab et al. 2014).

9. Conclusion

Dès leur apparition, les réseaux Pair-à-Pair ont connu un franc succès et sont devenus l'un des termes les plus discutés dans le domaine des technologies de l'information. Avec l'augmentation rapide de leurs actuelles utilisations et la prolifération de leurs applications sur Internet, les chercheurs s'y intéressent de plus en plus. L'objectif de ce chapitre était de fournir une vue d'ensemble des réseaux Pair-à-Pair. Dans cette optique, nous avons d'abord défini la notion du réseau P2P, puis nous avons présenté l'historique de son évolution, en commençant par son apparition à la fin des années 1960. Nous avons également mis en lumière les principales caractéristiques des réseaux P2P ainsi que leurs différentes facettes, structurées sous plusieurs architectures considérées comme des modèles alternatifs à celui fourni par l'architecture client-serveur traditionnelle. A cet égard, nous avons abordé la différence entre l'architecture Pair-à-Pair et l'architecture Client-Serveur. Ensuite, nous avons présenté les domaines d'application tendances des réseaux P2P. Enfin, nous avons fait part des principaux défis dans les environnements P2P. Bien qu'il reste encore des recherches à faire sur ce domaine, le P2P incarne l'avenir du partage de l'information que plusieurs technologies informatiques se précipitent pour adopter comme base de leur conception, notamment le Cloud Computing. Il permet en effet de décentraliser la réalisation des services et de mettre à disposition des ressources partagées dans le réseau, ce qui présente de nombreux avantages qui repoussent les limites du modèle Client-Serveur. Le chapitre suivant sera consacré à la présentation d'une étude sur le Cloud Computing, dans laquelle nous abordons l'élément de la combinaison des deux technologies ainsi que le modèle en réseau résultant auquel nous nous intéressons dans cette thèse.

Chapitre 3 : Cloud Computing

Sommaire

1. Introduction
2. Historique
3. Définition
4. Technologies Connexes
   - 4.1. Grid Computing
   - 4.2. Informatique Utilitaire
   - 4.3. Virtualisation
   - 4.4. Architecture Orientée Service
5. Architecture de base du Cloud Computing
   - 5.1. Utilisateurs Finaux
   - 5.2. API et Interface d'accès
   - 5.3. Réseau
   - 5.4. Centre de Données
   - 5.5. Virtualisation
6. Caractéristiques du Cloud Computing
   - 6.1. Libre-service à la demande
   - 6.2. Accès large au réseau
   - 6.3. Multi-location
   - 6.4. Mise en commun des ressources
   - 6.5. Élasticité rapide
   - 6.6. Service mesuré
   - 6.7. Système de Tarification Utilitaire
7. Modèles de services du Cloud Computing
   - 7.1. Infrastructure en tant que Service
   - 7.2. Plateforme en tant que Service
   - 7.3. Logiciel en tant que Service
8. Modèles de déploiement du Cloud Computing
   - 8.1. Cloud Public
   - 8.2. Cloud Privé
   - 8.3. Cloud Communautaire
   - 8.4. Cloud Hybride
9. Avantages du Cloud Computing
10. Défis de l'adoption du Cloud Computing
    - 10.1. Conception Centralisée
    - 10.2. Sécurité et Confidentialité
    - 10.3. Disponibilité
    - 10.4. Interopérabilité et Portabilité
    - 10.5. Performance
11. Vers une Architecture Cloud Computing basée sur Pair-à-Pair
    - 11.1. Présentation du Cloud Computing basé sur Pair-à-Pair
    - 11.2. Pourquoi un Cloud Computing basé sur Pair-à-Pair?
    - 11.3. Cloud Computing basé sur Pair-à-Pair VS Volunteer Computing
12. Conclusion

1. Introduction

Le Cloud Computing est apparu comme une technologie informatique révolutionnaire pour le secteur des Technologies de l'Information et de la Communication. Il s'agit d'une grande avancée technologique en réseau, qui ne cesse de se développer pour devenir le mot à la mode dans le domaine de l'informatique distribuée. Le Cloud Computing est un paradigme informatique qui implique la fourniture des ressources informatiques partagées, dynamiquement évolutives et virtualisées en tant que service à la demande aux utilisateurs finaux via l'Internet. Le Cloud Computing est une plateforme informatique distribuée dans de grands centres de données où les ressources sont accessiblespartout dans le monde et à tout moment dès qu'une connexion Internet est disponible. Il s'agit d'un nouveau type d'informatique utilitaire qui adopte un modèle de tarification basé sur l'utilisation, dans lequel les utilisateurs ne payent que pour ce qu'ils utilisent réellement et pour le temps dont ils ont besoin. Le Cloud Computing offre une meilleure résilience, une grande flexibilité et un coût réduit. Il promet une fiabilité, une efficacité et une évolutivité accrues. À la lumière des activités de recherche et de développement actuelles, le Cloud Computing a le potentiel pour se hisser au sommet, en revanche, il subsiste quelques défis à relever pour assurer sa croissance future (Hamze 2015) (Prasanth 2012).

Ce chapitre présente une étude du Cloud Computing. Tout d'abord, nous donnons un aperçu historique du Cloud Computing, puis nous présentons sa définition. Ensuite, nous présentons les principales technologies connexes avec lesquelles le Cloud Computing est comparé. Puis, nous décrivons les composants de base de l'architecture du Cloud Computing ainsi que ses principales caractéristiques. Nous abordons, par ailleurs, les modèles de service et les modèles de déploiement du Cloud Computing. Par la suite, nous élaborons les avantages du Cloud Computing et nous identifions ses principaux défis du point de vue de son adoption. Enfin, nous mettons la lumière sur les architectures Cloud basées sur les réseaux P2P.

2. Historique

Le Cloud Computing a connu une évolution rapide dans l'histoire et est récemment devenu le mot à la mode de la communauté de l'informatique distribuée et le moyen idéal pour fournir des solutions et des applications à différentes entreprises dans le monde. L'histoire du Cloud Computing a commencé au début des années 1960, lorsque John MacCharty, un des pionniers de l'intelligence artificielle, a prononcé une idée brillante lors d'un discours au MIT (Massachusetts Institute of Technology) : « Le calcul pourrait un jour être organisé comme un service public ». À l'époque, cette technologie informatique était mal appréciée par les gens qui croyaient que celle qu'ils utilisaient était suffisante pour eux.

Plus tard, en 1970, le terme "virtualisation", inventé dans les années 1960, a évolué pour décrire la création d'une machine virtuelle qui se comporte comme un système informatique entièrement fonctionnel. Par la suite, les entreprises ont commencé à offrir des services virtuels tels que des connexions VPN, ce qui a conduit au développement de l'infrastructure du CloudComputingvers les années 1990 (OPARA 2019).

Le terme « informatique en nuage » a été utilisé pour la première fois dans son contexte actuel lors d'une conférence donnée en 1997 par Ramnath Chellappa (Xing et Zhan 2012). Le monde technologique a attrapé l'idée et a commencé à la mettre en œuvre. Salesforce.com est apparu comme l'un des premiers acteurs de l'informatique en nuage qui a introduit le concept de fourniture des applications d'entreprise surl'Internet en 1999.

Le 21e siècle a connu, tout le long, la poursuite de la beauté de la phase informatique en nuage. En 2002, Amazon, le leader du commerce électronique (e-business), a pris le train en marche en lançant Amazon Web Services (abrégé AWS) pour fournir une variété de services sur l'Internet, tels que le stockage, le calcul etle réseau. En 2006, Amazon a introduit le service Web commercial Elastic Compute Cloud (abrégé EC2) dontl'idée consistait à ouvrir toutes les ressources inutilisées aux petites entreprises et aux particuliers, pour qu'elles les louent à la demande.L'année 2006 a également vu l'introduction du service Google Docs, qui a véritablement propulsé l'informatique dématérialisée au premier plan de la conscience publique (Xing et Zhan 2012). Ce service permet à unutilisateur de sauvegarder et de partager des documents avec précision avec divers utilisateurs (OPARA 2019).Il a été rapidement suivi par une collaboration, à l'échelle de l'industrie en 2007, entre Google, IBM et un certain nombre d'universités aux États-Unis. Puis vint Eucalyptus en 2008, la première plateforme open source compatible avec l'API AWS pour le déploiement de Clouds privés, suivie par Open Nebula, le premier logiciel open source pour le déploiement de Clouds privés et hybrides. Après cela, en 2009, Google Play a également commencé à fournir des applications d'entreprise de Cloud Computing. Ainsi, en 2009, Microsoft a lancé Microsoft Azure et d'autres entreprises comme Alibaba, IBM, Oracle et HP ont également introduit leurs services de Cloud Computing.Avec le temps, le Cloud Computing a gagné en popularité dans la presse et est devenu une compétence très populaire et importante (Hamze 2015).

3. Définition

Le terme Cloud Computing abrégé « CC » (traduit de l'anglais « Informatique en Nuage », connu aussi sous le terme l'Informatique Dématérialisée) a l'allure d'un mot à la mode en informatique. Le Cloud Computing désigne la nouvelle tendance d'hébergement et de fourniture des services Internet qui s'appuient sur des serveurs distants pour gérer les tâches. Le Cloud Computing est un changement de paysage de l'informatique orienté service où l'infrastructure et les solutions informatiques sont fournies en tant que service aux utilisateurs à la demande via Internet (Bandela et al. 2015) (Shahzad 2014). Ce paradigme a concrétisé le rêve de l'informatique utilitaire et a récemment attiré beaucoup d'attention. Cette technologie n'est pas nouvelle, mais il s'agit plutôt d'un nouveau modèle d'exploitation issu de la combinaison de plusieurs technologies existantes afin de répondre aux exigences variantes de la demande actuelle dans le secteur des technologies de l'information (abrégé TI) (Syed et Baig 2013) (Zhang et al. 2010). Depuis son apparition, de nombreuses définitions du « Cloud Computing » ont été proposées par le secteur académique et industriel. Cependant, la définition la plus utilisée dans le monde informatique est celle qui provient de l'institut national des normes et de la technologie (en anglais National Institute of Standards and Technology, abrégé NIST). NIST a défini le Cloud Computing (Singhal et al. 2007) ainsi :

« A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models. »

Cette définition couvre les fonctionnalités intéressantes et les caractéristiques avantageuses du Cloud Computing qui le rendent attrayant pour les propriétaires d'entreprise et le distinguent de toute autre technologie informatique. Le Cloud Computing est basé sur un modèle informatique dans lequel un ensemble de ressources informatiques est offert et affecté d'une manière dynamique et personnalisée à plusieurs utilisateurs, ce qui permet de réduire les coûts d'infrastructure et d'augmenter la vitesse de développement des applications. Le Cloud Computing repose sur l'Internet qui sert d'un réseau de

fourniture de services. Il apparaît aux utilisateurs comme un point d'accès unique pour tous leurs besoins informatiques. Les ressources Cloud incluent, entre autres, les serveurs, du stockage, des applications et des services qui sont accessibles par les utilisateurs de n'importe où et à tous moment via différents appareils, dont les ordinateurs de bureau, les ordinateurs portables, les téléphones mobiles ou n'importe quel assistant numérique personnel du moment qu'il dispose d'une connexion Internet. Les ressources informatiques peuvent être provisionnées, libérées dynamiquement en tant que service et exécutées en fonction des besoins actuels des utilisateurs pour évoluer rapidement et à la demande, ce qui peut réduire les dépenses d'exploitation. Généralement, les utilisateurs appelés «Cloud Service Users, abrégé CSU» ne sont pas, en effet, propriétaires de l'infrastructure informatique. Ils possèdent cependant la possibilité d'allouer ou d'accéder à des services informatiques du Cloud. Des fournisseurs de services Cloud dénommés « Cloud Service Providers, abrégé CSP» fournissent ces services. Un fournisseur de services Cloud peut être une entité, une personne ou une organisation ayant en charge la mised'un service à la disposition des consommateurs de services Cloud.Afin d'assurer des services fiables, l'infrastructure du Cloud Computing repose sur des centres de données constituant un ensemble de serveurs avec différents niveaux de technologies de virtualisation. Cet ensemble est le concept clé qui permet au Cloud de réaliser la multi-location et qui fait référence à la création logique d'une ressource informatique qui n'existe pas physiquement, incluant des serveurs d'applications et des périphériques de stockage virtuels, des disques durs par exemple. La virtualisation permet de regrouper des ressources informatiques issues de clusters de serveurs et d'affecter ou réaffecter de manière dynamique des ressources virtuelles à des applications à la demande. Le Cloud Computing a changé la promesse de l'informatique basée sur l'utilisation.En réalité, il adopte un système de tarification utilitaire, ce qui signifie que les utilisateurs ne paient que ce qu'ils utilisent réellement et pour le temps dont ils ont besoin (Bandela et al. 2015) (Prasanth 2012) (Zeng 2018) (Zhang et al. 2010).

4. Technologies Connexes

Le Cloud Computing est un paradigme technologique influent dans le secteur des TI, qui a révolutionné la manière dont les ressources informatiques sont gérées et utilisées (Xing et Zhan 2012). Le développement du Cloud Computing est dû essentiellement à la présence de plusieurs autres technologies et perçu comme une extension de celles-ci.Ces technologies incluent, notamment, le Grid Computing, utility Computing, la virtualisation et l'architecture Orientée Service qui partagent, chacune, certains aspects avec le Cloud Computing.

4.1. Grid Computing

Le Grid Computing est une forme d'informatique distribuée qui consiste à coordonner et partager des ressources informatiques, d'applications, de données et de stockage ou des ressources réseau dans une organisation dynamique et géographiquement dispersée afin d'atteindre un objectif commun, à savoir résoudre une seule tâche. Son apparition a été principalement motivée par des problèmes informatiques réels issus de la recherche scientifique avancée et a promis de changer la façon dont les organisations abordent ces problèmes complexes. Le Grid Computing transforme les grands problèmes en plusieurs petites unités appelées grilles et les diffuse àplusieurs serveurs ou ordinateurs qui sont reliés ou connectés par un réseau communafin d'offrir une puissance de calcul (Hashemi et Bardsiri 2012).

Le Cloud Computing et le Grid Computing sont deux technologies informatiques assez similaires fondées sur les réseaux et ayant toutes les deux une vision commune consistant à fournir des

services en partageant des ressources entre un grand nombre d'utilisateurs. Ces deux concepts impliquent la mise en commun de ressources distribuées pour atteindre des objectifs communs.Toutefois, le CloudComputing élimine la complexité de l'achat de matériel et de logiciels pour créer des applications en tirant parti des technologies de virtualisation et de multi-location pour réaliser le partage des ressources distribuées et leur approvisionnement dynamique. En outre, la gestion des ressources informatiques dans le Cloud Computing se fait de manière centralisée et elles sont situées sur plusieurs serveurs regroupés dans les centres de données des fournisseurs de Cloud Computing. Le Grid Computing, à la différence du Cloud Computing, est considéré comme un système distribué pour le partage collaboratif des ressources. Afin d'atteindre un objectif commun, il combine des ressources informatiques de différents domaines. Le Grid Computing est en fait un vaste réseau d'ordinateurs interconnectés dont chacunpeut accéder aux ressources de tous les autres ordinateurs. Ainsi, ils travaillent ensemble sur une tâche donnée. Le groupe d'ordinateurs, afin de fournir un accès évolutif et transparent à des ressources informatiques étendues qui sont géographiquement réparties, agit comme un superordinateur virtuel. Il les présente comme une ressource unique et unifiée afin d'exécuter des applications à grande échelle, comme l'analyse d'énormes ensembles de données (Khillar 2018).

4.2. Informatique Utilitaire

L'informatique utilitaire « en anglais, Computer Utility » est un concept établi à la fin des années 1960 par John McCarthy qui avait prédit que l'informatique pourrait un jour être organisée comme un service public tel que : les services téléphoniques, l'eau, l'électricité et le gaz. S'inspirant du modèle des services publics traditionnels, l'informatique utilitaire consiste à fournir des services informatiques en s'appuyant sur une méthode de facturation à la demande et à l'usage. La conception sous-jacente de l'informatique utilitaire repose sur un modèle informatique à la fois commercial et technologique, dans lequel les fournisseurs de services possèdent, exploitent et gèrent les ressources et l'infrastructure informatiques et les mettent à la disposition de leurs clients (utilisateurs/consommateurs). Ces derniers accèdent ensuite aux ressources informatiques fournies et les utilisent selon leurs besoins, et paient les fournisseurs pour une utilisation spécifique plutôt qu'un tarif fixe. L'informatique utilitaire permet aux utilisateurs d'obtenir des quantités appropriées de ressources auprès des fournisseurs de manière dynamique, en fonction de leurs besoins et exigences spécifiques en matière de services. Cela est particulièrement utile pour les utilisateurs dont les besoins informatiques augmentent et diminuent d'une manière rapide et imprévisible.Le concept avantageux de l'informatique utilitaire offre aux utilisateursune meilleurerentabilité ainsi qu'une grande flexibilité et facilité d'adaptation à l'évolution de leurs besoins et de leurs environnements (Chana et Kaur 2013) (Yeo et al. 2007) (Sood et al. 2016).

En fait, l'informatique utilitaire est devenue l'un des modèles de services informatiques les plus populaires.Le Cloud Computing peut être considéré comme étant une réalisation de l'informatique utilitaire. Il adopte, à des fins économiques, un système de tarification essentiellement fondé sur l'utilité. Les fournisseurs de services peuvent maximiser réellement l'utilisation efficace des ressources et minimiser leurs coûts d'exploitation associés grâce à l'approvisionnement des ressources à la tarification basée sur l'utilité et à la demande.

4.3. Virtualisation

La virtualisation est une technologie en développement dans le monde des technologies de l'information. Elleconsiste à créer une version virtuelle (plutôt que réelle) de n'importe quoi, incluant, mais sans s'y limiter, une plateforme matérielle informatique virtuelle, un système

d'exploitation ou des dispositifs de stockage, ou encore des ressources de réseau informatique (Naeem et al. 2016). Cette technologie a vu le jour, dans les années 1960 sur les ordinateurs centraux, en tant queméthode permettant de répartir logiquement les ressources du système fournies par les ordinateurs centraux entre différentes applications. Depuis lors, le terme de virtualisation s'est répandu et est devenu largement admis par les organisations informatiques qui y ont de plus en plus recours. La technologie de la virtualisation fait abstraction des détails du matériel physique et fournit des ressources virtualisées pour les applications de haut niveau. Elle permet la création logique des ressources informatiques qui n'existent pas physiquement, chacune pouvant être utilisée pour interagir avec un utilisateur simultanément. La virtualisation constitue la base de l'architecture du Cloud Computing etl'un de ses principaux composants contribuant à son émergence. Elle permet de mettre en commun une variété de types de ressources informatiques virtualiséesdans desclusters/groupes de serveurs et de les affecter ou les réaffecter dynamiquement aux applications à la demande. Elle permet également au fournisseur de services d'offrir une infrastructure informatique abstraite et émulée sous forme de service homogène simultanément à tous les clients. Le Cloud Computing s'appuie fortement sur la technologie de virtualisation pour atteindre le but de fournir des ressources informatiques à des locataires multiples, selon l'utilité et de façon dynamique. La combinaison intelligente des technologies du Cloud Computing et de la virtualisation favorise la réduction des dépenses d'investissement initiales des clients en leur permettant de faire évoluer leur infrastructure informatique au fur et à mesure de leur évolution, sans avoir à se soucier d'un approvisionnement excessif ou insuffisant (Hamze 2015) (Xing et Zhan 2012).

4.4. Architecture Orientée Services et Service Web

Au cours de la dernière décennie, l'architecture orientée services « en anglais, Service Oriented Architecture ou SOA» a pris de l'ampleur.C'est un concept intéressant résultant de l'évolution exponentielle des systèmes informatiques. L'architecture orientée services est unearchitecture technologique qui se concentre sur la conception et la construction de systèmes basés sur des services.Dans cette architecture d'application, des composants modulaires, accessibles, auto-descriptifs, faiblement couplés,découvrables, abstraits, autonomes, réutilisables et interopérables sont publiés sous forme de services qui peuvent être invoqués et consommés à distance via des interfaces standardisées invocables par d'autres applications ou combinés avec d'autres services (Fremantle et al. 2002) (Stencil 2002). Un service est un composant clé de l'architecture orientée services, consistant en une fonction bien définie, autonome et qui ne dépend d'aucun contexte ou de l'état d'autres services. L'architecture orientée services est destinée essentiellement à former une collection de services qui interagissent et communiquent entre eux, généralement à l'aide de protocoles standards, et qui peuvent fonctionner dans des systèmes et des domaines d'activité distincts à des fins de collaboration ou de partage. Cette approche favorisela flexibilité et l'évolutivité des applications et contribue à réduire les coûts tout en garantissant une solution solide de services commerciaux aux clients.

L'architecture SOA fournit une approche permettant la description et l'organisation des services et leurs interactions afin de faciliter leur découverte et leur utilisation (Schulte et Natis 1996). La *publication* (en anglais, *advertising*), la *découverte* (en anglais, *discovery*) et *l'invocation* des servicesreprésentent les opérations fondamentales accomplies par les composants d'une telle architecture. Chaque composant de l'architecture SOA peut remplir un ou plusieurs des trois rôles suivants: fournisseur de services (en anglais *Provider*), consommateur de services (en anglais *Consumer*) et registre de services (en anglais *registry* ou *repository*). En effet, les fournisseurs de services publient les services disponibles dans le registre des services pour que les consommateurs

puissent en profiter.Ce sont aussi des entités adressables par le réseau qui acceptent et répondent aux demandes des consommateurs. Le *registre de services* enregistre et classe les services publiés dans un référentiel des services disponibles. Il fournit des facilités et des capacités de découverte permettant de rechercher et localiser des fournisseurs de services pour les consommateurs de services intéressés. Ces derniers exploitentdonc le registre des services pour trouver les services qui répondent à leurs critères et les utiliser en fonction de leurs intérêts. Pour cela, les consommateurs de services procèdent à l'invocation du service qui représente la connexion et l'interaction du client avec le service en fonction des informations contenues dans sa description. La *description du service* détermine la manière dont un consommateur de services interagira avec le fournisseur de services. Elle spécifie le format de la demande de service et de la réponse, en précisant les paramètres d'entrée du service ainsi que le format et le type des données retournées. Cette description peut spécifier également un ensemble de conditions préalables, de conditions postérieures et/ou de niveaux de qualité de service (Endrei et al 2004).

Les services Web sont la réalisation la plus populaire de l'architecture orientée services (SOA) visant la prestation des services sur le Web. Autrement dit, les services Web fournissent une approche informatique distribuée pour intégrer des applications extrêmement hétérogènes sur l'Internet. Ils se reposent sur des technologies et des langages désormais standardisés.

- ***La norme UDDI (Universal Description Discovery Integration)*** recommandée par l'OASIS est un annuaire de services fondé sur XML, destiné à agir comme un courtier en informations entre les consommateurs et les fournisseurs de services. L'UDDI permet de publier et de découvrir des services Web de manière programmatique ou par le biais d'une interface d'utilisateur graphique généralement basée sur le Web. Par ailleurs, la norme UDDI adopte un mécanisme pour spécifier la manière de stocker, de localiser et de récupérer les services web ainsi que les informations associées dans une topologie Internet (Endrei et al 2004) (Gurugé 2004) (Singhal et al. 2007).
- ***Le protocole SOAP (Simple Object Access Protocol)*** développé par le World Wide Web Consortium (abrégé W3C), est un protocole de communication léger basé sur XML (Extensible Markup Language) pour permettre l'échange d'informationsstructurées via HTTP dans un environnement distribué décentralisé. Il permet également l'accès aux services web et l'interopérabilité des applications à travers le web.Le protocole SOAP repose essentiellement sur la technologie XML en vue de maintenir la simplicité et la possibilité d'extension (Vanneau et al. 2005).
- ***Le langage WSDL (Web Service Description Language)*** est un langage de spécification/description standard, recommandé par W3C et fondé sur le XML pour pouvoir décrire les communications d'une manière structurée afin d'utiliser un service.Il définit une grammaire XML pour décrire les services en réseau comme des ensembles de points de terminaison de communication capables d'échanger des messages. Le langage WSDL fournit aux fournisseurs de services un moyen simple de décrire le format de base des demandes adressées à leurs systèmes (Christensen et al. 2001).
- ***Le langage XML (Extensible Markup Language)*** est un langage de balisage généraliste recommandé par le W3C. XML occupe un rôle important dans de nombreux systèmes informatiques. C'est un langage incontournable et universel. Il a été conçu pour les descriptions efficaces et l'échange / publication de données sur le web (Shreve 2007).
- ***Le protocole HTTP (Hypertext Transfer Protocol en français, protocole de transfert hypertexte),*** est un protocole de communication populaire et basique pour tout échange de données sur le Web. Il est utilisé pour le transport des demandes de service et des réponses entre

les clients Web (par exemple les navigateurs Web) et les serveurs Web suivant le modèle classique client-serveur.Un client http ouvre une connexion et envoie un message de demande à un serveur HTTP. Le serveur, à son tour, renvoie un message de réponse (Pourebrahimi et al. 2005).

Le Cloud Computing et la SOA sont deux paradigmes orientés vers la réutilisation des services informatiques existants en vue de favoriser la flexibilité des entreprises, bien que leurs objectifs soient différents (Surianarayanan et Chelliah 2019). La SOA est une approche dans l'architecture, visant à fournir aux entreprises/clients des systèmes comprenant des services réutilisables et interchangeables. En revanche, le Cloud Computing a pour but de fournir une gamme complète de services réutilisables en fonction des besoins des entreprises/clients, allant de l'infrastructure matérielle de base aux applications logicielles en passant par la plateforme. Le Cloud Computing est plus ou moins basé sur l'évolution des technologies de l'architecture orientée services « SOA » et du Web Service, en particulier le modèle de service SaaS (Xiong 2018). L'architecture du Cloud est basée sur une approche modulaire permettant la flexibilité et la réutilisation des services. Cela est dû au fait que les bases de ces services ainsi que l'infrastructure du Cloud sont bien conçues et bien architecturées grâce à l'adoption des architectures orientées services (Hurwitz et al, 2010).

5. Composants Basiques de l'Architecture du Cloud Computing

L'architecture du Cloud Computing repose sur un ensemble de composants de base qui sont essentiels à la réalisation et au fonctionnement de ce concept technologique. Le développement de l'ensemble de la structure du système de Cloud Computing est soumis aux responsabilités de ces composants qui peuvent être présentés comme suit:

5.1. Utilisateurs Finaux

Le Cloud Computing est typiquement utilisé par des particuliers, des entreprises, des employés d'entreprise, des employés en déplacement, etc., qui utilisent les différents services Cloud à l'aide de tout type d'appareil connecté. Avec les avancées technologiques, les ordinateurs de bureau et les ordinateurs portables mais aussi les appareils mobiles, netbooks, tablettes, PDA, etc., sont devenus des terminaux (Khethavath 2014).

5.2. API et Interfaces d'accès

Les utilisateurs ont besoin d'un moyen pour accéder au Cloud à distance, étant donné qu'ils ne peuvent jamais se retrouvent à l'intérieur dans les centres de données. Les fournisseurs des services Cloud mettent à disposition des API et des interfaces d'accès qui permettent aux utilisateurs d'accéder eux-mêmes aux services et aux ressources requises via un portail Web. Autrement dit, les utilisateurs du Cloud accèdent aux API et aux interfaces via l'Internet et consomment les services voulus (Surianarayanan et Chelliah 2019).

5.3. Réseau

Le réseau se réfère à une infrastructure vaste par laquelle les fournisseurs de services de Cloud offrent leurs services aux utilisateurs. Cette infrastructure comprend le câblage physique, les commutateurs, les équilibreurs de charge et les routeurs qui assurent la disponibilité permanente de l'infrastructure du Cloud pour répondre aux besoins des clients. L'Internet est la principale base qui rend le CloudComputing possible. En fait, les ressources informatiques sont généralement fournies aux

utilisateurs via l'Internet, ce qui nécessite que les fournisseurs de services tiers créent et entretiennent l'infrastructure réseau qui rend cela possible (Surianarayanan et Chelliah 2019).

5.4. Centre de Données

Le centre de données ou centre de traitement «en anglais Data center, abrégé DC» fait référence à une infrastructure physique (immobilière et technique) qui constitue le cœur des services numériques, notamment ceux fournis via la technologie du Cloud Computing. Il est défini comme un site ou station physique hébergeant l'ensemble des systèmes, des installations et des équipements utilisés dans le domaine IT et non-IT (informatique et non-informatique) nécessaires au fonctionnement des systèmes et des applications informatiques (BENAC et al. 2015) (Maaref 2012). Les entreprises et lesclients du Cloud reposent fortement sur cette infrastructure physique afin d'organiser, traiter, stocker et entreposer de grandes quantités de données.

Un data centre est constitué d'un ensemble de composants élémentaires intégrés dans un bâtiment spécialisé et sécurisé (Maaref 2012):

- *Les équipements informatiques* constituent des serveurs d'application sur lesquels s'exécutent les logiciels, des serveurs de stockage de données, des baies et armoiresqui accueillent des serveurs etdifférents composants informatiques, des dispositifs de réseau interconnectant les serveurs (notamment les routeurs (router), les pare-feu (firewalls), les répartiteurs de charge (load balancers) et les commutateurs(switch) ainsi que des outils de gestion des systèmes et des équipements de réseau. L'ensemble de ce parc informatiqueest maintenu dans des salles sécurisées et assure principalement le traitement, le stockage et les échanges internes et externes.
- *L'infrastructure* est formée des locaux et des équipements (non-IT ou non informatiques) nécessaires pour assurer la continuité des opérations du centre de données. Ces équipements incluent les transformateurs électriques, les alimentations électriques, les générateurs, les armoires de climatisation, les dispositifs de refroidissement des équipements et des baies, les systèmes de distribution électrique, etc.
- *Les espaces d'exploitation* désignant le personnel d'exploitation qui pilote, entretient et répare les systèmes (informatiques et non informatiques) en cas de besoin.

L'infrastructure physique du Cloud Computingest fondée principalement surles centres de données avec 10.000 ou plus de serveurs sur site de taille très variable: d'une salle de quelques mètres carrés à plus de 10.000 mètres (Maaref 2012). Ces serveurs sont reliés à des réseaux de grande vitesse et sont consacrés au stockage et au traitement des données, à l'exécution des applications et à la fourniture des différents services aux clients du Cloud. Généralement, les centres de traitement des données ne sont pas situés au même endroit mais sont plutôt répartis sur différents sites géographiques.Ils peuvent être propres aux clients finaux, qui exploitent eux-mêmes leurs équipements, principalement des serveurs, ou bien être utilisés par des hébergeurs, qui exploitent des serveurs pour le compte de leurs clients selon des modalités bien définies (Hurwitz et al. 2010).

5.5. Virtualisation

Dans le monde informatique actuel, la virtualisation est un facteur technologique indispensable dans le Cloud Computing. Elle présente deux propriétés importantes, à savoir l'abstraction et l'encapsulation. Elle consiste à créer une couche abstraite de virtualisation, placée au-dessus de la couche physique (entre le matériel et le logiciel) de l'architecture du Cloud. La virtualisation permet de partitionner les serveurs et les ressources physiques en conteneurs virtuels, appelés généralement des machines virtuelles « en anglais, Virtual Machines (VMs) ». Ces VMs fournissent une plateforme permettant

d'installer et de configurer des systèmes d'exploitation et des applications afin de fournir tous les services qu'un serveur physique fournit habituellement (Singh et al. 2018). La création et le fonctionnement des machines virtuelles sont assurés par des micro-logiciels ou des programmes de bas niveau appelés «Hyperviseurs», qui agissent comme des gestionnaires des machines virtuelles (Subbareddy et Begam 2021).

La technologie de virtualisation est largement utilisée dans les centres de données du Cloud Computing, notamment au niveau des serveurs. Par le biais de la virtualisation, plusieurs serveurs virtuels sont autorisés à fonctionner simultanément sur un seul serveur physique. De ce fait, la virtualisation des serveurs offre une flexibilité accrue en matière d'équilibrage des charges et de reconfiguration des serveurs lors des évolutions ou des pannes temporaires. Outre les serveurs, les dispositifs de stockage et les réseaux entre autres, constituant l'architecture physique du Cloud, peuvent tous être virtualisés de sorte que le nombre de ces ressources matérielles peut être réduit.

Le concept avantageux de la virtualisation a conduit à son adoption dans le Cloud Computing pour satisfaire les exigences commerciales des accords dans le service. En effet, la virtualisation peut améliorer la mutualisation des ressources et favoriser leur approvisionnement rapide et élastique. Cela permet la mise en place d'un environnement Cloud flexible et agile, conduisant à une réduction significative des coûts en termes d'investissement, de consommation d'énergie et de systèmes de refroidissement.

6. Caractéristiques du Cloud Computing

L'informatique dématérialisée présente plusieurs caractéristiques saillantes qui démontrent leurs différences par rapport aux approches informatiques traditionnelles et les rendent attrayantes pour les propriétaires d'entreprises. Il s'agit de :

6.1. Libre Service à la Demande

Cette caractéristique signifie que les fournisseurs de services en nuage peuvent approvisionner automatiquement les ressources informatiques en tant que service, conformément aux besoins de leurs clients. Les consommateurs ayant des besoins instantanés à des moments précis peuvent donc accomplir unilatéralement toutes les actions nécessaires pour acquérir les ressources informatiques requises, de manière pratique et en libre-service, sans nécessiter d'intervention humaine auprès de chaque fournisseur de services (Dillon et Chang 2010) (Singhal et al. 2007).

6.2. Accès Large au Réseau

Le Cloud Computing offre des ressources informatiques qui sont diffusées pour être disponible sur le réseau et accessible via l'Internet. Cela a comme objectif de favoriser l'utilisation des ressources fournies par divers utilisateurs/consommateurs, dotées de plates-formes/appareils hétérogènes légères ou lourds connectés à l'Internet (tels que des téléphones mobiles, des tablettes, des ordinateurs portables et des assistants personnels numériques ou PDA) (Dillon et Chang 2010) (Singhal et al. 2007). En outre, pour atteindre une performance élevée du réseau en termes d'accessibilité, beaucoup de Cloud aujourd'hui sont constitués de centres de données situés à plusieurs endroits à travers le monde. Par ailleurs, cette géo-diversitéestun facteur important pour les fournisseurs de services qui peuvent facilement tirer parti de ses bénéfices pour obtenir une utilité maximale des services dans les environnements de CloudComputing (Zhang et al. 2010).

6.3. Multi-locataire

La multi-location est un élément crucial du Cloud Computing. Initialement, elle difinissait le fait qu'une instance de logiciel unique est utilisée pour servir plusieurs utilisateurs ou locataires. Récemment, le terme a pris un sens plus large, faisant référence à un ensemble de ressources partagées dans le réseau plutôt qu'à une simple instance logicielle partagée. La multi-location est une forme d'architecture du Cloud Computing dans laquelle plusieurs clients d'un même fournisseur de Cloud utilisent les mêmes ressources informatiques. Chaque client est appelé «locataire». Bien que les ressources sont partagées, les clients du Cloud ne sont pas conscients les uns des autres et leurs données sont totalement séparées. La multi-location permet à plusieurs instances (de plusieurs consommateurs de nuages) d'une application donnée de fonctionner sur une seule application (c'est-à-dire la même machine logique), offrant une meilleure utilisation des ressources à un coût beaucoup plus faible (Dillon et Chang 2010).

6.4. Mise en Commun des Ressources

Le Cloud Computing permet aux utilisateurs un accès distant et facile à un réseau performant très riche par une variété de services, à l'aide d'un navigateur Web, indépendamment de leur emplacement ou de l'appareil qu'ils utilisent (Prasanth 2012). Les fournisseurs d'infrastructure Cloud offrent un pool/ensemble de ressources informatiques qui sont mises en commun afin de servir plusieurs consommateurs en s'appuyant sur la multi-location ou la virtualisation, avec différentes ressources physiques et virtuelles attribuées et réattribuées dynamiquement en fonction de la demande du consommateur. Cette capacité d'affectation dynamique des ressources offre une grande flexibilité aux fournisseurs d'infrastructure pour gérer l'utilisation de leurs propres ressources et les coûts d'exploitation. Elle permet de maximiser l'utilisation des ressources dans le réseau tout en minimisant les coûts, notamment ceux de consommation d'énergie et de refroidissement (Zhang et al. 2010). Par conséquence, le résultat de cette caractéristique est que les consommateurs ne possèdent aucun contrôle ou connaissance globale sur le réseau. Généralement, les consommateurs ne sont pas en mesure de savoir l'emplacement exact des ressources informatiques physiques fournies, mais peuvent être capables de spécifier l'emplacement à un niveau d'abstraction plus élevé (pays, état ou centre de donnéespar exemple). En outre, ils ne peuvent pas également savoir leurs formationset leurs originalités (Dillon et Chang 2010) (Shahzad 2014) (Singhal et al. 2007).

6.5. Élasticité Rapide

Le Cloud Computing se caractérise par une conception élastique qui reflète sa capacité à s'adapter aux besoins et aux demandes en approvisionnant et désapprovisionnant des ressources de manière automatique et rapide, de telle façon à ce que les ressources fournies soient conformes à la demande actuelle des utilisateurs. Pour les consommateurs, les ressources informatiques deviennent immédiates plutôt que persistantes et sont fournies sans aucune nécessité d'engagements ou de contrats initiaux. En outre, ces ressources informatiques disponibles semblent souvent illimitées pour eux, et peuvent êtreacquises et utilisées à leur guise, en n'importe quelle quantité et à n'importe quel moment, et libérées dès qu'elles ont fini pour d'autres utilisations. A la différence du modèle traditionnel qui fournit des ressources en fonction des pics de demande, la consommation via l'approvisionnement dynamique peut augmenter et diminuer rapidement afin de répondre aux besoins à tout moment,ce qui permet de réduire et économiser considérablement les coûts d'exploitation lorsque la demande de services est faible (Dillon et Chang 2010) (Shahzad 2014) (Zhang et al. 2010).

6.6. Service Mesuré

Cette caractéristique reflète la capacité de mesure du Cloud Computing. Les systèmes Cloud sont capables decontrôler et optimiser automatiquement l'utilisation des ressources en s'appuyant sur des mécanismes de mesure appropriés afin de quantifier l'utilisation de ces ressources pour chaque consommateur individuel. L'utilisation des ressources peut être surveillée, contrôlée et signalée, en assurant la transparence autant pour le fournisseur que pour le consommateur du service utilisé (Dillon et Chang 2010) (Singhal et al. 2007).

6.7. Système de Tarification Utilitaire

L'une des caractéristiques distinctives du Cloud Computing est le modèle de tarification adopté pour utiliser ses services. Les services Cloud sont facturés à la carte. Il s'agit de facturation à l'utilisation «pay as you go». Plutôt de payer un taux forfaitaire mensuel ou annuel, les utilisateurs des services Cloud ne paient que pour les ressources qu'ils utilisent réellement et pour le temps dont ils ont besoin (Bandela et al. 2015). D'autre part, les fournisseurs de services n'ont pas besoin d'investir dans l'infrastructure pour commencer à bénéficier du Cloud et offrir leurs services. Ils louent simplement des ressources du Cloud en fonction des besoins des utilisateurs. La tarification basée sur l'utilité permet de faire des économies considérables en réduisant le coût d'exploitation des services Cloud (Zhang et al. 2010). Bien que le modèle de tarification le plus fréquemment utilisé dans le Cloud est le paiement à l'utilisation, il existe quelques services Cloud qui sont gratuits.

7. Modèles du Service Cloud

Le Cloud Computing a évolué en tant que paradigme populaire et universel pour l'informatique orientée services. Il utilise un modèle commercial axé sur les services où les ressources sont fournies en tant que services à la demande des utilisateurs finaux. Ces services sont offerts en termes detrois modèles, à savoir Infrastructure en tant que service, Plateforme en tant que service et Logiciel en tant que service. Ces modèles de fourniture de services de Cloud Computing sontégalement connus sous le nom «Pyramide du Cloud Computing» ou «Pile du Cloud Computing», du fait qu'ils se superposent les uns sur les autres (Figure 3.1) (Crutcher et al. 2021).

Figure 3. 1: Modèles des Services Cloud.

7.1. Infrastructure en tant que Service

Le modèle de services Infrastructure en tant que Service (en angais Infrastructure as a Service, abrégé IaaS) est le plus basique et le plus flexible du Cloud Computing. Il fait référence à la mise en place d'une infrastructure Cloud Computing (serveurs /machine virtuelle, stockage, réseaux, systèmes d'exploitation et autres ressources informatiques fondamentales) offerte sous forme de service à la demande et accessible sur l'Internet. Ces ressources informatiques sont généralement fournies dans un environnement virtualisé pour que les utilisateurs peuvent les obtenir en tant que service et les utiliser pour déployer et exécuter leurs applications. La virtualisation dans le Cloud IaaS permet decréer un pool de ressources virtuelles en intégrant/décomposant les ressources physiques de manière ad hoc pour répondre à la demande croissante ou décroissante en ressources de la part des consommateurs de Cloud (Bandela et al. 2015) (Jagli et Gupta 2013) (Prasanth 2012) (Parekh 2014). Les propriétaires du Cloud fournissant l'IaaS sont appelés des fournisseurs IaaS. Le modèle IaaS fournit essentiellement une infrastructure informatique virtualisée qui est hébergée et gérée par les fournisseursdes services dans les centres de données Cloud. Les clients du Cloud souscrivent, achètent les ressources auprès d'un fournisseur de services en nuage et les utilisent en tant que service en fonction de leurs besoins. Ils achètent généralement des services IaaS sur la base d'une tarification utilitaire. Par conséquent, le modèle IaaS offre un accès rapide et facile à l'infrastructure informatique, permettant aux particuliers et aux entreprises de se débarrasser des coûts initiaux et de la complexité liée à l'acquisition et à la gestion de leurs propres infrastructures. Il assure également une grande évolutivité et une réduction des dépenses en termes d'argent et de temps (Crutcher et al. 2021) (Tchao et al. 2021).

Parmi les exemples de fournisseurs IaaS, nous citons GoGrid, Flexiscale, Cisco Metapod ou Google Compute Engine et Amazon Elastic Compte Cloud « Amazon EC2 ». Ce dernier est la solution de Cloud d'Infrastructure proposée par Amazon et l'un des services fondamentaux de son offre Amazon Web Services. Ce service repose sur une infrastructure Cloud composée de plusieurs serveurs informatiques répartis sur diverses régions dans le monde. Amazon EC2 offre aux entreprises et aux particuliers la possibilité de louer des serveurs sur lesquels ils exécutent leurs propres applications web sur le Cloud Public. Il permet un déploiement extensible des applications où les clients peuvent créer des instances de serveur constituées en machines virtuelles et configurer facilement la mise à l'échelle de la capacité des instances à l'aide de l'interface web EC2. Selon leurs besoins, les clients peuvent créer, lancer et arrêter des instances de serveurs, et payent en fonction du temps d'usage des serveurs, d'où le terme «Élastique». EC2 offre aux utilisateurs la possibilité de placer des instances de serveurs à plusieurs emplacements dispersés géographiquement, de telle façon qu'il soit possible, en cas de panne, de restaurer les instances défaillantes et d'assurer la continuité du service (Zhang et al. 2010).

7.2. Plateforme en tant que Service

Le modèle Plateforme en tant que service (en anglais Platform as a Service, abrégé PaaS) est un middleware des modèles des services Cloud qui fournit aux développeurs d'applications un environnement de développement en tant que service. Ce modèle offreune infrastructure informatique comprenant les installations matérielle et logicielle nécessaires (un système d'exploitation, un environnement d'exécution de langage de programmation, une base de données et un serveur Web) pour prendre en charge l'intégralitédu cycle de vie complet, de la création et de la fourniture d'applications et de services web entièrement disponibles sur l'Internet. L'infrastructure informatique sous-jacente fournie est abstraite pour les développeurs d'applications. Ces derniers utilisent une plateforme Cloud pour développer, tester, déployer, héberger et gérer leurs solutions logicielles en tant que service PaaS sans avoir connaissance de ce qui se passe dans l'infrastructure sous-jacente à ce service. Par conséquent, le modèle PaaS permet aux développeurs de se débarasser des coûts et des

complexités liés àl'achat, la mise en place et la gestion de l'infrastructure sous-jacente de matériel et de logiciels nécessaires au développement et au déploiement des applications (Bandela et al. 2015) (Crutcher et al. 2021) (Jagli et Gupta 2013) (Prasanth 2012) (Syed et Baig 2013).

Certains exemples particuliers de fournisseurs de PaaSsont Google App Engine, Amazon Web Services, Microsoft Windows Azure et Force.com. *Microsoft Windows Azure* est la solution de Cloud Computing PaaS proposée par Microsoft. Il a été désigné récemment comme étant la plateforme du Cloud public la plus utilisée. C'est une plateforme fonctionnant en tant que service qui fournit aux développeurs des capacités de calcul et de stockage à la demande pour héberger, faire évoluer et gérer des applications Web sur l'Internet par le biais des centres de données de Microsoft. Windows Azure est une plateforme ouverte qui prend en chargedes outils, des langages et des environnements Microsoft et non-Microsoft tels qu'Eclipse, Ruby, PHP et Python. La plateforme Azure est flexible, évolutive et interopérable. Elle peut être utilisée pour créer de nouvelles applications à exécuter à partir du Cloud ou pour améliorer les applications existantes avec des capacités basées sur le Cloud.*Google App Engine* est une autre plateforme informatique en Cloud, permettant de développer, de stocker, d'héberger et d'exécuter des applications Web sur les centres de données gérés par Google en utilisant des langages de développement tels que java et python. Il constitue une plateforme extensible offrant aux utilisateurs la possibilité de créer et de configurer des applications de manière simple et rapide, ainsi que d'évoluer en fonction de l'augmentation du trafic et des besoins de stockage (Zhang et al. 2010).

7.3. Logiciel en tant que Service

Le logiciel en tant que service, parfois appelé « logiciel à la demande », est un modèle de distribution de logiciels qui consiste à fournir des applications complètes aux utilisateurs à la demande sur l'Internet.Il est également connu sous le terme « SaaS », qui désigne l'expression anglophone « Software as a Service ». Dans le modèle SaaS, les fournisseurs des services sont connus sous le nom de « SaaS Providers ». Ils hébergent leurs applications dans les centres de données Cloud dont l'utilisation des interfaces de ces applications se fait généralement par le biais d'un navigateur web standard. Ilsconcèdent leurs applications à ses clients sous forme de service à la demande dans un environnement multi-locataire, en s'appuyant sur la technologie de virtualisation où un grand nombre d'utilisateurs peut exploiter une instance du service hébergée côté serveur (Kaushal et al. 2015) (Sandanayake et Jayangani 2018). Ces services sont fournis soit par le biais d'un abonnement via un modèle de paiement à l'utilisation ou gratuitement, et sont accessibles par les dispositifs des clients issus de partout dans le monde tant qu'il existe une connexion d'Internet. Les services gratuits constituent une catégorie particulière de services logiciels distribués sur l'Internet. Ces services sont fournis lorsqu'il est possible de générer des revenus à partir de flux de non-utilisateurs, tels que le financement publicitaire (Prasanth 2012). Les plateformes des réseaux sociaux et les services de diffusion de médias, dont Facebook et Youtube, constituent les exemples les plus fréquents. Par ailleurs, les services payants sont souvent régis par des accords de niveau de service (en anglais Service Level Agreement, abrégé SLA) entre les fournisseurs et les utilisateurs finaux. Les accords de niveau de service définissent formellement le niveau de service auquel le client peut s'attendre et doivent aborder la latence et la QoS. Ils précisent également les compensations fournies aux clients dans le cas où les niveaux de service convenus ne sont pas respectés, y compris d'indisponibilité du service vendu (Höfer et Karagiannis 2011). Le SaaS est considéré comme le leader des modèles de fourniture des services Cloud, et se situe au sommet de leur pyramide. Le SaaS est utilisé dans Microsoft Office 365, Google Apps, Salesforce.com, Citrix GoToMeeting, Onlive, Cisco WebEx et Netflix, VoIP de Vonage, Skype, etc.Salesforce.com, considéré comme le pionnier des fournisseurs de logiciel en tant que service, est situé à San Francisco aux États-Unis.Salesforce.com a été créé par

Mark Benioff avec la vision de faire la gestion de la relation client (en anglais Customer Relationship Management, abrégé CRM) en proposant des outils d'analyse commerciale et de CRM qui soient bénéfiques aux entreprises et aux centres d'affaires. Salesforce.com fournit principalement des logiciels en tant que service pour les ventes et le marketing. Il propose également une multitude de fonctionnalités et de services utiles pour le développement simple et rapide d'applications logicielles (Panimalar et al. 2017) (Syed et Baig 2013).

Le modèle SaaS offre des services Cloud bénéfiques visant le plus grand nombre de particuliers et d'entreprises du fait que leur utilisation ne nécessite pas de connaissances particulières en informatique et en télécommunications. En premier lieu, il permetd'éliminer, ou plutôt réduire, les coûtsd'investissement initial relatif aux logiciels et aux serveurs. Dans un modèle SaaS, le client n'achète pas le logiciel mais le loue via un modèle de facturation par abonnement en fonction de ses besoins d'utilisation, qui peuvent être adaptés de manière dynamique. En outre, il fournit des applications hébergées et gérées dans le Cloud où leur livraison utilise une approche d'accès standardisés avec le Web. Ce qui permet aux clients de jouir simplement des applications logicielles, via l'Internet, sans se soucier des aspects de développement, de déploiement et de maintenance. Finalement, les applications SaaS se distinguent fortement des autres services Cloud parune architecture multi-tenante et évolutive. Pour y parvenir, les opérations peuvent être clonées sur plusieurs machines virtuelles durant l'exécution afin de répondre à l'évolution des besoins changeants des clients (Bandela et al. 2015) (Crutcher et al. 2021) (Prasanth 2012) (Syed et Baig 2013) (Zhang et al. 2010).

8. Modèles De Déploiement

Un modèle de déploiement définit la nature du Cloud, son objectif et son emplacement en se répondant aux besoins des services et des utilisateurs en particulier. Le Cloud Computing peut être déployé sous différents modèles, dont chacun a ses propres avantages et inconvénients (Crutcher et al. 2021) (Hashemi et Bardsiri 2012) (Syed et Baig 2013).

8.1. Cloud Public

Actuellement, le Cloud Public est le modèlede déploiement le plus dominant du Cloud Computing. Ilfait référence à une infrastructure volumineuse et très efficace, dans laquelle les ressources sont mises à la disposition du grand public en tant que service à la demande.Les clients accèdent à ces ressources sur l'Internet et les utilisent en fonction de leurs besoins via un paiement à la carte ou gratuitement. L'infrastructure du Cloud Public est détenue, gérée et exploitée par des fournisseurs de services en nuage qui sont entièrement responsables de sa propre politique, sa valeur, ses avantages, ses coûts et son modèle de facturation. De nombreux services en nuage populaires sont des nuages publics, notamment Amazon EC2, Google AppEngine et Force.com.

Le Cloud public constitue une solution avantageuse pour les usagers et particulièrement les petites organisations, permettantde réduire ou éliminer les coûts d'investissement élevés dans l'infrastructure. En revanche, il ne garantit pas un contrôle précis sur les données, le réseau et les paramètres de sécurité, ce qui entrave son efficacité dans de nombreux scénarios (Bandela et al. 20015) (Dillon et Chang 2010) (Dhote et Deshpande 2016) (Zhang et al. 2010).

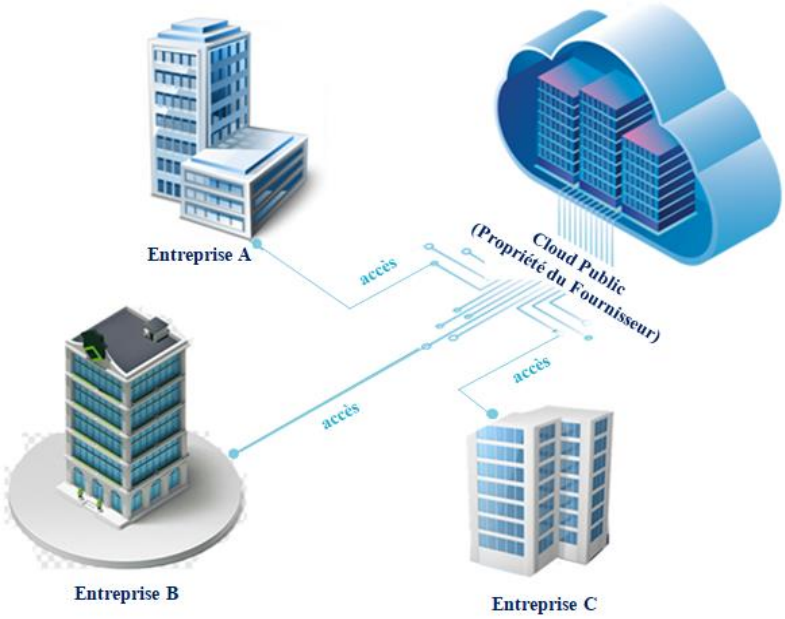


Figure 3.2: Cloud Public.

8.2. Cloud Privé

Le Cloud Privé, connu aussi sous le nom de Cloud Interne, consiste en une infrastructure Cloud où les ressources sont conçues et utilisées exclusivement par une organisation spécifique. Il réside généralement derrière le pare-feu de l'organisation et sur un réseau privé installé au sein de l'entreprise.La gestion et l'hébergement des nuages privés sont généralement assurés parl'organisation elle-même, mais certaines organisations font appel à des fournisseurs de services tiers, ou encore à une combinaison des efforts de ces entités. Eucalyptus, OpenNebula et OpenStack sont des exemples de solution pour la mise en place du Cloud privé.

L'orientation vers le modèle privé est motivée par le degré de contrôle fourni sur les performances, la fiabilité et la sécurité (Bandela et al. 2015) (Zhang et al. 2010). En effet, le Cloud Privé maximise l'exploitation et l'utilisation des ressources internes existantes. Par ailleurs, les données des utilisateurs du Cloud Privé sont restreintes au centre de données de l'entreprise et ne sont paspartagées avec d'autres organisations, qu'elles soient gérées par un fournisseur de Cloud ou par un fournisseur de Cloud tiers (Kumar et Ramachandhra 2014) (Syed et Baig 2013). Par conséquent, les risques de sécurité dans ce modèle sont plus faciles à détecter, ce qui le rend idéal pour la protection des informations confidentielles et les applications clés. Le Cloud privé peut néanmoins être intéressant pour les entreprises dont les exigences réglementaires sont très strictes. Cependant, il est souvent critiqué pour ses inconvénients associés à la réduction des dépenses d'investissement élevées dans l'infrastructure informatique. Les organisations propriétaires du Cloud sont responsables de la création, du déploiement et de la gestion de l'ensemble de l'infrastructure et des logiciels, ce qui fait du Cloud privé un modèle moins économique que le Cloud public dans la plupart des cas (Crutcher et al. 2021) (Kahanwal et Singh 2012) (Zhang et al. 2010).

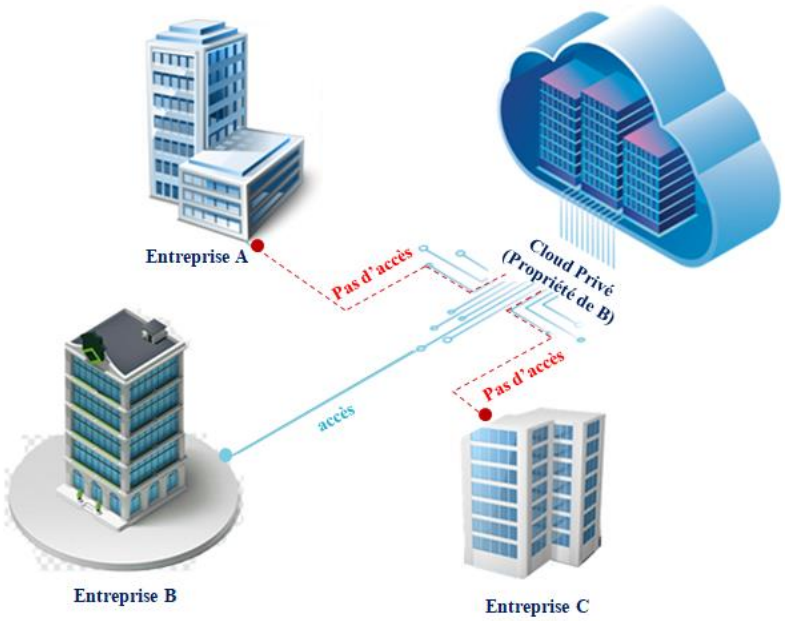


Figure 3.3: Cloud Privé.

8.3. Cloud Communautaire

Dans ce modèle, l'infrastructure du Cloud est partagée entre plusieurs organisations d'une communauté donnée ayant des politiques, des exigences, des valeurs, des intérêts et des préoccupations communes. L'infrastructure Cloud peut être hébergée, gérée et exploitée par une ou plusieurs organisations de la communauté, un fournisseur de services tiers ou une combinaison de ces entités.Ce modèle de déploiement peut contribuer à limiter les coûts d'investissement pour sa mise en place car les coûts sont partagés entre les organisations. Parmi les exemples de Cloud communautaires figurent (Bandela et al. 2015) (Dillon et Chang 2010) (Dhote et Deshpande 2016) (Kahanwal et Singh 2012) (Kumar et Ramachandhra 2014) (Syed et Baig 2013) (Tchao et al. 2021):

- GSA (General Services Administration) aux Etats Unis : Il a lancé, pour les organisations gouvernementales américaines, un site communautaire.
- Amadeus : Il a été créé il y a 20 ans par Air France, Lufthansa, Iberia et SAS. C'est le fournisseur principal de solutions informatiques à l'industrie du voyage et du tourisme. C'est actuellement le premier acteur mondial dans le domaine des voyages, avec 2 500 informaticiens mobilisés, plus de 150 compagnies aériennes clientes et 280 millions de transactions quotidiennes.
- CMed : Lancée en 2010, le but de cette startup était de créer un nouveau Cloud communautaire destiné aux laboratoires pharmaceutiques.

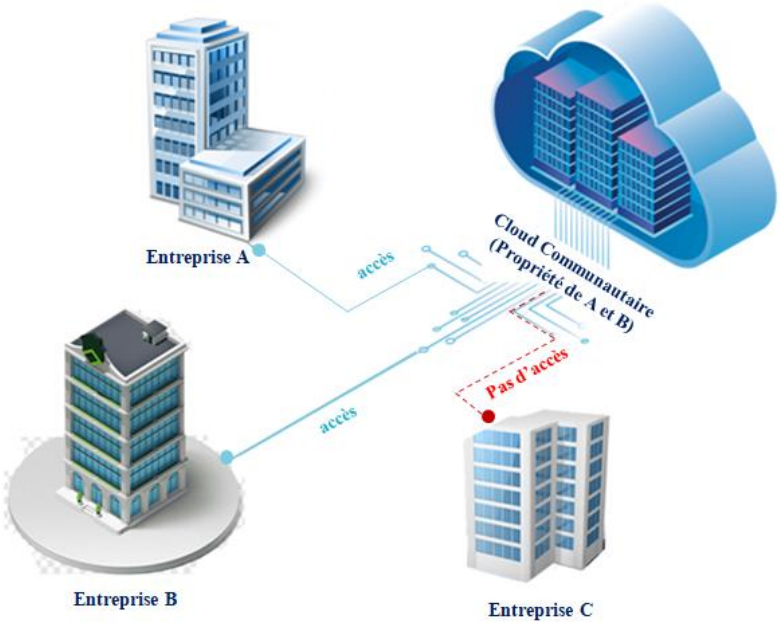


Figure 3.4: Cloud Communautaire.

8.4. Cloud Hybride

Le Cloud hybride est, comme son nom l'indique, un modèle de déploiement où l'infrastructure en nuage comprend deux ou plusieurs types de Cloud (publics, privés ou communautaires) offrant les avantages de plusieurs modèles de déploiement. Il utilise une technologie structurée et propriétaire pour connecter plusieurs systèmes en nuage de manière à permettre aux programmes et aux données d'être facilement déplacés d'un système de déploiement à un autre (Bandela et al. 2015) (Syed et Baig 2013) (Tchao et al. 2021). Le Cloud hybride a surmonté les enjeux de standardisation et d'interopérabilité du Cloud tout en permettant la portabilité des données applicatives. Il offre plus de flexibilité qu'un Cloud Privé et Public en assurant un meilleur compromis entre l'évolutivité d'un Cloud Public et la sécurité et le contrôle d'un Cloud Privé (Crutcher et al. 2021) (Dillon et Chang 2010). La conception des Clouds hybrides doit faire l'objet d'une étude détaillée afin de pouvoir déterminer avec précision la meilleure répartition entre les composants des nuages publics et privés qui répondent aux besoins (Zhang et al. 2010).

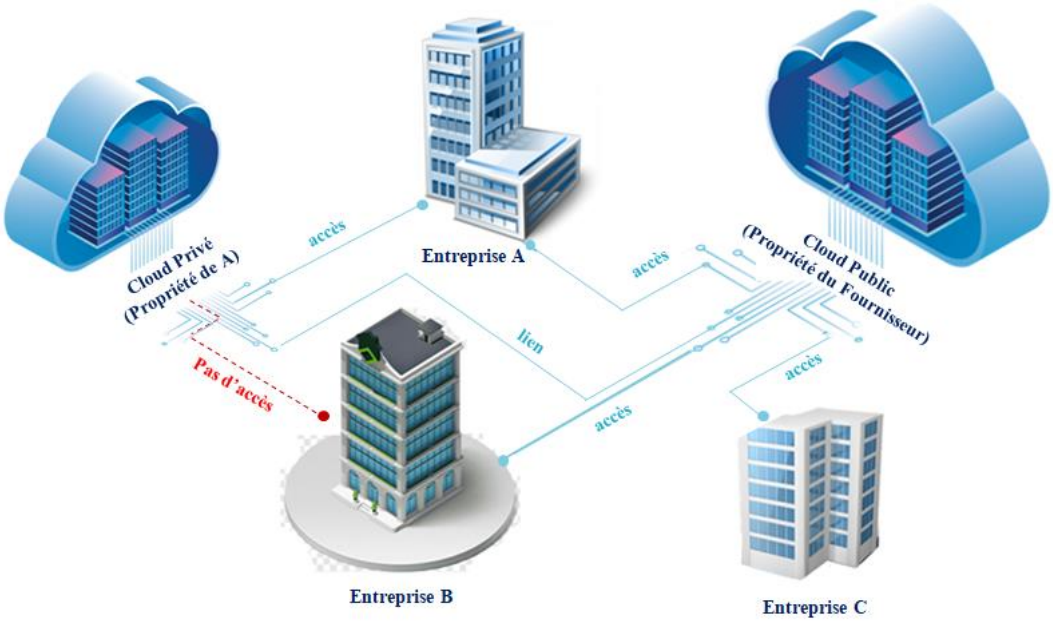


Figure 3. 5: Cloud Hybride.

9. Avantages du Cloud Computing

Le Cloud Computing, en tant que paradigme informatique innovant pour l'hébergement et la fourniture de services sur l'Internet, a suscité beaucoup d'intérêt au cours des dernières années. Il s'agit d'une solution attrayante pour les propriétaires d'entreprises et d'organisations, dont beaucoup ont déjà adopté ou sont en voie de définir une stratégie de « Cloud Computing » pour en tirer des avantages optimaux. La technologie duCloudComputingoffre aux organisations de nombreux avantages, qui sont les suivants :

1. Flexibilité

Le Cloud Computing est une technologie informatique fondamentalement flexible, permettant à ses utilisateurs d'accéder à un ensemble de ressources informatiques qui peuvent évoluer dynamiquement pour répondre à leurs besoins en temps réel. Les utilisateurs peuvent adapter l'infrastructure informatique disponibleet la redimensionner si nécessaire selon leurs besoins, personnaliser les applications et accéder aux services de Cloud Computing depuis n'importe quel endroit dès qu'ils disposent d'une connexion à l'Internet. La flexibilité du Cloud Computing permet aux ressources engagées pour une application de croître et de diminuer rapidement suivant les charges de travail réelles, de sorte que les consommateurs du Cloudne payent que pour le temps dont ils ont besoin et pour ce qu'ils utilisent réellement. Cela permet de mieux satisfaire les demandes dynamiques des utilisateurs sans prendre des risques financiers associés à la prévision exacte de la demande future (Bauer et Adams 2012) (Syed et Baig 2013).

2. Évolutivité

L'évolutivité est unpoint de pivot du Cloud Computing qui reflète sa capacité de s'étendre dynamiquement, de manière facile et rapide, pour répondre aux demandes changeantes des utilisateurs.Autrement dit, ce critère fait référence à la capacité du Cloud Computing à s'adapter aux charges variables en matière de traitement, de mise en réseau, d'infrastructure et de logiciels tout en assurant une stabilité sans faille. L'évolutivité dans le Cloud est soutenue par le modèle d'utilisation et de fourniture des ressources adopté par le Cloud, et particulièrement parles fonctionnalités de

virtualisation et de multi-location adopté par cette technologie. Les fournisseurs de services Cloud peuvent facilement mettre à disposition les ressources logicielles et matérielles nécessaires à l'expansion et à l'adoption rapide. En utilisant l'infrastructure existante, le Cloud Computing permet aux utilisateurs d'ajuster les ressourcesinformatiques en augmentant ou en diminuant l'infrastructure informatiqueà mesure de l'évolution de leurs besoins, sans pour autant perturber leurs activités ou affecter les performances (Crutcher et al. 2021) (Xue et Xin 2016).

3. Mise en œuvre rapide

À la différence du provisionnement de ressources matérielles et du déploiement d'applications traditionnelles qui peuvent prendre des mois, voire des trimestres ou des semestres, la technologie du Cloud Computing présente l'avantage de la facilité et de la rapidité des déploiements des services. Étant donné que les ressources informatiques dans le Cloud Computing sont mises en commun par les fournisseurs de services en Cloud et fournies sous forme de services à la demande, pratiquement toutes ces ressources informatiques peuvent être approvisionnées rapidement (en quelques minutes).Cela offre une grande flexibilité aux clients du Cloud et élimine la pression de la planification des capacités. En effet, les consommateurs de Cloud Computing n'ont plus besoin de se procurer, d'installer et de mettre en service de nouvelles infrastructures et capacités de calcul avant de proposer de nouvelles applications ou de répondre à des charges de travail accrues. En revanche, ils peuvent facilement acheter les ressources informatiques nécessaires auprès de fournisseurs de services en Cloud, ce qui simplifie et accélère le cycle de déploiement des services et des applications en rendant possible la mise en place de systèmes entièrement fonctionnels en quelques minutes (Bauer et Adams 2012) (Crutcher et al. 2021).

4. Economie de Coûts

Le Cloud Computing est apparu comme une technologie rentable tant pour les particuliers et les entreprises. Il permet à leurs clients d'utiliser simplement des ressources informatiques fournies par les fournisseurs de services en Cloud qui prennent entièrement en charge les coûts d'investissement. De ce fait, le Cloud Computing évite les dépenses initiales substantielles et les complexités liées à l'achat des matériels et des logiciels, à l'hébergement physique du matériel et à la mise en place des centres de données. En effet, les particuliers et les entreprises n'ont plus besoin de consacrer des budgets importants et du temps au déploiement et à la maintenance de leur propre infrastructure, notamment du matériel complexe sur site.De plus, le modèle de tarification à l'usage, adopté par le Cloud Computing,fait que les utilisateurs ne paient que pour les ressources qu'ils utilisent activement, en augmentant ou en diminuant la capacité dynamiquement selon les besoins.Au-delà de l'efficacité et des économies considérables en termes de coûts d'exploitation et de gestion des ressources informatiques que génère le Cloud Computing, ce dernier offre un autre bénéfice économique résultant principalement de son flexibilité.Le Cloud Computing facilite l'accès et l'utilisation des ressources informatiques, permettant aux fournisseurs de services,aux entreprises particulièrement,de déployer des applications plus rapidement et d'augmenter la capacité de stockage et la puissance de calcul à la demande. En outre, il permet aux entreprises de s'adapter plus rapidement à l'évolution du marché et aux demandes des clients. Par conséquent, l'investissement dans les applications Cloud entraine une croissance plus rapide des revenus (Crutcher et al. 2021) (Syed et Baig 2013).

5. La mobilité

Depuis quelques années, le Cloud Computing est devenu une technologie incontournable dans le monde entier. Parmi ses avantages figure la puissante mobilité offerte. La mobilité du CloudComputing offre à ses utilisateurs la possibilité d'accéder facilement à tous les services et à

toutes les solutions disponibles à tout moment, en tout lieu et depuis n'importe quel appareil, à condition qu'ils disposent d'une connexion à l'Internet. Le Cloud Computing a permis de généraliser la collaboration entre les partenaires commerciaux, les collègues et les travailleurs à distance. La pandémie de la Covid-19 a provoqué, au cours des deux dernières années, un changement culturel massif en faveur du travail et des étudesà distance. Des entreprises et des universités du monde entier ont fermé leurs portes. Les employés et les enseignants travaillaient désormais à domicileet les étudiants suivaient leurs cours à distance, souvent pour des durées indéterminées.Dans le secteur de l'enseignement universitaire, la mobilité du Cloud Computing facilite la connectivité et l'échange entre lesgens (les étudiants, les enseignants, les chercheurs, etc.) qui peuvent bénéficier de la possibilité de mieux se connecter aux autres personnes et aux informations. Elle permet aux personnes d'accéder instantanément aux données, aux logiciels, aux réunions, aux conférences, aux cours, aux programmeset aux documentsessentiels, de n'importe où et à tout moment. Dans la même logique, le travail nomade et à distance au sein du secteur des affaires est également facilité par le déploiement du Cloud.Le Cloud Computing offre les avantages des postes de travail virtuels en fournissant aux employés travaillant à domicile un accès rapide, sécurisé et stable aux applications et aux données de l'entreprise, directement depuis leur domicile (XpertTechnologies 2018).

6. Productivité

Le Cloud Computing apporte une forte productivité à ses utilisateurs, notamment dans le secteur commercial. Dans le Cloud Computing,la mise en place et la gestion de l'infrastructure et des ressources informatiques, y compris la configuration du matériel, les correctifs logiciels, les mises à jour de sécurité, etc.,sont effectuées par les fournisseurs de services Cloud.Ceux-làdéveloppent et déploient une variété de servicessuivant différents modèles, à savoir IaaS, PaaS et SaaS,qui s'adressant à des milliers de clients. En exploitant lessolutions du Cloud Computing, les clients et les entreprises peuvent consacrer leur temps et leurs efforts à la réalisation de leurs objectifs commerciaux en choisissant simplement d'exploiter le reste en tant que service. En intégrant de nouvelles technologies, les fournisseurs de services Cloud proposent des outils modernes aux entreprises afin d'accroitre la rapidité d'exécution et d'automatiser leurs activités de gestion. En outre, le Cloud Computing a permis de généraliser la collaboration entre travailleurs à distance, collègues et partenaires commerciaux. Son utilisation offre également une grande flexibilité pour l'expansion et la croissance. Il est évident que ces facilités aident les entreprises à se développer rapidement, réduire l'espace disponible dans leurs bureaux, réaliser des gains de productivité avec une meilleure gestion du temps et augmenter leurs revenus. Par conséquent, l'informatique en nuage constitue une aubaine pour ses consommateurs, en offrant une productivité à tous les niveaux (Crutcher et al. 2021).

10. Défis de l'Adoption du Cloud Computing

Depuis son apparition, le phénomène du « Cloud Computing » ne cesse d'intéresser le secteur industriel ainsi que celui de la recherche dans le monde entier. L'informatique dématérialisée a offert d'énormes opportunités au secteur des technologies de l'information, menant à sa large adoption par l'industrie. Dans cette optique, elle doit être abordée correctement afin de concrétiser le Cloud dans toute son ampleur. Cependant, l'adoption actuelle des services Cloud est associée à de nombreux défis qui constituent des risques et des obstacles particuliers à son intégration, dont ceux présentés ci-dessous et pouvant être considérés comme étant une préoccupation majeure. Ces défis doivent être relevés soigneusement afin de fournir des services de haute qualité aux utilisateurs tout en respectant les exigences des fournisseurs de services.

10.1. Conception Centralisée

Le Cloud Computing est fondamentalement mis en œuvre dans de grands centres de données et exploité de manière centralisée. Cette conception présente l'avantage d'une économie d'échelle qui, en réduisant le coût total de possession, permet une fourniture moins chère de ressources informatiques aux clients. C'est pour soutenir la demande croissante de calcul et de stockage de données d'entreprise, en pleine croissance, et de l'industrie mondiale que sont généralement construits les grands Data Centers. Cependant, l'exploitation de grands centres de données présente un certain nombre d'inconvénients, notamment les suivantes:

- ✓ ***Point de Défaillance Unique***

Quelle que soit l'ingéniosité de sa conception, l'architecture centralisée du Cloud Computing présente un gros inconvénient, qui est le point de défaillance unique. En effet, elle est basée sur des centres de données centralisés qui rassemblent un grand nombre d'équipements informatiques et non-informatiques, répliqués pour assurer son fonctionnement correct. Ces centres de données représentent un point de défaillance unique pour le Cloud et peuvent être facilement affectés par des facteurs et phénomènes techniques, humains, climatiques et environnementaux.

Les équipements constituants des centres de données sont susceptibles de tomber en panne à tout moment, ce qui provoque l'instabilité, voire même le blocage, des centres de données. Ces pannes peuvent empêcher les utilisateurs d'accéder aux applications et éventuellement entraîner une perte de données.Par ailleurs, toutes les données précieuses et sensibles sont stockées et accessibles à partir des serveurs dans un réseau centralisé, ce qui augmente le risque pour la sécurité. Des pirates (personnes malveillantes) peuvent s'introduire dans les centres de données et accéder facilement à toutes ces données. De plus, les centres de données créent un goulot d'étranglement et ont tendance à présenter des problèmes d'évolutivité puisque la capacité des serveurs est limitée et ne peut pas supporter un trafic infini.En outre, ils ne sont pas en mesure d'offrir une protection absolue et garantie contre les événements catastrophiques tels que les incendies, les tempêtes, les ouragans, les tremblements de terre et les inondations. De plus, le "point de défaillance unique" peut être un fournisseur de services en nuage unique ou une organisation qui possède plusieurs centres de données. Outre les risques de « point de défaillance unique » évoqués ci-dessus, il est également possible que cette organisation fasse faillite.

L'apparition de l'un des problèmes ci-dessus peut conduire à la défaillance des centres de données, ainsi que tous les services qui y étaient hébergés. Il est important que les ressources nécessaires à la mise à disposition des services Cloud puissent pallier toute défaillance afin que les services restent continuellement disponibles pour les utilisateurs (BENAC et al. 2015).

- ✓ ***Coûts***

Pour le fournisseur de services Cloud, la création et l'exploitation des infrastructures centrales de Cloud Computing nécessitent des investissements notables. Ces infrastructures évoquent des images d'immenses centres de données où sont regroupés de grands nombres de serveurs, de racks remplis de processeurs, d'équipements de stockage et de communication, avec des stations d'alimentation et de refroidissement adjacentes ou proches. La construction des centres de données coûte approximativement de 5000 à 10000 euros par mètre carré, suivant le degré de redondance de leurs composants. En effet, la redondance est nécessaire pour maintenir un haut niveau de qualité de service et une grande disponibilité d'infrastructures. En plus des énormes dépenses initiales en immobilier et en matériel, les Data centers entraînent également des frais d'exploitation importants représentés par les coûts d'électricité, de refroidissement et de personnel technique. En outre, les dépenses d'entretien

des bâtiments et des aménagements intérieurs ainsi que celles de maintenance des équipements représentent un surcoût de plus de 5% par an de l'investissement initial. Selon les experts des secteurs IT et non-IT, les investissements liés à la construction d'un datacenter doublent tous les huit ans et varient fortement en fonction de la taille, des caractéristiques techniques et des acteurs impliqués dans les Data centers (BENAC et al. 2015). Par conséquent, une telle infrastructure de Cloud Computing est coûteuse à construire et reste une option viable uniquement pour les grandes entreprises comme Google, Amazon et Microsoft (Babaoglu et Marzolla 2014) (Kisembe et Jeberson 2017).

- ✓ ***Consommation d'Energie***

Pour maintenir son fonctionnement, le Cloud Computing se repose sur de grands centres de données, contenant les ressources matérielles du Cloud, notamment les serveurs d'application et de stockage et les périphériques réseau. Dans les centres de données, ces ressources consomment une quantité importante d'énergie selon leur niveau de puissance, même quand elles sont au repos. Elles doivent être fonctionnelles 24 heures sur 24, tous les jours de l'année, pour qu'elles puissent répondre aux demandes à tout moment et ne pas perdre les données des clients.Étant donné que les demandes des utilisateurs du Cloud ne cessent de croître, la consommation d'énergie des data centers a plus que doublé dans le monde au point de représenter 4% de la consommation énergétique mondiale en 2015,selon RTE (Réseau de transport d'électricité).En effet, la concentration d'équipements informatiques fait d'un datacenter un gros consommateur d'électricité qui est souvent comparable à la consommation nécessaire pour gérer une petite ville.La continuité et la stabilité électrique sont des facteurs critiques pour assurer l'alimentation des Data centers, même en cas de panne. Des spécialistes estiment que le secteur informatique devrait doubler ses besoins en énergie d'ici la prochaine décennie, en grande partie en raison des data centers, pour au final engloutir 11 % de la production électrique mondiale en 2030.En outre, les équipements informatiques sont des appareils électroniques de grande taille qui dégagent beaucoup de chaleur dans les Data centers où la température peut dépasser rapidement les seuils recommandés pour le bon fonctionnement de ces équipements. Le bon fonctionnement de ces derniers est assuré par des systèmes de refroidissement et de climatisation très énergivores (Djouhra 2016). Les systèmes de refroidissement et de ventilation consomment en moyenne environ 40% de l'énergie totale utilisée. Le fait que les centres de données consomment beaucoup d'énergie engendre la production de gaz à effet de serre (GES). Ces derniers contribuant au problème du réchauffement climatique. D'après Jones (2018), de 2 % des émissions mondiales de carbone sont représentés par l'écosystème des technologies de l'information et des communications (TIC) dans son ensemble, alors que les centres de donnéescontribuent pour 0,3 % environ aux émissions globales de carbone (Mengistu et Che 2019) (Zhang et al. 2010).

Le Cloud Computing vise principalement à fournir des services aux clients tout en augmentant leurs revenus et en optimisant la consommation d'énergie par ses infrastructures physiques. Par conséquent, les fournisseurs d'infrastructures subissent une pression énorme pour réduire la consommation d'énergie dans les centres de données. L'objectif n'est pas uniquement de réduire les coûts énergétiques, mais aussi de rendre les services informatiques écologiquement durables en respectant les réglementations gouvernementales et les normes environnementales (Panzieri et al. 2011) (Zhang et al. 2010).

- ✓ ***L'emplacement Géographique***

C'est sur des centres de données centralisés répartis géographiquement que repose le Cloud Computing. Par conséquent, il est possible que l'emplacement géographique de ces centressoit préférable pour les propriétaires.Cependant, il ne l'est pas forcément pour les clients. C'est le cas, par exemple, lorsque des restrictions sont imposées par les gouvernements sur les données sensibles qui

traversent les frontières nationales. Ainsi, il se peut qu'un centre de données se trouvant dans un pays donné soit interdit aux clients dans autres pays (Babaoglu et Marzolla 2014).

10.2. Sécurité et Confidentialité

Depuis l'avènement du Cloud Computing, la sécurité et la confidentialité ont été des préoccupations majeures. Selon une enquête menée par Gartner, elles ont joué le rôle le plus important dans l'obstruction du Cloud Computing. Le paradigme du Cloud Computing implique un approvisionnement à la demande avec des coûts d'investissement initiaux réduits dans l'infrastructure informatique ainsi que l'externalisation des ressources informatiques avec des capacités d'évolutivité des ressources consommables (Kahanwal et Singh 2012). Il repose essentiellement sur les technologies de la virtualisation et de la multi-location. Outre les risques liés à la sécurité de l'information encourus par l'architecture informatique traditionnelle, le concept du Cloud ainsi les technologies associées ont introduit de nouveaux risques et des vulnérabilités de sécurité spécifiques au Cloud Computing.En effet, en confiant les données critiques à un fournisseur de services qui essaie de les protéger d'autres sources intérieures ou extérieures, un utilisateur prend des risques sur la disponibilité, la confidentialité et l'intégrité de ses données. Ces dernières sont jugées comme les trois attributs fondamentaux pour répondre aux exigences de la sécurité. Dans le Cloud Computing, les ressources informatiques sont mises en commun (mutualisation) et fournies aux clients en tant que services à la demande par des interfaces de gestion basées sur le web, ce qui crée de nouvelles surfaces d'attaque et provoque la probabilité des risques confidentiels. Les utilisateurs du Cloud veulent garantir le fait que leurs données critiques ne soient pas visibles, accessibles et utilisées illégalement par des utilisateurs non autorisés, même par les fournisseurs du Cloud. De plus, la disponibilité est une exigence majeure dans le Cloud qui peut être affectée si les données de l'utilisateur sont indisponibles au moment où il en a besoin. Etant donné que les données sont la base de la fourniture de services de Cloud Computing, le maintien de leur intégrité est une tâche fondamentale. L'intégrité peut être affectée si les données des utilisateurs sont perdues ou endommagées à la suite d'une activité malveillante ou par inadvertance (Zhang 2021).

Les risques de sécurité majeurs dans le Cloud peuvent être classés en trois catégories, à savoir : «Menaces provenant de l'hôte (hyperviseur)», «Menaces provenant des MVs» et «Menaces entre clients et Cloud (centre de données)». Dans la première catégorie, l'hôte est un logiciel qui contrôle la couche entre le matériel et les systèmes d'exploitation. Dans les systèmes de Cloud Computing, toutes les communications avec les MVs hébergées doivent passer par la couche de l'hôte. Pour cette raison, les attaques sur cette couche sont attrayantes pour les pirates qui peuvent prendre le contrôle de cette couche et de toutes les machines virtuelles hébergées s'ils peuvent injecter, installer et exécuter leurs logiciels malveillants. Concernant la deuxième catégorie, les serveurs informatiques du Cloud peuvent contenir des dizaines de machines virtuelles qui sont autorisées à communiquer et partager des ressources entre elles. Cela peut introduire des risques de sécurité pour chaque machine virtuelle et la rendre vulnérable aux attaques à cause de facteurs suivants : l'infrastructure de communication partagée, le réseau virtuel et la mauvaise configuration de sécurité. Concernant la troisième catégorie, le Cloud Computing s'appuie sur l'Internet et des serveurs distants afin de conserver les données qui peuvent être accédées ou gérées par le biais des services fournis par les fournisseurs de services Cloud. Les services du Cloud sont accessibles via l'Internet en utilisant des mécanismes et des protocoles d'Internet standards afin de transmettre des données ou des applications entre les clients et le Cloud. En effet, les données en transit peuvent être la cible de plusieurs attaques malveillantes (Jathanna et Jagli 2017) (Jouini et Rabai 2015) (Rao et Selvamani 2015). Parmi les principales menaces, on peut citer les attaques par inondation, le déni de service, la perte ou la fuite de données, les insiders

malveillants, le détournement de compte, de service et de trafic, l'abus et l'utilisation néfaste du Cloud Computing, la programmation d'applications non sécurisées, etc. Par ailleurs, ENISA a élaboré un rapport à la fin de l'année 2009 dans lequel il a identifié les neuf principaux risques de sécurisé, à savoir la perte de maîtrise de gouvernance, l'isolement des environnements et des données, les risques de conformité, la publication des interfaces de gestion, la protection des données, la suppression dangereuse ou incomplète des données et les utilisateurs malveillants (Zayed et al. 2015).

La sécurité et la confidentialité dans l'informatique dématérialisée nécessitent une sérieuse remise en question et devraient être un souci pour tous, à savoir les fournisseurs, les prestataires et les utilisateurs. Les risques énoncés ci-dessus peuvent être atténués en développant et en adoptant des mesures, des mécanismes et des normes concrètes pour protéger les environnements de Cloud Computing. Ce dernier doitaller plus loin pour une gestion stricte des mots de passe et des privilèges de connexion en pensant à la sécurité en termes d'utilisation et de types de données. Plus ces données sont confidentielles, plus la sécurité doit être élevée et plus le choix du type de Cloud est critique et crucial. Dans cette optique, un modèle de sécurité des données comprenant l'authentification, le cryptage et l'intégrité des données, la récupération des données et la protection des utilisateurs doit être conçu pour améliorer la sécurité et la confidentialité des données dans le Cloud (Jathanna et Jagli 2017) (Rao et Selvamani 2015).

10.3. Disponibilité

L'un des défis croissants et récurrents dans les systèmes distribués à forte intensité logicielle, dont le Cloud Computing, est la disponibilité.Les services basés sur le Cloud apportent une réduction considérable des investissements informatiques initiaux et une grande flexibilité en adoptant une politique de tarification utilitaire. Cependant, l'un des obstacles majeurs est la disponibilité en tant que principale propriété interne de la fiabilité, car elle empêche que les clients décident la possibilité de passer au Cloud Computing, et ce, qu'ils soient personnels ou commerciaux (Siebenhaar et al. 2013). La disponibilité reflète, dans son sens le plus large, l'aptitude d'un composant ou du système à remplir ses fonctions requises de manière continue et correcte, indépendamment de tout facteur de défaillance. La disponibilité est définie, dans le cadre du Cloud Computing, comme étant la probabilité que le service soit, à un instant t, utilisable, opérationnel et accessible facilement sur demande par une entité autorisée (Critchley 2014) (Yanovsky et al. 2016). Elle fait référence au temps de disponibilité du système, du réseau de système, des logiciels et des matériels qui fournissent le service désiré de manière collective pendant son utilisation. La disponibilité est considérée par les utilisateurs de services Cloud Computing comme étant une exigence importante qui doit être satisfaite. Ces utilisateurs veulent que les solutions fonctionnent de manière permanente, surtout en cas de services critiques. Les fournisseurs de services Cloud ont pris l'initiative, dans cette optique, de fournir des services rentables et disponibles.En revanche, ces solutions ne fonctionnent forcément comme elles sont censées le faire. les systèmes Cloud et leurs services peuvent, en effet, faire face à des pannes inattendues ou planifiées à cause de différents incidents, dont les pannes matérielles, les pannes d'électricité de longue durée ou de courte durée, les erreurs de congestion, la Cyber-attaque, les bogues logiciels, les litiges administratifs ou juridiques et les erreurs humaines (Adekunle et Osimosu 2013).Ces pannes, entraînant principalement une perturbation ou une perte d'utilisation et d'accès des services, ont un impact sérieux sur la disponibilité des services du Cloud.Ceci a remis en question la fiabilité des environnements du Cloud.Une perte de revenus, de productivité et de réputation peut être entrainée par l'indisponibilité des services Cloud, ce qui engendre une dégradation des performances, voire l'écroulement de l'ensemble du système de Cloud Computing (Critchley 2014). Nous pouvons nous attendre, dans l'avenir, à des services plus riches et à un plus grand nombre de fournisseurs de Cloud Computing au vu de leur importance croissante dans nos vies.Les efforts des fournisseurs des

services Cloud doivent être concentrés sur la résilience et la robustesse des infrastructures Cloud face aux temps d'arrêt intentionnels et accidentels afin que l'utilisateur puisse optimiser les avantages qu'il en retire. Il est essentiel de fournir des services disponibles et fiables dans le Cloud Computing afin de maintenir la satisfaction, la confiance et la fidélité des clients et d'éviter les pertes de revenus (Mesbahi et al. 2018). Les fournisseurs de Cloud doivent collaborer avec les clients du Cloud afin de comprendre leurs besoins particuliers de manière claire et déterminer le niveau de disponibilité souhaité ou la criticité pour planifier des solutions Cloud dont le but est de garantir leurs exigences (Puthal et al. 2015).Ils sont tenus, par ailleurs, de comprendre et étudier les menaces à la disponibilité des services dans le Cloud Computing. C'est ce qui permettra d'élaborer de bonnes pratiques pour les corriger après les avoir détectées et déterminées.Ceci signifie que les fournisseurs de services Cloud doivent être aptes à proposer des idées novatrices et élaborer des techniques systématiquesd'atténuation, l'objectif escompté étant de garantirla disponibilité constante et la continuité des services Cloud et de minimiser les temps d'arrêt.

10.4. Interopérabilité et portabilité

Le Cloud Computing ne cesse, à ce jour, de multiplier ses offres.La concurrence entre ses fournisseurs de services est en pleine effervescence. En revanche, l'interopérabilité et la portabilité sont des défis auxquels cette technologie se trouve encore confrontée, ce qui empêche ses utilisateurs de profiter du véritable gain du Cloud Computing. La portabilité et l'interopérabilité, souvent utilisés de manière interchangeable, sont deux concepts étroitement liés. L'interopérabilité désigne, dans le paysage du Cloud Computing, la capacité de migration, de déploiement, d'intégration et de gestion transparente des charges de travail d'applications sur diverses ressources logicielles et matérielles fournies par plusieurs fournisseurs Cloud, tels que GoGrid et Amazon. La portabilité dans le Cloud reflète, par ailleurs, la capacité de déplacement d'une infrastructure sur site à un Cloud public ou privé, ou des applications ou des données d'un fournisseur de Cloud à un autre, à mesure que les données restent intactes et non corrompues et que l'application reste exécutable et utilisable (Kaur et Singh, 2017) (Ranjan 2014) (Sajid et Raza 2013).

Le Cloud Computing se trouve tujours incapable d'assurer les conditions d'utilisation des services aux clients.Un fournisseur Cloud n'est toujours pas en mesure de répondre à toutes les exigences de ses clients. Ainsi, ces derniers essaient souvent de passer à d'autres fournisseurs, principalement pour des considérations techniques, économiques et juridiques, ou d'intégrer des services supplémentaires provenant de fournisseurs disparates.Toutefois, ces actes s'avèrent fastidieux dans la majorité des cas. Les clients peuvent ainsi se trouver confrontés à de nombreux problèmes.Les principaux obstacles qui freinent la concrétisation de la portabilité et de l'interopérabilité résident, en effet, dans les risques de verrouillage et de dépendance aux fournisseurssusceptibles d'être nuisibles aux objectifs recherchés par les clients et les entreprises dans leur migration vers le Cloud. L'absence d'API et des protocoles ouverts, la disparité des technologies et des langages de programmation et le manque d'interfaces standards sont, entre autres, les autres facteurs pouvant entraver la réalisation desdits objectifs (Kaur et Singh, 2017) (Ranjan 2014) (Maaref 2012).

De ce fait, il est nécessaire que les acteurs industriels et académiques concernés combinent et multiplient leurs efforts afin d'améliorer la portabilité et l'interopérabilité, le but étant de permettre l'ouverture et la flexibilité du Cloud Computing. Il faut que les utilisateurs de l'informatique dématérialisée puissent être en mesure de migrer à l'extérieur et à l'intérieur des Cloud de changer de Cloud à n'importe quel moment.Cela signifie qu'ils doivent pouvoir choisir n'importe quelle solution d'un fournisseur de service Cloud de manière libre, en étant sûr qu'elle pourra s'intégrer à d'autres solutions de Cloud Computing sans que son fonctionnement soit interrompu, et que celui-ci soit

assuré avec des rentabilités et des performances maximales (Dillon et Chang 2010) (Sajid et Raza 2013).

10.5 Performance

Le succès du déploiement et de l'adoption du Cloud Computing dépend fortement de la mise en œuvre de solutions performantes, qui est considérée comme l'un de ses principaux enjeux. La performance est un facteur important lorsque les utilisateurs s'orientent vers des solutions basées sur le Cloud. De nos jours, la performance dépasse le cadre d'un concept classique et intègre des concepts fondamentaux plus étendus tels que la fiabilité, l'efficacité énergétique et l'évolutivité. Elle désigne généralement les capacités des applications exécutées sur les systèmes en Cloud. En effet, les performances du Cloud Computing et de ses ressources peuvent être affectées par plusieurs facteurs. La majorité des applications Cloud comptent un grand nombre d'utilisateurs, même supérieur à leurs capacités nominales, ce qui entraine une augmentation de la quantité d'informations et de données à transférer ainsi qu'une charge élevée sur le matériel et les logiciels. Ces applications à forte intensité de données et de transactions sont plus difficiles à supporter l'évolutivité et à fournir les ressources appropriées. Le manque de ressources appropriées, à savoir l'espace disque, la bande passante limitée, la vitesse de l'unité centrale inférieure, la mémoire, mes connexions réseau, etc., entraîne des performances médiocres des applications. De plus, des performances inadéquates sont parfois le résultat d'une architecture d'application qui n'est pas en mesure de bien distribuer ses processus sur les ressources du Cloud disponibles. Par ailleurs, les utilisateurs préfèrent souvent utiliser des services de plus d'un Cloud. L'emplacement des utilisateurs et leur distance par rapport à l'emplacement des centres de données des fournisseurs de services Cloud sont également un facteur important qui peut affecter les performances. Les utilisateurs qui se trouvent éloignés des fournisseurs de services de Cloud Computing peuvent subir des latences et des retards élevés. Les latences peuvent être aussi le produit d'un équilibrage de charge inefficace qui se traduit par le fait que le serveur n'est pas capable de distribuer efficacement le trafic entrant en vue de fournir la meilleure expérience utilisateur. De plus, la tolérance aux pannes est un facteur ayant un impact particulier sur les performances du Cloud Computing. La défaillance d'un ou de plusieurs composants dans les environnements Cloud empêche parfois les opérations de se poursuivre comme prévu, conduisant à la détérioration des performances. Dans le même contexte, en présence d'erreurs, de défaillances et de pertes de données dans le Cloud, le temps nécessaire à leur récupération et les volumes de données récupérables peuvent favoriser la dégradation des performances et diminuer la satisfaction des clients. La satisfaction des clients est atteinte lorsque les « accords de niveau de service » discutés avec les fournisseurs de services en Cloud remplissent parfaitement leurs demandes attendues et leurs exigences en matière de performance (Ahmed et al. 2017) (Khanghahi et Ravanmehr 2013) (Xiong 2018). Les performances du Cloud peuvent également être affectées par d'autres facteurs, tels que la disponibilité, la sécurité, la facilité d'utilisation, l'évolutivité, la capacité de mémoire tampon, la redondance et la puissance du processeur.

Les problèmes des faibles performances dans les environnements Cloud peuvent effectivement mettre fin à la prestation d'un service et entrainerla perte des clients, la réduction de la productivité des employés, la réduction des revenus, etc. Les utilisateurs doivent s'assurer que les performances des applications sont optimisées lorsqu'elles sont déplacées vers une infrastructure de Cloud Computing.Toute solution potentielle visant à surmonter les obstacles des performances doit être soigneusement analysée afin de vérifier son efficacité dans les situations réelles. Les experts en performance devraient se pencher sur les transactions techniques des services de Cloud Computing sous-jacents avant de conseiller les utilisateurs et les fournisseurs de Cloud pour ces services (Sajid et Raza 2013).

11. Vers une Architecture Cloud Computing basé sur Pair-à-Pair

Le Cloud Computing est considéré comme étant l'un des principales tendances dans le domaine des technologies de l'information. Il est qualifié de système distribué à grande échelle qui est révolutionnaire pour la prochaine génération informatique. Le Cloud Computing est généralement hébergé dans de grands centres de données et géré de manière centralisé. L'architecture centralisée du Cloud Computing a suscité des inquiétudes quant à son succès en raison des défis qu'elle pose (section 9.1.). Ces défis ont été présentés comme des obstacles susceptibles de favoriser sa chute en termes de progrès et d'intérêts sur les plans universitaire et commercial. Par conséquent, des recherches récentes ont étudié une stratégie différente de conception du Cloud Computing sans aucune installation centralisée en utilisant des technologies P2P. Ce mariage de technologies peut être considéré comme une « démocratisation » du Cloud Computing qui devient rapidement le sujet le plus brûlant dans le domaine de l'informatique. Une architecture Cloud Computing basée sur un système Pair-à-Pair vise à fournir une infrastructure Cloud évolutive à des coûts réduits en exploitant autant que possible les ressources gratuites des pairs. Par conséquent, les services informatiques actuels passent des services Internet traditionnels au Cloud basé sur P2P, ce qui donne lieu à une vaste gamme d'applications, de recherches et d'études scientifiques qui envisagent le couplage du Cloud Computing et des systèmes Pair-à-Pair comme moyen pour bénéficier de leurs avantages et surmonter certains des problèmes posés par les deux paradigmes (Jo et Han 2018) (Kavalionak et Montresor 2012).

11.1. Présentation du Cloud Computing basé sur Pair-à-Pair

Une architecture Cloud Computing basée sur un système Pair-à-Pair (en anglais Cloud Computing based on Peer-To-Peer, abrégé CC-P2P) se réfère à une architecture Cloud totalement décentralisée constituée d'un nombre de nœuds individuels (des milliers ou des millions de nœuds), répartis dans le monde entier et interconnectés par un réseau de communication public comme l'Internet. Au sein d'une telle architecture, les nœuds individuels distribués géographiquement sont assemblés sans aucune unité centrale de surveillance ou un composant de gestion et de coordination. A la différence d'un Cloud centralisé, le CC-P2P pourrait être construit à partir des ressources informatiques de stockage et de communication existantes avec un investissement initial pratiquement nul. Les périphériques susceptibles de participer à une architecture CC-P2P incluent des modems à large bande, des routeurs, des décodeurs, des consoles de jeux, des ordinateurs de bureau et des PC, étant donné que la majorité d'entre eux offrent une connectivité Internet, des capacités de calcul raisonnables et certaines capacités de stockage. Ces ressources ordinaires sont transformées en une infrastructure de calcul qui peut être utilisée par des clients. Le CC-P2P fournit un système distribué permettant l'approvisionnement des ressources à un coût faible ou nul, de sorte que le système puisse s'auto-assembler et s'autogérer. De nombreuses applications peuvent bénéficier de ce modèle technologique mixte, notamment celles desauvegarde, de stockage, de calculs parallèles, de collaboration, de partage et de distribution de contenu, de diffusion multimédia en continu « streaming », de jeux en ligne, etc (Babaoglu et Marzolla 2014) (Kisembe et Jeberson 2017) (Panzieri et al. 2011).

11.2. Pourquoi un Cloud Computing basé sur Pair-à-Pair

Le système P2P fait office, dans un CC-P2P, de back-end pour la technologie du Cloud Computing en lui offrant une architecture décentralisée héritant ses caractéristiques fondamentales. L'architecture CloudP2P est basée principalement sur l'absence d'une structure centralisée. Une telle architecture se base sur une approche de communication décentralisée capable d'éliminer le point unique d'attaque et de défaillance. En effet, elle ne contient pas d'entité unique la possédant, la contrôlant et sur laquelle

repose la fonctionnalité du système dans son ensemble. D'autre part, il est possible que la création et l'exploitation d'un CC-P2P se fassent dans le cadre d'un effort local, le consentement ou l'autorisation d'une autorité particulière n'étant pas nécessaires. En effet, en installant le logiciel approprié sur leur machine locale, les clients peuvent participer volontairement au CC-P2P. Un plus grand nombre de ressources gratuites et hétérogènes peut ainsi être disponible sur le réseau. Certaines des ressources disponibles, dont les données, les programmes et les informations, peuvent être catégorisées comme contenu, de même que différentes capacités, notamment le stockage, la mémoire, les processeurs et la bande passante. La valeur de l'infrastructure du Cloud résultante serait, par conséquent, proportionnelle à la nature et au nombre des ressources et des pairs qui y contribuent. Par ailleurs, les pairs participants au CC-P2P peuvent, en cas de besoin, se fournir mutuellement des ressources. Ceci participe à la réduction de la latence du réseau, notamment pour les applications interactives. Lorsque les coûts d'utilisation et de déploiement sont réduits, le CC-P2P offre aussi les avantages des commodités informatiques du Cloud Computing, combinés à la flexibilité et l'évolutivité d'un système P2P, devenant ainsi une solution viable à très faible coût pour l'informatique (Babaoglu et Marzolla 2014) (Kisembe et Jeberson 2017) (Mayer et al. 2013) (Mengistu et Che 2019) (Panzieri et al. 2011).

Sur un plan non informatique, des avantages sont également offerts par l'architecture CC-P2P du fait que ses composants sont petits et bien répartis géographiquement, sa connectivité réseau est garantie par différents fournisseurs de services Internet et son alimentation électrique est assurée par différents réseaux électriques. Ainsi, la consommation d'énergie et le risque de point de défaillance unique associé aux centres de données sont ainsi réduits considérablement par une telle architecture. De plus, cette dernière élimine complètementles problèmes de dissipation thermique des centres de données ainsi que le besoin d'une infrastructure de refroidissement dédiée. Par conséquent, le CC-P2P représente une solution économique et écologiquepour l'industrie des technologies de l'information et de la communication (Babaoglu et Marzolla 2014) (Mengistu et Che 2019).

11.3. Cloud Computing basé sur Pair-à-Pair VS Volunteer Computing

Le calcul volontaire « en anglais Volunteer Computing, abrégé VC » est un type de calcul distribué qui exploite les ressources informatiques inutilisées offertes par des volontaires. Ces derniers peuvent être des membres du grand public, des organisations ou des entreprises qui possèdent des dispositifs généralement connectés à l'Internet, tels que les ordinateurs de bureau et les routeurs. Le VC permet l'agrégation des ressources informatiques des dispositifs volontaires pour fournir l'infrastructure informatique nécessaire à l'exécution des tâches à forte intensité de calcul et de stockage. Il fait partie d'un effort permanent visant à trouver une infrastructure informatique alternative relativement peu coûteuse et plus écologique pour les applications, et à favoriser l'utilisation optimisée des ressources informatiques. Le premier projet de calcul volontaire a été GIMPS (Great Internet Mersenne Prime Search), lancé en 1996 dans le but de trouver le plus grand nombre premier. Il a été suivi par plusieurs autres projets, dont nous citons distributed.net, SETI@home et Folding@home. Depuis sa création, le calcul volontaire a été appliqué avec succès dans de nombreuses applications et projets, comprenant la résolution de problèmes scientifiques, l'astronomie, la vie intelligente, la gestion des urgences, etc.

Le calcul volontaire est un paradigme informatique bien connu, dans lequel les utilisateurs installent une application spécifique sur leurs machines personnelles afin d'exécuter des applications au profil d'autres personnes. Via l'application spécifique, chaque utilisateur récupère et traite les données d'entrée à partir d'un site central et lui renvoie les résultats. Pour y parvenir, un système middleware est généralement utilisé comme intermédiaire. Il fournit également les différentes fonctionnalités du système, notamment la répartition des tâches volontaires, l'ordonnancement optimal des tâches et des ressources, la modélisation de la communication entre les nœuds, la fourniture de services de

confiance et de sécurité, l'octroi de mécanismes permettant d'attirer et de retenir les volontaires, etc. L' infrastructure ouverte de Berkeley pour l'informatique en réseau (en anglais, Berkeley Open Infrastructure for Network Computing abrégé BOINC) est le système middleware complet le plus utilisé.

Un CC-P2P se distingue du VC par le fait qu'il ne repose pas, en général, sur des répertoires ou des composants de coordination centralisés. Comme toute application P2P, le CC-P2P décrit des systèmes dans lesquels les ressources sont échangées entre les "pairs" du réseau directement, sans l'intervention d'un serveur central. En revanche, il n'y a généralement pas de communication entre pairs dans les VC (Mengistu et Che 2019) (Panzieri et al. 2011).

12. Conclusion

La facilité d'utilisation et les progrès technologiques ont fait du Cloud Computing un paradigme incontournable pour la fourniture et la gestion de services sur l'Internet. Dans ce chapitre, nous sommes efforcés de clarifier les concepts de base du Cloud Computing. D'abord, nous avons présenté l'historique du Cloud Computing qui a conduit à son émergence. Ensuite, nous avons introduit et clarifié la définition de référence du Cloud Computing proposée par le NIST. Dans la mesure où le Cloud Computing est le fruit de l'évolution de certaines technologies antérieures, celles-ci ont été présentées et comparées par le Cloud Computing. Par ailleurs, nous avons apporté un éclairage sur les composants de base des architectures Cloud Computing ainsi que leurs principales caractéristiques. Nous avons examiné par la suite les modèles de services et ceux de déploiement du Cloud Computing. En outre, nous avons abordé les avantages scintillants de divers services basés sur le Cloud. Ensuite, nous avons présenté en détails les principaux défis et challenges qui pèsent sur les environnements Cloud Computing. Enfin, nous avons mis en évidence une perspective prometteuse pour l'avenir du Cloud Computing, visant à résoudre les problèmes associés à la centralisation. Cette voie de recherche a envisagé une vision différente de la conception du Cloud Computing fondée sur les technologies Pair-à-Pair, en vue de bénéficier des avantages et de surmonter certains des problèmes liés à ces deux paradigmes. Cependant, l'évolution de l'architecture CC-P2P est conditionnée par la nécessité d'accorder une attention particulière aux défis hérités. Cela signifie que cette nouvelle architecture doit maintenir les principales caractéristiques du «Cloud Computing» afin de construire un environnement plus complet et plus fiable. L'étude menée dans ce chapitre permet de mieux comprendre, dans le chapitre suivant, la disponibilité des services en tant que défi majeur du Cloud Computing, ainsi que les mécanismes et les solutions visant à élever cette exigence.

Chapitre 4 : Disponibilité des Services dans le Cloud Computing

Sommaire

1. Introduction
2. Unités et Facteurs de Disponibilité
3. Définition de la Disponibilité
4. Taux de Disponibilité
5. Mesure de la Disponibilité du Service
6. Les Coûts de l'Indisponibilité
7. Mécanismes de Disponibilité dans le Cloud Computing
   7.1. Equilibrage de Charge
   7.2. Tolérance aux Pannes
   7.3. Redondance
   7.4. Réplication des données
8. Solutions pour la Disponibilité dans le Cloud Computing
9. Conclusion

1. Introduction

La disponibilité permanente est une propriété interne principale de la fiabilité. C'est celle qui est la plus couramment discutée dans la plupart des systèmes informatiques, en particulier le Cloud Computing. Etant une technologie en plein essor, le « Cloud Computing » est de plus en plus utilisé pour héberger de nombreuses applications commerciales ou d'entreprise. Cependant, l'utilisation intensive des services Cloud entraîne un problème de disponibilité de ces derniers, tant pour les fournisseurs que pour les utilisateurs. Le fait d'assurer la disponibilité permanente des services dans le Cloud Computing demeure une préoccupation majeure pour les fournisseurs de services Cloud et l'un de leurs plus grands défis. Des efforts sont fournis dans ce sens par les chercheurs ainsi que les fournisseurs de services Cloud lors de l'édition des livres blancs et des documents techniques. Nous présenterons dans ce chapitre un état de l'art relatif à la disponibilité des services dans le Cloud, ceci étant un axe de notre recherche pour la thèse. Nous aborderons d'abord, dans la section 2, les principales unités et principaux facteurs d'interruption des services dans le Cloud. Nous définirons, dans la section 3, la disponibilité des services dans le Cloud Computing. Dans la section 4 seront exposés les taux de disponibilité des services standards dans les environnements Cloud. Suite à quoi nous introduirons, dans la section 5, les mesures liées à la disponibilité des services et les méthodes de base pour leur calcul. Par ailleurs, l'impact de l'indisponibilité des services dans le Cloud sera abordé en termes de coûts dans la section 6. Ensuite, un aperçu des principaux mécanismes dédiés à l'amélioration de la disponibilité dans le Cloud Computing sera établi dans la section 7 qui synthétise notamment leurs avantages et leurs faiblesses. Enfin, nous présenterons dans la section 8 les principales solutions et travaux de recherche qui traitent du problème de la disponibilité dans le Cloud Computing.

2. Unités et Facteurs de Disponibilité

La disponibilité dans le Cloud Computing implique la mise en place de solutions qui soient capables de pénétrer tous les niveaux participant à la fourniture de services Cloud. En effet, chacun de ces niveaux ainsi que leurs éléments ont leurs vulnérabilités et sont susceptibles de subir des défaillances (Bauer and Adams 2012). Le laboratoire Bell (en anglais, Bell Labs) a développé un modèle complet et exhaustif consistant en un cadre de huit ingrédients (abrégé cadre 8i) utile pour considérer toutes les vulnérabilités éventuelles d'un système (Rauscher et al. 2006). Ces huit ingrédients sont cruciaux dans le fonctionnement réel de systèmes tels que le Cloud Computing et incluent : 1) le matériel, 2) le logiciel, 3) l'alimentation, 4) l'environnement, 5) le réseau, 6) la charge utile (payload), 7) l'humain, et 8) la politique.

Un système est formellement défini par le système de mesure de la qualité pour l'industrie de l'information et de la communication «TL 9000» comme étant « un ensemble d'éléments matériels et/ou logiciels situés à un ou plusieurs emplacements physiques où tous les éléments sont nécessaires au bon fonctionnement». Dans ce sens, un système Cloud est physiquement hébergé dans de grands centres de données construits à partir du *matériel* qui héberge les *logiciels* d'application et de plate-forme. Le matériel se réfère à un ensemble de dispositifs physiques constitué des équipements électroniques et physiques qui composent le système. Les logiciels permettent de contrôler les matériels physiques du système et de fournir des services de qualité aux utilisateurs. Le *matériel* dépend directement de *l'alimentation* électrique appropriée pour alimenter tout dispositif physique et d'un *environnement* de fonctionnement approprié. En effet, les installations matérielles sont sensibles aux conditions environnementales ambiantes souvent difficiles, notamment la température, l'humidité, la poussière et les gaz corrosifs. Les systèmes Cloud sont intégrés dans un contexte de solution en

réseau via lequel ils interagissent avec les utilisateurs et d'autres systèmes transportant des *charges utiles* d'applications structurées selon des protocoles standards ou propriétaires. Les systèmes sont pris en charge par des ingénieurs de maintenance ou des opérateurs *humains* qui suivent un ensemble de procédures et de *politiques*. Chacun de ces ingrédients est sujet à des erreurs et des défaillances qui pourraient perturber ou interrompre leur fonctionnement (Rauscher et al. 2006). Par conséquent, la disponibilité du système Cloud Computing, ainsi que ses services, peut être affectée par des pannes/interruptions planifiées ou non planifiées. L'interruption désigne simplement l'indisponibilité temporaire de la fourniture d'un service ou du réseau de distribution à un client, c'est-à-dire la période pendant laquelle le système ou le service est arrêté de façon à ne pas pouvoir exécuter et fournir les fonctions prévues à ses utilisateurs. Une étude sur le taux de temps d'arrêt, réalisée par *Strategic Research*, a révélé que 13 % des temps d'arrêt des services sont des pannes/ interruptions non planifiées et que 87 % sont des pannes/ interruptions planifiées (Critchley 2014).

- ***Les Interruptions non Planifiées***
  Une interruption non planifiée représente le temps d'arrêt qui se produit pendant les heures de fonctionnement prévues du système ou du service Cloud. Elle résulte des défaillances imprévues et incontrôlables du système, associées à des défauts survenant dans les composants matériels ou logiciels. Une interruption non planifiée du système Cloud peut être causée par plusieurs facteurs internes ou externes, notamment des défauts de conception et de fabrication des composants matériels, des erreurs de programmation, de spécification et d'intégration des composants logiciels, des erreurs humaines de manipulation et de maintenance, des cyber-attaques, l'usure des composants, des catastrophes naturelles, etc (Vargas 2000) (Rauscher et al. 2006). La figure 4.1 montre les causes les plus courantes des temps d'arrêt non planifiés. A partir d'enquêtes menées par des analystes du secteur informatique Gartner/Dataquest, il a été conclu que les défaillances logicielles représentent 27 % des causes (Marcus et Stern 2003). Les interruptions non planifiées sont coûteuses. Elles doivent être évitées ou minimisées par la mise en place de mesures de détection et d'atténuation intégrées dans la conception et le fonctionnement du système (Vargas 2000).

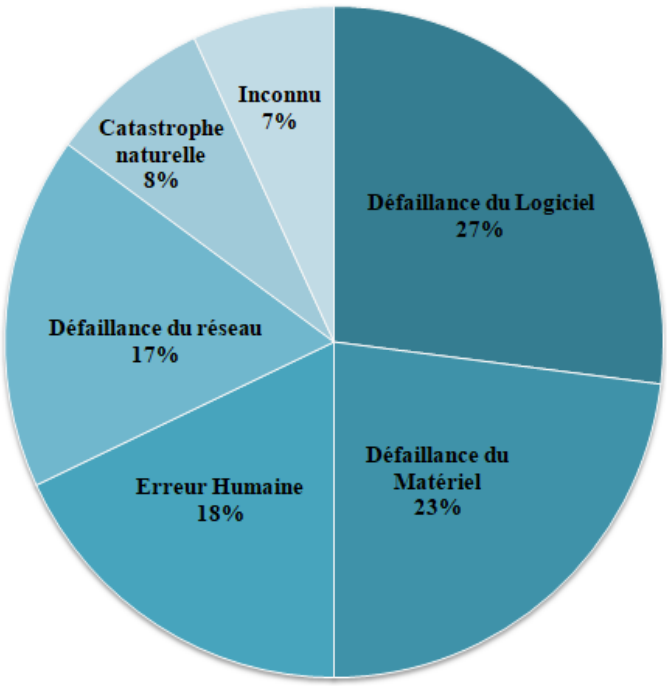


Figure 4. 1: Les Principales Causes des Interruptions non Planifiées.

- ***Les Interruptions Planifiées***
  Une interruption planifiée représente le temps pendant lequel le système doit être arrêté suite à des événements de correctifs et de maintenance tournant autour d'opérations de réparation, de sauvegarde ou de mise à niveau des applications, des logiciels ou du matériel. Elle constitue la principale source d'indisponibilité des services et provoque parfois des interruptions non planifiées lorsqu'elle est exécutée de manière incorrecte. Par conséquent, les interruptions planifiées doivent être bien programmées pour que leur impact sur la disponibilité du système et des services soit minimal (Vargas 2000). La figure 4.2 montre les causes les plus courantes des temps d'arrêt non planifiés.

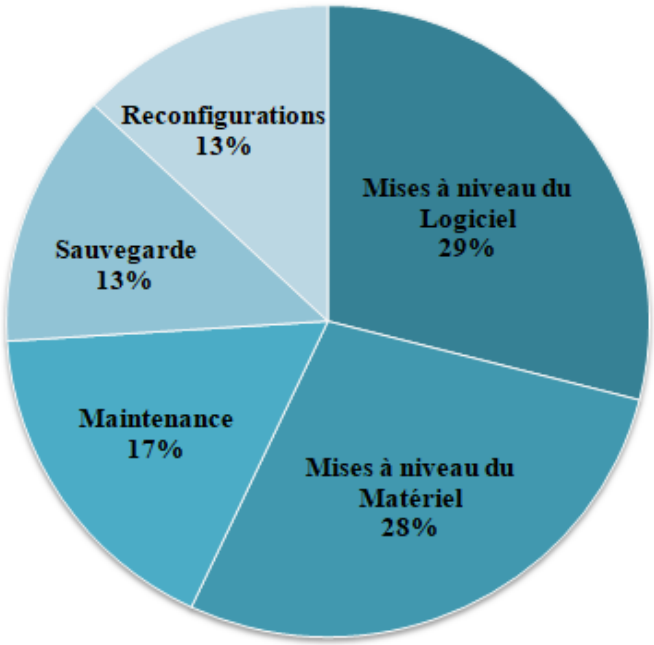


Figure 4. 2: Les Principales Causes des Interruptions Planifiées.

3. Définition de la Disponibilité

En informatique, la disponibilité désigne la capacité d'un utilisateur à accéder efficacement à des informations, à des ressources, à un service ou à un système à un emplacement spécifique et au moment où il en a besoin. Le système de mesure de la qualité pour l'industrie de l'information et de la communication «TL 9000» a développé des mesures sophistiquées de la disponibilité des services qui ont permis de fournir une définition simple de la disponibilité adaptée à de nombreuses applications d'entreprise. Dans l'industrie des télécommunications, la disponibilité du service est formellement définie comme (TL 9000) :

> *« L'aptitude d'une unité à être prête à exécuter une fonction requise pour toute période dans un intervalle de temps donné, ou à un instant donné, en supposant que les ressources externes sont fournies si nécessaire. »*

Traditionnellement, c'est le système ou le nœud du système qui est l'unité par rapport à laquelle la disponibilité du service est normalisée. En ce sens, la disponibilité reflète la capacité d'un système, d'un réseau, d'une entité de système comprenant tous les composants matériels et/ou logiciels pertinents, à remplir leurs fonctions de manière continue dans le but de fournir individuellement ou collectivement le service requis (Bauer and Adams 2012).

Alignée majoritairement sur la «TL 9000», notre recherche documentaire a révélé une variété de définitions de la disponibilité des services qui varient encore plus dans le contexte des services Cloud. Nous présentons ainsi de nombreuses définitions de la notion de disponibilité, dont chacune est adaptée à une solution proposée. Ces définitions couvrent divers aspects de la disponibilité. Les termes utilisés sont similaires mais ont des interprétations et des mesures différentes.

Pour certains spécialistes du domaine, la disponibilité représente *« la probabilité qu'un système ou un service soit accessible et opérationnel à un moment donné »*, c'est-à-dire le temps pendant lequel un service ou un système fonctionne effectivement par rapport au temps total durant lequel il est censé fonctionner (Toeroe et Tam 2012) (Varljen 2017). De manière similaire, la disponibilité a été définie par la norme internationale consacrée à la sécurité comme étant « la *propriété d'être accessible et utilisable sur demande par une entité autorisée* ». Cette définition a été utilisée par la norme internationale de base sur le Cloud Computing (ISO/IEC 17788 2014), ayant précisé que «l'entité autorisée» est généralement un client des services Cloud. En conséquence, la disponibilité désigne la probabilité qu'un service Cloud soit opérationnel et accessible par les clients lorsqu'il doit être utilisé (Critchley 2014).

La norme internationale (ISO/IEC 20000-1 2011) définit la disponibilité comme étant « *l'aptitude d'un service ou d'un composant de service de remplir sa fonction requise à un moment convenu ou sur une période de temps convenue* ». En accord avec cette définition, les chercheurs dans (Lu et al. 2013) ont défini la disponibilité comme étant « *la mesure du temps de fonctionnement du système ou du service sur une longue durée*». De même, la disponibilité est définie dans (Neamtiu et Dumitras 2011) comme étant *«la fraction du temps pendant lequel un service est prêt à fournir ses fonctions de façon correcte et ne subit pas de temps d'arrêt planifié ou non planifié*». C'est la propriété d'un service de fournir une disponibilité approximativement proche d'une valeur de disponibilité en régime permanent et qui représente le pourcentage de temps pendant lequel le service est opérationnel dans des circonstances normales tout au long de sa vie utile. La disponibilité reflète la capacité du service à remplir ses fonctions de manière continue au profit de ses utilisateurs, c'est à dire le service doit être conçu pour que les utilisateurs ne subissent pas d'interruptions, y compris les interruptions imprévues généralement dues à des pannes et les interruptions planifiées pour des fins de maintenance et de mise à niveau (Critchley 2014) (Varljen 2017).

D'autres chercheurs ont établi dans (Kaur et al. 2012) la définition de la disponibilité sur la base de paramètres de performance, notamment le temps de réponse. Dans ce contexte, la disponibilité est définie comme étant *«la capacité des services à fournir des réponses rapides aux demandes des clients Cloud »*.Cela permet de garantir un accès rapide et fiable aux services Cloud et à leur utilisation qui doit satisfaire les demandes des clients.

Certains documents de recherche fondent également leur définition de la disponibilité sur les Accords de Niveau de Service (*SLA*). Un SLA est un accord destiné à définir les engagements officiels entre un fournisseur de services et un consommateur de services, qui spécifie généralement la qualité du service fourni, y compris sa mesure et son niveau minimum, ainsi que les critères de violation de cette qualité spécifiée (Toosi et al. 2014). Par conséquent, la disponibilité est considérée comme étant *« la probabilité de fournir un service selon des exigences définies»* (Undheim et al. 2011). Dans cette optique, la disponibilité quantifie la capacité d'un système à supporter un niveau de service engagé (Vargas 2000). Étant donné que le moment de l'interruption/temps d'arrêt constitue l'élément clé de la disponibilité du service, les fournisseurs de services Cloud précisent dans le *SLA* sa définition, ses critères et la manière dont il est mesuré.

4. Taux de Disponibilité

Les fournisseurs de services Cloud tentent d'atteindre des taux élevés de disponibilité pour leurs services afin de garantir les performances prédéfinies et de répondre aux exigences strictes des environnements Cloud Computing. Dans de tels environnements, le taux de disponibilité du service est généralement quantifié par le nombre de neuf ou la classe de neuf dans les chiffres. Ce pourcentage reflète le temps d'arrêt total acceptable pour une période donnée. Pour des applications non critiques telles que les systèmes de planification des vacances du personnel interne d'une organisation, une disponibilité de 95 % peut être parfaitement acceptable, tandis que 99,5 % peut être totalement inacceptable pour d'autres applications telles que les systèmes de réservation de billets d'avion. Le tableau 4.1 ci-dessous indique la durée maximale d'arrêt de service pour les taux de disponibilité courants, conformément à (GR2841). Chaque temps d'arrêt dans le tableau définit un niveau de disponibilité du service exprimé par l'une des valeurs de disponibilité, où les cinq derniers niveaux sont souvent appelés respectivement Bronze, Argent, Or, Super-Or et Platine.

Tableau 4.1: Taux de Disponibilité et Temps d'Arrêt du Service par (GR2841).

Disponibilité du Service(%)	Classe de Neufs	Temps d'Arrêt Annuel	Temps d'Arrêt Mensuel
90%	Un neuf	36,5 jours	73 heures
99%	Deux neufs	3,65 jours	7,3 heures
99,9%	Trois neufs	8,76 heures	43,8 minutes
99,99%	Quatre neufs	52,6 minutes	4,38 minutes
99,999%	Cinq neufs	5,26 minutes	26,4 secondes
99,9999%	Six neufs	31,5 secondes	2,64 secondes
99,99999%	Sept neufs	3.15 secondes	262,97 millisecondes

La haute disponibilité dans le Cloud est une exigence rigoureuse qui permet de limiter les temps d'arrêt. Elle est définie par la bibliothèque pour l'infrastructure des technologies de l'information (en anglais Information Technologies Infrastructure Library, abrégé ITIL) comme étant «la capacité et la propriété d'un service informatique à minimiser ou masquer les effets de la défaillance d'un composant sur les activités des utilisateurs». En effet, un système ou un service en haute disponibilité doit rester continuellement disponible pour ses utilisateurs. Cela se traduit par la nécessité d'une disponibilité d'au moins 99,999 % (cinq neuf) du temps, ce qui permet un temps d'arrêt maximal de cinq minutes et 26 secondes par an Tam 2012). Pour répondre à cette exigence, il convient de garantir que les ressources requises pour fournir un service soient capables d'atténuer toute défaillance (BENAC et al. 2015). Bien que la plupart des fournisseurs de Cloud Computing prétendent abstraitement à une haute résilience et indiquent une disponibilité élevée dans leurs accords de niveau de service, leur temps d'arrêt moyen est en fait plus long que ce qui est considéré comme une haute disponibilité.

5. Mesure de la Disponibilité du Service

La disponibilité du service est un indice qui peut être quantifié comme étant le rapport entre le temps de bon fonctionnement du service (en anglais, Uptime) et le temps total d'usage requis, soit la somme

du temps de bon fonctionnement et du temps d'arrêt du service (en anglais, Downtime) (Bauer et Adams 2012). Elle peut être mesurée à l'aide de l'équation 1.

$$Disponibilité = \frac{Temps\ de\ fonctionnement\ du\ service}{(Temps\ de\ fonctionnement\ du\ service + Temps\ d'arrêt\ du\ service)} \quad (1)$$

Où le « $Tempsdefonctionnement\ duservice$ » est la durée pendant laquelle le service est opérationnel et fournit correctement ses fonctionnalités, et où le « $Tempsd'arrêtduservice$ » (ou temps d'interruption) est la période pendant laquelle le service n'est pas fourni.

La majorité des fournisseurs de services Cloud surveillent attentivement les temps d'arrêt des services qu'ils proposent. Dans ce sens, le calcul de la disponibilité peut se baser essentiellement sur le temps d'arrêt du service (en anglais, Downtime) ainsi que sur le temps total pendant lequel le système était censé être en ligne/opérationnel (en anglais, Total In Service Time) (Bauer et Adams 2012).

$$Disponibilité = \frac{Temps\ Total\ de\ Fonctionnement - Temps\ d'arrêt\ du\ service}{Temps\ Total\ de\ Fonctionnement} \quad (2)$$

Où, le « $Temps\ d'arrêt\ du\ service$ » est la durée de temps pendant laquelle le service ne peut être fourni, et où me « $Temps\ Total\ de\ Fonctionnement$ »désigne la durée de temps au sein de la période de mesure, pendant laquelle le système était considéré comme étant fonctionnel, de sorte que les temps d'arrêt planifiés soient ignorés implicitement.

En revanche, l'ITIL propose une formule simple (équation 3) pour quantifier la disponibilité, en excluant explicitement les temps d'arrêt planifiés dans les calculs de disponibilité. Elle utilise le terme « Temps de Fonctionnement de service convenu » (en anglais, AgreedServiceTime) afin de préciser que les systèmes ont souvent des périodes de maintenance planifiée pendant lesquelles le service peut être interrompu pour des raisons de maintenance (ITILv3SD):

$$Disponibilité\ (\%) = \frac{Temps\ de\ Fonctionnement\ de\ service\ convenu - Temps\ d'arrêt\ du\ service}{Temps\ de\ Fonctionnement\ de\ service\ convenu} \times 100\% \quad (3)$$

Où le «$Temps\ de\ Fonctionnement\ de\ service\ convenu$ » correspond au temps inclus dans la période de mesure, pendant lequel le système a été estimé opérationnel, en ignorant les temps d'arrêt planifiés.

La disponibilité du service représente le pourcentage de temps pendant lequel un service est prêt à exécuter ses fonctions. Elle peut être mesurée en se basant sur :

1) Le temps moyen entre les défaillances (en anglais, MeanTimeBetweenFailure, abrégé MTBF) qui reflète le temps pendant lequel le service est censé être opérationnel ou en ligne avant de tomber en panne (à nouveau).
2) Le temps maximum pour réparer (en anglais, Maximum Time To Repair, abrégé MTTR) qui reflète le temps nécessaire pour résoudre un problème particulier et remettre le service en ligne.

La formule 4 est utilisée pour calculer le facteur de disponibilité du service (Bauer et Adams 2012) (Varljen 2017) (Toeroe et Tam 2012).

$$Disponibilité = \frac{MTBF}{MTBF + MTTR} \quad (4)$$

Il convient de noter que des valeurs faibles de MTTR et des valeurs élevées de MTBF sont souhaitables. En effet, à mesure que MTTR s'approche de la valeur zéro, la disponibilité augmente pour atteindre 100 %. En revanche, à mesure que MTBF devient plus grand, le MTTR aura moins

d'impact sur la disponibilité. Cette équation est équivalente à l'équation 1 définissant la disponibilité de service. A noter que MTBF correspond à une estimation du temps fonctionnel et MTTR à une estimation du temps d'arrêt. L'équation (4) permet de calculer les valeurs standards présentées dans le tableau 1. Ces valeurs constituent une base de référence utilisée dans l'évaluation des services Cloud et la planification de leurs déploiements (Bauer et Adams 2012).

6. Coût de l'Indisponibilité

Le Cloud Computing joue actuellement un rôle important dans la fourniture de services informatiques à des entreprises de tailles diverses. Les coûts globaux des services Cloud sont influencés par les niveaux de disponibilité et les investissements requis pour fournir des services qui répondent aux exigences préétablies. Dans la mesure où la disponibilité est un facteur crucial dans une telle infrastructure, il convient de souligner que l'indisponibilité des services a des coûts considérables, voire désastreux, dans le cas de systèmes d'entreprise critiques (Alshayeji et al. 2018). L'impact total de l'indisponibilité d'un service dans le Cloud peut être obtenu en agrégeant toutes les pertes subies pendant l'indisponibilité. La compréhension des coûts d'indisponibilité constitue un facteur de performance essentiel dans les phases de conception et d'exploitation de la gestion des niveaux de service. Parmi les coûts engendrés par une indisponibilité, on peut citer:

- ✓ ***Coûts de la Perte Commerciale***

L'indisponibilité des services Cloud entraine la perte d'opportunités commerciales. En effet, elle entrave les ventes et empêche de gagner de la clientèle par exemple. Les fournisseurs de services perdent ainsi des revenus financiers, ce qui est engendré par une perte de productivité. De ce fait, l'indisponibilité des services Cloud a un impact direct sur les transactions informatiques et commerciales (Adekunle et Osimosu 2013).

- ✓ ***Coûts des Amendes ou des Pénalités***

Les fournisseurs de services Cloud sont, dans certaines situations, contraints de payer des pénalités ou des amendes qui leurs sont imposées lorsqu'une défaillance de leurs services est constatées, notamment lorsqu'il s'agit de données confidentielles de clients (Adekunle et Osimosu 2013).

- ✓ ***Coût des Heures de Travail Perdues***

Une perte de productivité du personnel est susceptible d'être entrainée par les heures de travail perdues. A long terme, ceci peut s'avérer coûteux autant pour les clients que pour les fournisseurs de services Cloud. Dans la majorité des cas, qu'ils atteignent ou non leurs objectifs de travail, les employés permanents sont quand même payés (Adekunle et Osimosu 2013). Les clients peuvent afficher une insatisfaction à cause l'indisponibilité des services pendant ces heures tout comme en raison de l'arrêt du travail.

- ✓ ***Coût de Récupération***

Les fournisseurs de Cloud doivent engager des coûts de récupération spécifiques pour se remettre d'une panne. Il s'agit par exemple des couts des services de reprise après sinistre ou encore du payement d'heures supplémentaires aux employés (Adekunle et Osimosu 2013).

- ✓ ***Perte de Réputation***

La réputation d'un fournisseur de services Cloud peut subir des dommages durables à cause de l'indisponibilité. Or, elle est importante car les clients, pour choisir un fournisseur, se réfèrent normalement à un modèle de confiance basé sur les preuves de disponibilité, donc la réputation

(Adekunle et Osimosu 2013). De ce fait, les fournisseurs de services Cloud risquent de perdre la confiance de leurs clients à cause des interruptions de service.

7. Mécanismes de Disponibilité dans le Cloud Computing

Dans les systèmes à forte intensité tels que le « Cloud Computing », la disponibilité constitue une préoccupation récurrente et croissante. Dans un tel système, elle permet d'assurer et de garantir un fonctionnement approprié et permanent des services ou applications proposés. Dans cette optique, les fournisseurs de services Cloud ont concentré leurs efforts sur la mise en œuvre de toutes les actions et les dispositions techniques visant à améliorer leurs infrastructures et leurs stratégies de gestion. Par conséquent, différents mécanismes ont été envisagés. Parmi les plus populaires, on trouve l'équilibrage des charges, la redondance, la réplication des données et la tolérance aux pannes. Ces techniques ont pour objectif d'assurer la disponibilité et la fiabilité des services tout en prévenant la dégradation des performances des systèmes Cloud.

7.1. Equilibrage de Charge

L'équilibrage de charge est un mécanisme très important car il fournit un débit maximal avec un temps de réponse minimal, ce qui favorise la disponibilité des ressources et des réseaux (Durkee 2010). L'équilibrage des charges joue un rôle crucial dans le Cloud Computing en raison de la capacité de mise à l'échelle automatique que le Cloud offre à ses clients ainsi que de l'arrivée irrégulière des tâches. En effet, une protection contre la surcharge de toute ressource de calcule est offerte par la mise à l'échelle automatique qui permet l'optimisation des ressources. Toutefois, cette mise à l'échelle automatique doit être compléter par un mécanisme d'équilibrage de charge, répartissant la charge entre les ressources informatiques (Nabi et al. 2016). Le planificateur assure l'équilibrage de la charge. Il planifie les tâches et les affecte aux nœuds du Cloud. Les données, en répartissant le trafic entre les serveurs, peuvent être reçues et envoyées sans que des retards importants ni de longues réponses du système soit enregistrés. Les techniques d'équilibrage de charge au sein du Cloud Computing sont appliquées à travers les différents centres de données. Elles servant à minimiser l'utilisation du matériel informatique, à maximiser la disponibilité du réseau et des ressources et à réduire de manière drastique les temps d'arrêt et les pannes qui pourraient affecter les systèmes Cloud instantanément. Ainsi, les équilibreurs de charge contribuent à promouvoir les performances du Cloud ainsi que la disponibilité de ses ressources. Les algorithmes d'équilibrage de charge qui ont été élaborés sont nombreux. Ils peuvent être classés en deux catégories principales (Chaczko et al. 2011):

- ***Les Algorithmes Statiques:***

Les algorithmes de planifications statiques sont fondés sur la base du concept de la distribution égale, entre les serveurs, du trafic. Les algorithmes adoptés dans ce concept sont annoncés comme algorithme Round-Robin. C'est pour les environnements stables et homogènes qu'ils sont principalement utilisés. Ils sont capables de produire de très bons résultats. Malgré le fait que la règle sur laquelle sont basés ces algorithmes est simple, une augmentation des charges sur les serveurs est entrainée par eux, ce qui cause un déséquilibre du trafic. De ce fait, le Round-Robin pondéré a été défini en attribuant aux serveurs des poids. Les serveurs ayant le poids le plus élevé reçoivent le plus de connexions. Cette approche est acceptable dans le cas de poids inégaux. Cependant, elle constitue une surcharge pour les poids égaux qui sont imposés aux serveurs (Chaczko et al. 2011) (Vani et Priya 2015). Ajouté à cela le fait que ce genre d'algorithmes ne sont pas flexibles ni capables de s'adapter aux changements dynamiques du système et de les accepter.

- ***Les Algorithmes Dynamiques:***

Des poids appropriés sont attribués aux serveurs par les algorithmes de planifications dynamiques. Puis, ces algorithmes cherchent dans tout le réseau le serveur le plus léger qui pourrait équilibrer le trafic. Ils sont flexibles et efficaces dans les environnements informatiques dynamiques et hétérogènes. Ces techniques sont jugées compliquées en matière de mise en œuvre, bien qu'elles soient acceptables. Néanmoins, elles souffrent d'une surcharge importante du système et de sérieux problèmes de trafic. En effet, une communication en temps réel avec les réseaux est nécessaire pour sélectionner un serveur approprié, ce qui entraine une diminution des performances du système, engendrée par le trafic supplémentaire qu'il subit (Chaczko et al. 2011) (Vani et Priya 2015).

7.2. Redondance

La redondance fait partie des mécanismes les plus populaires pour augmenter la disponibilité dans le Cloud Computing. Il s'agit d'utiliser des éléments redondants, non nécessaires si le fonctionnement du système était correct (Nabi et al. 2016) (Toeroe et Tam, 2012). Le fait que toute défaillance individuelle ait une solution de repli dans l'architecture est garanti par la redondance dans le Cloud. C'est ainsi qu'est assurée l'absence de temps d'arrêt, même dans les pires scénarios. *Cela signifie qu'en cas de perturbation des opérations informatiques, les activités et les services Cloud peuvent continuer normalement*. La redondance consiste à dupliquer des composants de manière à ce qu'un composant puisse prendre le relais en cas de défaillance de son partenaire et que le système informatique ne soit pas affecté. Dans une telle situation, les composant primaires et les composants supplémentaires peuvent collaborer suivant deux modes (ou stratégies) de fonctionnement principaux, à savoir le mode active-active et le mode active-standby/passive. Les composantes première et secondaire partagent, dans le mode de redondance active-active, la charge de travail du système en fonctionnant en parallèle. Si une défaillance survient dans l'un d'entre eux, un autre composant répondra aux demandes de service formulées par les utilisateurs du système. Par ailleurs, les composants primaires répondent aux demandes des utilisateurs du système dans le mode de redondance active-standby, et dans ce cas, les composants secondaires demeurent en attente. Lorsqu'une panne survient dans le composant primaire, toute réplique parmi les composants secondaires est capable de reprendre les tâches du nœud défaillant et répondre ainsi aux demandes des utilisateurs du système (Melo et al. 2018). Le nom « failover », qui signifie basculement, désigne le transfert du contrôle d'un composant défaillant à son partenaire. Ledit transfert est aussi appelé « switchover » dans certains cas (Critchley 2014).

Il est possible d'organiser les éléments redondants de manières nombreuses se selon différentes règles composant ensemble des modèles redondance. En effet, le modèle de redondance se réfère au grand nombre de façons par lesquelles les systèmes peuvent combiner des répliques de types actif et standbay pour prévenir toute défaillance susceptible de causer des interruptions de service. Quantre modèles de redondance très répandus sont définis par le cadre de gestion de la disponibilité (en anglais Availability Management Framework, abrégé AMF). Il s'agit de : 2N, N+M, Nway et Nway active. Pour ce qui est de la portée, la redondance peut concerner le serveur physique complet qui héberge l'application, la machine virtuelle qui héberge l'application ou l'application elle-même. Les chercheurs et les spécialistes du domaine proposent d'utiliser toutes ces approches pour assurer des niveaux élevés de la disponibilité des applications Cloud (Endo et al 2016) (Nabi et al. 2016).

De ce fait, les data centers dupliques les infrastructures. Ils englobent divers stratégies et modèles de redondance rendant possible, en cas de défaillance du site principale, le basculement automatique des applications et des données sur un site de repli (BENAC et al. 2015). En effet, le temps nécessaire à la redondance dans le Cloud pour permettre de rétablir les services est bien moins conséquence que celui

nécessaire pour réparer l'élément défaillant. D'un point de vue opérationnel, le basculement du service vers un élément redondant devrait être beaucoup plus rapide (en quelques secondes par exemple) que la réparation de l'élément défaillant (pouvant prendre des heures). Par conséquent, la redondance demeure nécessaire afin de maintenir la disponibilité dans le Cloud Computing. Toutefois, lorsqu'elle est utilisée de manière excessive, elle est susceptible d'affecter la vitesse et les performances globales et d'entrainer une surcharge du réseau. Par ailleurs, la stratégie active présente la difficulté de maintenir la cohérence.

7.3. Réplication des Données

L'une des principales exigences dans le Cloud Computing est la disponibilité des données. La réplication des données est, depuis de nombreuses années, l'un des mécanismes les plus étudiés. Il est utilisé pour accroitre la disponibilité dans les systèmes distribués à grande échelle, dont le Cloud Computing. Il s'agit dans mécanisme d'intérêt dans le Cloud Computing, consistant en la création de plusieurs copies identiques de certaines données (bases de données, fichiers, etc.) et en leur stockage sur plusieurs répliques, celles-ci étant des sites géographiquement distribués. La réplication est également connue comme étant la duplication des données. Elle permet la création de copies d'une donnée et leur stockage sur plusieurs nœuds différents qui sont situés dans le réseau. La gestion de la réplication des données considère, au sein du Cloud Computing, des contraintes de disponibilité includes dans quatre processus essentiels. Il s'agit du choix du moment de la réplication, du choix du fichier à répliquer, du choix du nombre et de l'emplacement de la réplique. Il faut noter que la méthode de création de la réplique consiste à décider du fichier de données devant être répliqué dans le système Cloud et à trouver ensuite le moment opportun pour créer de nouvelles répliques de ces données. Par ailleurs, cette méthode choisit le nombre de nouvelles répliques appropriées devant être crées dans le système Cloud, en plus de leur emplacement parmi les nœuds du réseau. Le but étant de satisfaire les exigences des usagers en matière de consommation de bande passante, de temps d'attente, d'équilibrage de la charge, de la performance d'accès aux données et de la disponibilité de ces dernières (Milani et Navimipour, 2016) (Vani et Priya 2015). Il est possible que la réplication des données dans le Cloud Computing soit effectuée selon un protocole de réplication passive, active ou semi-active. S'agissant du protocole de réplication passive (en anglais, passive replication ou backup replication), une seule copie, nommée copie primaire (primary copy), reçoit la requête d'un client et l'exécute. Pour ce qui est des autres copies, appelées copies de sauvegarde (backup) ou copies secondaires (secondary copies), elles ne sont là que pour prendre le relais dans le cas où une copie primaire subit une défaillance. Concernant le protocole de réplication active (en anglais, state machine approche ou active replication), toutes les copies jouent un même rôle. Elles reçoivent toutes, de la part des clients, la même séquence de requêtes totalement ordonnée (Vijay et Reddy 2013), l'exécutent de façon déterministe et renvoient la même séquence des réponses totalement ordonnée. Il existe un protocole hybride, connu comme protocole de réplication semi-active, qui se situe entre les deux protocoles précédents. Toutes les copies y exécutent en même temps les requêtes des clients, mais seulement l'une d'entre elles (leader) émet la réponse. Les autres copies (suiveurs), étroitement synchronisées avec le leader, mettent à jour leur état interne (Meslem et Debbas 2017).

Il est possible de classer les techniques de réplication des données en deux groupes principaux : la stratégie de réplication statique et celle dynamique (Milani et Navimipour, 2016).

- ✓ *Techniques de réplication Statiques*

Les stratégies de réplication statiques sont telles que le nombre de répliques et l'emplacement de ces dernières sont définis et fixés à l'avance, des politiques déterministes étant suivies à cet effet (Milani

et Navimipour, 2016). La mise en œuvre de ces techniques et simple et celles-ci offrent une grande efficacité et une réponde rapide, permettant une haute disponibilité.

- ✓ *Techniques de réplication dynamiques*

Les répliques sont automatiquement créées et supprimées par les stratégies dynamiques de réplication des données, et ce, en fonction de la capacité de stockage, des changements du modèle d'accès des utilisateurs et de la bande passante. Des choix intelligents sont faits sur la localisation des données, en fonction des besoins de l'environnement actuel et des informations que l'on possède sur celui-ci. Ces stratégies sont, de ce fait, principalement plus adaptées à un environnement orienté service.

La réplication des données dans les systèmes Cloud a pour objectif de garantir une meilleure disponibilité des données, en rendant meilleures les performances d'accès aux données. En effet, elle permet de réduire le temps de transfert des données et la latence d'accès en offrant une autonomie accrue, un accès efficace et rapide ainsi qu'un parallélisme lors de la consultation de la même donnée. La réplication des données dans les systèmes Cloud permet aussi d'améliorer la tolérance aux pannes, l'évolutivité et la fiabilité, ainsi que d'offrir un équilibrage de la charge acceptable en répartissant cette dernière entre les nœuds du réseau. Cependant, les performances nécessaires peuvent être garanties par le mécanisme de réplication à un goût élevé ayant un effet négatif sur le système. Le coût est, en effet, directement proportionnel au nombre de répliques généré dans le réseau. En fait, un coût de maintenance et de gestion élevé peut être engendré par l'augmentation supplémentaire du nombre de répliques. En outre, plusieurs ressources, telles que l'espace de stockage, sont nécessaires à cette augmentation. Cette dernière peut entrainer une forte consommation d'énergie et de bande passante ainsi qu'une surcharge de réseau en stockant plusieurs répliques. De plus, la réplication des données, particulièrement celle dynamique, présente la difficulté de maintenir la cohérence des fichiers de données et de collecter les informations d'exécution de tous les nœuds de données dans une infrastructure de Cloud complexe. Ceci est l'un de ses plus importants défis (Milani et Navimipour, 2016) (Meslem et Debbas 2017).

7.4. Tolérance aux Pannes

La tolérance aux pannes « en anglais Fault Tolerance, abrégé FT » est une méthode de conception permettant à un système de demeurer fonctionnel, de manière réduite éventuellement, au lieu de tomber en panne complètement. Elle lui permet de rester relativement opérationnel lorsqu'un dysfonctionnement survient dans l'un de ses composants. La tolérance aux pannes fait référence à l'aptitude d'un réseau ou d'un système poursuivre le traitement informatique de manière continue et de maintenir pour cela ses fonctionnalités en cas de panne de certains de ses nœuds. Pour ce faire, la panne doit être détectée et le système est tenu d'en remettre. La dissimulation de l'apparition de l'occurrence des pannes et la minimisation des impacts de ces dernières sur la tâche globale du réseau/système sont les deux objectifs que la tolérance aux pannes permet de concrétiser (JALOTE 1994).

Dans les systèmes de Cloud Computing, la tolérance aux pannes se réfère à la capacité du Cloud à résister aux changements brusques qui se produisent en raison de différents types de défaillances : matérielles, logicielles, de congestion du réseau, etc. Selon la manière et le moment du traitement des pannes, deux politiques de tolérance aux pannes sont disponibles pour les environnements Cloud Computing, à savoir la politique de tolérance aux pannes proactive (en anglais, Proactive Fault Tolerance, abrégé PFT) et la politique de tolérance aux pannes réactive (en anglais, Reactive Fault Tolerance, abrégé RFT) (Ataallah et al. 2015). C'est pour fournir des techniques de tolérance aux pannes dans le Cloud que ces politiques sont principalement utilisées. La réparation et le

remplacement rapides des composants défectueux se dont grâces à ces techniques, ce qui permet de conserver un état sain du système et donc de poursuivre la prestation des services (Mesbahi et al. 2018).

- ***Tolérance aux Pannes Réactive***

Le traitement des mesures appliquées est la base sur laquelle repose la tolérance aux pannes réactives. Cela permet de réduire les pannes ayant déjà au lieu dans un système Cloud (Mesbahi et al. 2018). Ladite tolérance essaye de maintenir les services Cloud par une technique de récupération. Plusieurs techniques différentes sont basées sur cette politique, telles que le point de contrôle/redémarrage (checkpoint/restart), la réplication, la migration des tâches, réessayer (retry-it), les resoumissions des tâches, la gestion des exceptions définies par l'utilisateur, le sauvetage du flux de travail, SGuard, etc (Alshayeji et al. 2018) (Bala et Chana, 2012).

- ***Tolérance aux Pannes Proactive :***

La tolérance aux pannes proactive signifie le fait d'éviter toute défaillance ou panne anticipées avant qu'elles se produisent réellement en prenant des mesures préventives appropriées. Elle permet de remplacer les composants suspects par d'autres fonctionnels, et ce, de manière proactive et après avoir prédit la défaillance. Ceci permet de garantir une exécution réussie et complète des services Cloud (Mesbahi et al. 2018).

Parmi les techniques basées sur cette politique figurent la migration préemptive, le rajeunissement des logiciels, l'auto-guérison, etc (Alshayeji et al. 2018) (Bala et Chana, 2012).

De nombreux avantages découlent de l'utilisation de la tolérance aux pannes dans le Cloud Computing, notamment la réduction des coûts, la récupération des pannes et l'amélioration de la fiabilité et des mesures de performance, surtout la disponibilité (Hosseini et Arani, 2015). Toutefois, lors de l'élaboration de notre revue de la littérature, nous avons pu constater différents défis liés à l'intégration de la tolérance aux pannes dans le Cloud Computing. En effet, une réflexion et une analyse approfondies sont nécessaires à la mise en œuvre de la tolérance aux pannes, en raison de leur interdépendance et de leur complexité. Le plus grand obstacle pouvant entraver la localisation des pannes est l'hétérogénéité du Cloud. Ajouté à cela le fait que l'interprétation de l'état changeant du système est difficile car les environnements Cloud sont dynamiquement évolutifs et inattendus. Lorsque plusieurs instances sont exécutées sur différentes machines virtuelles, les techniques de tolérance aux pannes ne fonctionnement pas (Kaur et Singh, 2017). En outre, une surcharge du système, des interruptions temporaires, une augmentation du coût et des frais généraux élevés peuvent être entraînés par certaines de ces techniques (Mesbahi et al. 2018).

8. Solutions pour la Disponibilité dans le Cloud Computing

La disponibilité des services est considérée comme l'une des caractéristiques essentielles pour maintenir la qualité de service dans les environnements distribués orientés services. De nombreux efforts de recherche et des études actuelles ont été consacrés au développement de mécanismes visant à garantir et à augmenter la disponibilité des ressources et les performances dans les environnements distribués, en particulier dans le Cloud Computing. Les auteurs de (Nabi et al. 2016) ont présenté une taxonomie pour classer les articles de recherche et les solutions des fournisseurs de Cloud. En adoptant proportionnellement cette taxonomie, cette section aborde certains des travaux qui visent à améliorer et à maintenir la disponibilité dans les environnements de Cloud Computing.

Certains auteurs fondent leurs approches de garantie de disponibilité sur *les accords de niveau de service (SLA)* entre les fournisseurs de services et les utilisateurs, qui font référence à la probabilité de fournir un service conformément aux exigences définies. Dans (Pavlik et al. 2014), les auteurs ont proposé une solution complète de gestion des accords de niveau de service (SLA) dans le Cloud Computing, en partant d'une perspective centrée sur le client, afin de maintenir la disponibilité requise des services du Cloud. L'approche proposée mesure et surveille la disponibilité des services Cloud en utilisant des systèmes qui détectent les demandes des clients pour les services Cloud et un système de surveillance BlackBox qui collecte des ensembles de données (succès ou échec et latence). En outre, les auteurs ont utilisé l'analyse de la qualité de l'ajustement (distribution de probabilité) pour évaluer la représentation de la disponibilité. Dans (Grover 2014), les auteurs ont proposé une approche pour la génération automatique de services de classement pour les systèmes «Cloud Computing Peer-to-Peer», basée sur un classement de multiples facteurs de confiance qui sont: la surveillance des SLA, l'algorithme d'équité des ressources, la méthode de paiement et le schéma tarifaire.L'approche mesure automatiquement la valeur des paramètres de QoS, la violation des SLA, la surveillance des SLA et la pénalité des SLA. Ces informations SLA sont ensuite analysées afin de générer une liste de classement pour l'utilisateur en tant que fournisseur de Cloud. Cette approche peut être utile aux utilisateurs pour obtenir un service de confiance à moindre coût.

L'amélioration de la disponibilité des services Cloud dans d'autres documents de recherche est assurée par le développement de nouvelles politiques ou l'adoption de méthodes standard de *réplication* et de *planification*. N.Saadat et A.M.Rahmani (Saadat et Rahmani 2012) ont proposé un algorithme de réplication dynamique des données basé sur la pré-extraction (en anglais, pre-extraction-based dynamic data replication algorithm, abrégé PDDRA). Sur la base de l'historique d'accès aux fichiers des sites de la grille, PDDRA prédit les besoins futurs et pré extrait les fichiers vers le site de la grille du demandeur, de sorte que la prochaine fois que ce site aura besoin d'un fichier, il sera disponible localement. Cela permet de réduire considérablement la latence d'accès, le temps de réponse et la consommation de bande passante. Cependant, l'emplacement optimal des répliques n'a pas été étudié dans cet article. Dans (Sashi et Thanamani, 2010), les auteurs ont présenté un algorithme qui prend en compte le nombre de demandes de fichiers et le temps de réponse pour placer la réplique dans un emplacement optimal du cluster. En conséquence, le temps d'exécution moyen des tâches est réduit au minimum. Les auteurs de (Rama et Reddy, 2013) ont proposé un système de réplication de données dans le Cloud Computing, basé sur l'utilisation de l'algorithme d'extraction de motifs fréquents de l'exploration de données pour identifier la popularité des données et générer un seuil adaptatif pour la réplication, qui s'est avéré être un meilleur mécanisme pour assurer un processus de réplication efficace dans le système Cloud. La disponibilité du système et la probabilité de défaillance des données sont mesurées pour identifier l'emplacement où les données répliquées seront stockées. Dans (Hussein et Mousa, 2014) les auteurs ont proposé, une stratégie de réplication des données légère (en anglais, Light-Weight Data Replication Strategy, abrégé LWDRS). La stratégie sélectionne de manière adaptative les fichiers de données à répliquer à l'aide d'une technique de séries chronologiques légères appelée lissage exponentiel linéaire de Holt (en anglais, Holt Linear Exponential Smoothing, abrégé HLES), qui analyse le schéma récent des demandes de fichiers de données et fournit des prévisions pour les futures demandes de données. Dès qu'un facteur de réplication basé sur la popularité des fichiers dépasse un seuil spécifique, le signal de réplication est lancé. La stratégie proposée utilise une recherche heuristique pour déterminer dynamiquement le nombre de répliques ainsi que les nœuds de données actuels pour la réplication. Dans (Rahmati et Rahmani 2017), les auteurs ont proposé une technique de planification basée sur la réplication des données (en anglais, Data replication based scheduling, abrégé DRBS) qui intègre deux techniques: la réplication des données et la

planification des tâches afin d'atteindre l'objectif de réduire le temps d'accès aux données dans le Cloud, d'augmenter l'équilibrage des charges et d'améliorer les performances des applications gourmandes en données. Dans (Netes 2018), l'auteur a clarifié la définition de la disponibilité quantitative ainsi que qualitative des services de «bout en bout Cloud Computing », en considérant tous les composants de l'infrastructure du service. Les résultats montrent que la haute disponibilité nécessite la redondance des composants tels que les connexions réseau et les centres de données. D'autres stratégies et algorithmes de réplication ont été proposés, notamment: Stratégie de réplication dynamique des données (en anglais Dynamic Data Replication Strategy, abrégé D2RS) (Sun et al. 2012), Stratégie de propagation rapide modifiée (en anglais, Modified Fast Spread, abrégé MFS) basée sur la technique standard de propagation rapide (Bsoul et al 2012), Stratégie de réplication hiérarchique de la bande passante (en anglais Bandwidth Hierarchy Replication, abrégé BHR) (Park et al. 2003), Stratégie BHR modifiée (Sashi et Thanamani 2011), Modèle de réplication hiérarchique des données (en anglais Hierarchical Data Replication Model, abrégé HDRM) (Lu et all 2010), Schéma de gestion de la réplication dynamique des données (en anglais Cost-Effective Dynamic Data Replication Management Scheme, abrégé CDRM) qui est rentable pour un système de stockage en nuage prenant en charge les systèmes de fichiers distribués Hadoop (en anglais Hadoop Distributed File Systems, abrégé HDFS) (Wei et all. 2010), Algorithme de réplication des données qui est basé sur le principe du problème du flux maximal au coût minimal (Vijaya et Ramachandhra 2014). Ces études montrent que la réplication est une approche essentielle qui peut être utilisée pour améliorer l'accessibilité des données et le temps de réponse, maximiser le taux d'exécution réussie des tâches et minimiser la consommation de la bande passante afin de garantir la disponibilité et d'améliorer les performances des applications de données volumineuses dans le Cloud.

D'autres chercheurs ont envisagé des techniques de tolérance aux pannes pour protéger les systèmes de Cloud Computing contre tout type de défaillance matérielle ou logicielle et maintenir la disponibilité du service. Dans (Noraziah et al. 2013), une technique de réplication appelée BVACQ (en anglais, Binary Vote Assignment on Cloud Quorum) a été proposée pour préserver la disponibilité et la cohérence des données en cas de défaillance du Cloud. Cette technique combine les mécanismes de réplication et de tolérance aux pannes en considérant la fragmentation de la base de données distribuée. De plus, elle prend en compte la gestion des transactions, ce qui permet de préserver la cohérence et d'augmenter la disponibilité des données et la fiabilité du système. Dans (Thein et Park, 2009), les auteurs proposent un mécanisme de tolérance aux pannes pour assurer la disponibilité des applications, en abordant le rajeunissement des logiciels, basé sur les machines virtuelles(en anglais Virtual Machine based Software Rejuvenation, abrégé VMSR). Un agent de rajeunissement logiciel est installé dans chaque machine virtuelle et surveille le vieillissement des logiciels et les défaillances des applications. Dans cette solution, la détection est faite au niveau de l'application, alors que la réplication est faite au niveau de la machine virtuelle, ce qui soulève le problème de la propagation de la défaillance de l'application aux machines virtuelles en attente. Singh et al (Singh et al. 2012) ont proposé des stratégies de basculement intelligentes pour les centres de données dans le Cloud, à l'aide d'algorithmes de point de contrôle intégrés qui comprennent la prise en charge d'algorithmes d'équilibrage de charge et de point de contrôle multi-niveaux. La stratégie de basculement envisagée s'exécutera au niveau de la couche applicative et fournira une haute disponibilité pour la fonctionnalité PaaS (Platform as a Service) du Cloud Computing. L'étude (Jhawar et Piuri 2013) a examiné l'importance de la gestion adaptative des ressources pour la tolérance aux pannes dans les applications de Cloud Computing. Les auteurs ont étendu ce concept avec un contrôleur en ligne afin de fournir un algorithme heuristique pour restaurer les exigences de l'application au moment de l'exécution en cas de défaillance. Des chaînes

de Markov et des algorithmes de réseaux de files d'attente sont utilisés pour estimer les attributs de disponibilité et de performance d'une implémentation différente. L'approche proposée a augmenté la disponibilité et réduit la dégradation des temps de réponse du système par rapport aux schémas statiques traditionnels. D'autres efforts (Liu et all. 2016) (Zhang et all. 2014) ont été consacrés au développement d'intergiciels et d'architectures de systèmes tolérants aux pannes capables de gérer diverses défaillances logicielles afin d'assurer la disponibilité et la fiabilité des applications serveur dans un environnement virtualisé Cloud VDC.En outre, certains travaux (Makhsous et al. 2016) (Benz et Bohnert 2013) (Brenner et al. 2014) ont développé des approches intéressantes pour atteindre une disponibilité de service adaptative et évolutive dans des environnements Open Stack Cloud Computing et pour assurer un fonctionnement continu de tous les services du Cloud en cas de défaillance. Dans ces travaux, les chercheurs visent à surmonter les faiblesses des approches de tolérance aux pannes existantes en proposant des améliorations des mécanismes de basculement, de surveillance et de pondération, qui ont permis de minimiser le temps de réponse moyen et les coûts d'exploitation, d'améliorer le degré de disponibilité des services requis et les performances du système. Le tableau 4.2 résume les principaux travaux décrits ci-dessus en soulignant leurs avantages et leurs limites. En outre, il présente les principaux critères de performance du Cloud Computing qui ont été pris en compte, notamment : (a) la disponibilité, (b) l'évolutivité, (c) la fiabilité, (d) la réduction des coûts, (e) la réduction du temps d'exécution, (f) la consommation de bande passante, (g) l'équilibrage des charges.

Tableau 4. 2: Résumé des principaux travaux sur l'amélioration de la disponibilité dans le Cloud via des mécanismes de réplication, de planification et de tolérance aux pannes.

Référence	Mécanisme	Architecture/ algorithme	Avantages	Inconvénients	Principaux critères de performance considérés						
					a	b	c	d	e	f	g
(Saadat et Rahmani 2012)	Réplication	Pre-fetching based Dynamic Data Replication algorithm (PDDRA)	Amélioration du temps d'exécution des tâches, de l'utilisation du réseau, du taux de réussite et de l'utilisation du stockage. Réduction de la latence d'accès et de la consommation de bande passante.	Ne considère pas la meilleure sélection de répliques.	☑	☑	☑	☑	☑	☑	☑
(Sashi et Thanamani, 2010)	Réplication	Dynamic replica creation and placement algorithm.	Augmentation de la disponibilité des données et utilisation efficace du réseau. Réduction des coûts de réplication et de placement inutiles ainsi que du temps moyen d'exécution des tâches.	Ne considère que le nombre de demandes et le temps de réponse pour placer les répliques. N'inclue pas la sécurité dans la sélection des répliques.	☑	☑	☑	☑	☑	☒	☒
(Rama et Reddy, 2013)	Réplication et Planification	Data Replication Model based Frequent Patent Mining	Disponibilité des données accrue, probabilité de perte de données réduite, meilleure efficacité et temps d'accès aux données.	Coût de réplication élevé et faible équilibrage de charges	☑	☑	☑	☒	☑	☑	☒
(Hussein et Mousa, 2014)	Réplication	Light-Weight Data Replication (LWDRS)	Optimisation de la qualité de service non fonctionnelle et de la fiabilité globale. Haute disponibilité et accès efficace aux centres de données.	Frais généraux élevés et faible équilibrage de la charge.	☑	☑	☑	☑	☒	☒	☒
(Rahmati et Rahmani 2017)	Réplication et Planification	Data Replication Based Scheduling Algorithm (DRBS)	Amélioration du temps de réponse et de l'équilibrage de la charge. Faible consommation de bande passante. Hautes disponibilité, efficacité et fiabilité.	Coût élevé de stockage et de réplication.	☑	☑	☑	☒	☑	☑	☒
(Sun et al. 2012)	Réplication	Dynamic Data Replication Strategy (D2RS)	Amélioration de la disponibilité et du taux de réussite. Minimisation des délais du réseau et de l'utilisation de la bande passante.	Coût de réplication élevé et faible équilibrage des charges.	☑	☑	☑	☒	☑	☑	☒
(Bsoul et al 2012)	Réplication	ModifiedFast Spread (MFS)	Amélioration des performances en termes de temps de réponse et de consommation totale de bande passante.	Faible fiabilité et coût de stockage élevé.	☑	☑	☒	☒	☑	☑	☒

(Park et al 2003)	Réplication	Bandwidth Hierarchy based Replication (BHR)	Réduction du temps d'accès aux données et du temps d'exécution des tâches. Prévention de la congestion du réseau. Amélioration de l'efficacité du réseau et de l'utilisation des ressources de stockage.	Ne fonctionne bien que lorsque la capacité de stockage est limitée.	☑	☑	☑	☒	☑	☑	☑
(Sashi et Thanamani 2011)	Réplication	Modified Bandwidth Hierarchy based Replication (MBHR)	Réduction du temps moyen d'exécution des tâches, de l'utilisation du réseau, de la consommation de bande passante et du trafic réseau. Amélioration de la disponibilité et des performances des données. Minimisation du temps d'accès aux données et de la réplication inutile.	Faible équilibrage de la charge.	☑	☑	☑	☑	☑	☑	☒
(Lu et al. 2010)	Réplication	HierarchicalReplication Model (HRM)	Haute efficacité, transmission accélérée des données, amélioration de la disponibilité et de l'accessibilité des données.	Non spécifié.	☑	☑	☑	☑	☑	☒	☒
(Vijaya et Ramachandhra 2014)	Réplication	Data Replication quality of service (DRQS) algorithm	Réduction du temps de réponse et des coûts de réplication. Disponibilité, évolutivité et capacités accrues.	Coût de stockage et consommation d'énergie élevés dans les centres de données.	☑	☑	☑	☒	☑	☒	☒
(Wei et al. 2010)	Réplication	Cost-effective dynamicreplication management scheme (CDRM).	Schéma rentable qui peut optimiser le nombre de répliques, le parallélisme inter-requête et intra-requête. Amélioration de la disponibilité, de la latence d'accès, des performances et de l'équilibre de la charge pour les applications réelles.	Ne traite pas les nœuds en panne contenant des morceaux de fichiers supprimés.	☒	☒	☒	☒	☒	☒	☒
(Noraziah et all. 2013)	Réplication et Tolérance aux Pannes	Binary Vote Assignment on Cloud Quorum (BVACQ).	Préservation de la cohérence des données et de la fiabilité du système, et augmentation de la disponibilité des données.	Coût de stockage élevé et faible performance en raison de la réplication.	☑	☑	☑	☑	☑	☑	☒
(Thein et Park, 2009)	Tolérance aux Pannes	Virtual Machine based Software Rejuvenation (VMSR)	Prévention, disponibilité et efficacité élevées en matière de tolérance aux pannes. Réduction des coûts de gestion des applications vieillissantes. Optimisation des performances du système.	Non adapté aux profils de vieillissement non déterministes et aux défaillances transitoires.	☑	☑	☑	☑	☑	☒	☒
(Singh et al. 2012)	Tolérance aux Pannes	Failover Strategies based on Integrated Checkpointing Algorithms	Fournir une haute disponibilité aux demandes des clients cloud en cas de défaillance/ récupération des nœuds de service. Réduction des frais généraux liés aux points de contrôle.	N'intègre pas le rajeunissement du moniteur de la machine virtuelle. Retards dans l'exécution dus à l'allocation aléatoire des nœuds.	☑	☑	☑	☑	☒	☒	☑
(Jhawar et Piuri 2013)	Tolérance aux Pannes	Adaptive Resource Management for Balancing Availability And Performance in Cloud Computing	Amélioration de la disponibilité et des performances des applications. Réduction du temps de réponse du système même en cas de défaillance et de récupération.	Ne supporte pas les environnements Cloud à grande échelle.	☑	☑	☑	☑	☑	☒	☒

En examinant de près ces travaux, on se rend compte que l'interprétation de la disponibilité peut différer d'un document à l'autre et d'un fournisseur à l'autre. Bien que la plupart de ces techniques ont eu un effet positif sur l'augmentation et l'amélioration de la disponibilité et des performances du système tout en évitant les pannes fréquentes, elles ne sont pas en mesure de garantir la disponibilité souhaitée en raison de leurs diverses limitations. Outre le volume dynamique des données analysées dans les environnements distribués qui nécessite une gestion optimale, les techniques mentionnées ci-dessus permettent généralement la production d'énormes quantités de données circulant dans le réseau, ce qui peut les limiter pour maintenir idéalement les performances du système. Cette grande quantité de données nécessite un espace de stockage plus important, ce qui rend la maintenance et la gestion permanente trop coûteuses et complexes, augmentant inutilement les coûts de gestion et de stockage. Elle nécessite également une connaissance globale du réseau, ce qui la rend plus difficiles à mettre en œuvre, limitant de ce fait la performance de la recherche des données requises. Outre le retard de traitement et d'exécution dans les réseaux étendus, la difficulté d'accès aux données, l'augmentation du temps de réponse et la limitation de la bande passante du réseau, la réduction de la priorité de la reprise après défaillance et l'augmentation du temps d'inactivité des pairs, elle crée un goulot d'étranglement très sensible à l'accès simultané aux données dans l'environnement distribué.

9. Conclusion

Dans ce chapitre, nous avons présenté les facteurs et les unités vulnérables aux différentes défaillances pouvant entraîner l'interruption des services Cloud. Ensuite, nous avons présenté diverses définitions de la disponibilité des services dans le contexte du Cloud Computing, dont chacune peut influencer une solution proposée par des chercheurs ou des fournisseurs de services Cloud. Ceux-là s'efforcent d'obtenir des taux de disponibilité de services conformes ou proches des taux standard déclarés par les experts du domaine. Après avoir exposé ces taux, nous avons introduit des mesures variées pour quantifier et évaluer la disponibilité des services dans les systèmes Cloud. Ensuite, nous avons identifié les coûts et les pertes importantes liées à l'indisponibilité et qui peuvent affecter les systèmes Cloud ainsi que leurs services. Puis, nous avons présenté les mécanismes de disponibilité les plus répandus dans la littérature et qui servent à protéger les services Cloud contre les défaillances à tous les niveaux. Par ailleurs, nous avons abordé les principales solutions et travaux de recherche associés à chaque mécanisme, qui visent principalement à améliorer la disponibilité et à maintenir les performances des systèmes Cloud Computing. Bien que ces techniques ont prouvé leur efficacité dans l'amélioration de la disponibilité, elles présentent encore diverses limitations et défis à relever. Cette thèse adopte une voie de recherche alternative en vue de proposer une approche permettant de répondre aux exigences de la disponibilité des services dans un environnement CloudP2P. Le chapitre suivant porte sur cette voie, laquelle préconise une politique différente consistant à utiliser des techniques de Clustering pour maintenir la disponibilité et gérer efficacement la dynamique et l'hétérogénéité des systèmes Cloud.

Chapitre 5 : Clustering pour la Disponibilité dans le Cloud-SaaS

Sommaire

1. Introduction
2. Data Mining
   - 2.1. Définition
   - 2.2. Processus de Découverte de Connaissance dans une Base de Données
   - 2.3. Stratégies et Techniques du Data Mining
   - 2.4. Applications du DataMining
3. Data Mining et Cloud Computing
   - 3.1. Cloud Computing basé sur Data Mining
     - 3.1.1. Data Mining dans le modèle SaaS
     - 3.1.2. Modélisation d'un Processus d'Exploration De Données SaaS
   - 3.2. Data Mining basée sur le Cloud Computing
     - 3.2.1. Data Mining basé sur Cloud SaaS
4. Clustering
   - 4.1. Définition
   - 4.2. Distance et Similarité
   - 4.3. Approches du Clustering
   - 4.4. Méthodes de Clustering
     - 4.4.1. Clustering par Partitionnement
     - 4.4.2. Clustering Hiérarchique
   - 4.5. Techniques de Validation de Clustering
     - 4.5.1. Validation Externe
     - 4.5.2. Validation Interne
5. Clustering pour la Disponibilité dans le Cloud Computing
6. Conclusion

1. Introduction

La disponibilité est reconnue comme étant l'une des caractéristiques essentielles pour maintenir la qualité de service dans les environnements de Cloud Computing, en particulier le modèle SaaS. En effet, les fournisseurs de services SaaS doivent être en mesure de répondre aux exigences de disponibilité de leurs solutions afin de fournir des services fiables et efficaces à leurs utilisateurs. Dans un tel contexte, il est nécessaire d'étudier les évolutions technologiques récentes et les travaux existants dans la littérature scientifique afin de pouvoir proposer une solution adéquate au problème de disponibilité des services dans les environnements de Cloud Computing, notamment le modèle SaaS. En ce sens, l'intégration de techniques de Clustering issues du Data Mining dans le Cloud Computing - SaaS- est devenue monnaie courante qui peut favoriser une fourniture appropriée de services SaaS, tout en offrant agilité et accès rapide à la technologie. Cette voie de recherche constitue la base de la présente étude, dont nous présenterons d'abord, dans la section 2, les principaux concepts du Data Mining. Ensuite, la section 3 expliquera la manière dont le Data Mining et le Cloud Computing, notamment le modèle SaaS, peuvent se servir mutuellement. Elle décrira particulièrement le processus d'intégration du Data Mining pour améliorer les services SaaS dans le Cloud Computing. Nous présenterons dans la section 4 les concepts fondamentaux qui sous-tendent la méthode de Clustering, en commençant par sa définition, suivie des mesures de distance et de similarité communément utilisées dans la littérature pour le Clustering. Ensuite, un aperçu des principales approches et techniques et des algorithmes de Clustering adoptés dans notre recherche sera établi. Par ailleurs, diverses approches de validation des solutions de Clustering seront discutées. Enfin, en utilisant notamment des techniques de Clustering, nous présenterons dans la section 5 un état de l'art des solutions et travaux de recherche importants, appliqués pour la disponibilité dans le Cloud Computing, particulièrement dans le modèle SaaS.

2. Data Mining

Selon ***Digital Around The World*** en Janvier 2021, 59.5% de la population mondiale utilise désormais l'Internet (4,66 milliards de personnes) en tant que système informatique innovant de communication qui est devenu de jour en jour un lieu important. Cette direction a permis de propulser et d'accumuler d'énormes quantités de données au fil du temps avec un manque de connaissance, ce qui a rendu la situation pathétique. Par conséquence, l'activité générique informatique a été mise face à la nécessité de stocker et traiter chaque jour une quantité de données d'une complexité gigantesque en termes de volume (quantité de données), de vitesse et de variété (internes ou externes, structurées, non structurées et semi-structurées) (Kautkar 2014) (Jain et Srivastava 2014).

La question est de savoir comment analyser intelligemment des données aussi abondantes de manière efficace? Dans ce sens, le raisonnement humain s'appuie à la base sur des connaissances et des expériences antérieures pour analyser et comprendre ce qui se passe autour de nous ou parfois prédire les actions futures possibles. Cet esprit cognitif intelligent a été suivi comme inspiration pour la naissance de la base du concept de l'exploration de données (Mukherjee et al. 2015).

2.1. Définition

L'exploration de données (en anglais Data Mining, abrégé DM) est apparue comme étant l'un des domaines les plus spectaculaires et les plus exigeants de notre époque, englobant une variété de techniques et d'outils pour répondre aux besoins de cette dernière. Elle permet de découvrir des tendances polyvalentes et des schémas utiles dans les données qui manquent de connaissances.

Bien que les concepts d'exploration de données ont été introduits il y a longtemps, le terme « Data Mining » a été largement accueilli et accepté au milieu des années 1990. À l'heure actuelle, l'exploration de données a fait ses preuves dans le domaine de la recherche et du traitement de l'information et il n'y a aucune incertitude qu'à l'avenir, il jouera un rôle primordial dans de nombreux domaines (Kautkar 2014) (Karahoca et al. 2012). Selon le groupe Gartner (Gartner 2022) et les auteurs de (Jain et Srivastava 2014):

> *«L'exploration de données est le processus qui consiste à analyser et découvrir des corrélations, des modèles, des règles et des tendances significatives dans une grande quantité de données stockées dans des dépôts, en utilisant des moyens automatiques ou semi-automatiques, des technologies de reconnaissance de modèles ainsi que des techniques statistiques et mathématiques.»*

L'exploration de données est un processus qui porte sur l'analyse des données recueillies afin d'extraire des informations implicites autonomes et de découvrir des corrélations, des règles et des connaissances potentiellement utiles à partir de grands ensembles de données précédemment collectés pour former des modèles informatiques qui soient capables de s'adapter à des situations particulières. De ce fait, l'incorporation des outils et des techniques Data Mining peut aider à tirer des leçons, optimiser l'utilisation des informations, réduire les coûts, améliorer la recherche, prendre des décisions et faire des projections futures en transformant les données rigides en données utiles dans de nombreux domaines d'application, dont l'informatique, le génie, les mathématiques, la physique, les neurosciences , la science cognitive, les banques, les assurances, la médecine et la vente.

Le Data Mining est apparu comme étant un sous-domaine interdisciplinaire de l'informatique.Sa naissance n'est fortuite.Elle est issue de la consolidation efficace de divers domaines qui ont mûri avec le temps, dont l'apprentissage machine, l'intelligence artificielle, les statistiques, la gestion des données et des bases de données, la recherche d'information et la visualisation. Tous ces domaines ont un impact significatif sur l'exploration des données, et ce, par leurs contributions à la recherche de solutions efficaces à de nombreux problèmes complexes ainsi que pourla découverte et la présentation de connaissances significatives et compréhensiblespar les humains. D'autre part, la recherche en exploration de données a donné lieu à une grande variété de techniques d'apprentissage qui ont le potentiel de régénérer plusieurs domaines scientifiques et industriels (Kautkar 2014) (Karahoca et al. 2012).

2.2. Processus de Découverte de Connaissance dans une Base de Données

Le processus d'exploration de données, connu sous le nom « découverte de connaissances dans une base de données » (en anglais *KnowledgeDiscovery in Databases*, abrégé KDD),est considéré comme étant un processus d'analyse avancée et d'extraction non triviale qui vise à trouver des modèles cachés et des informations implicites parmi le vaste ensemble de données disponibles,et à les interpréter en connaissances et en informations utiles. Le KDD est un processus itératif. Après la présentation des connaissances à l'utilisateur, les mesures d'évaluation peuvent être améliorées, l'exploitation de données peut être affinée, de nouvelles données peuvent être sélectionnées ou transformées, et de nouvelles sources de données peuvent être intégrées, le tout dans le but d'avoir de meilleurs résultats. Les gens confondent souvent entre «l'exploration des données» et « découverte de connaissances dans les bases de données», et les traitent en tant que synonymes, sachant que l'exploration des données est considérée comme étant une étape du processus itératif de découverte des connaissances dans une base de données. Ce système est fondé sur la fusion des sept étapes successives clés (figure 5.1.) (Kaushal et al. 2015) (Kautkar 2014) (Mehta 2017) (Mukherjee et al. 2015) (Sharma et al. 2014) (Dhote et Deshpande 2016):

Figure 5. 1: Processus de Découverte de Connaissances dans les Bases de Données.

1) ***Nettoyage des données:*** cette étape est précédée par un pas préliminaire consistant en la "***collecte des données***", durant lequel un recueil des données est effectuéà partir de sources hétérogènes internes ou externes. Les données collectées peuvent contenir du bruit et des informations incohérentes. La phase de nettoyage des données permet de se débarrasser du bruit et des données non pertinentes des données brutes collectées.
2) ***Intégration des données:*** cette phase permet de combiner et d'assembler l'ensemble des données brutes à partir de différentes sources afin de former une base de données cibles.

Il est courant de combiner les deux étapes dites "***nettoyage des données***" et "***intégration des données***". Ces étapes peuvent être effectuées ensemble en tant que phase de prétraitement pour générer une base de données.

3) ***Sélection des données:*** cette étape permet de récupérer les données appropriées (données cibles) de la base de données pour les analyser et les traiter.
4) ***Transformation des données:*** cette étape permet de cimenter les données dans des formats standards appropriés pour effectuer l'exploitation.
5) ***Exploration de données:*** cette étape s'appuie sur les différents concepts et les différentes techniques intelligentes afin de tirer les connaissances, les règles et les modèles utiles en matière de données.
6) ***Évaluation des modèles :*** cette étape permet d'identifier les modèles intéressants représentant les connaissances sur la base d'un ensemble de mesures.
7) ***Représentation des connaissances:*** cette étape conclue le processus KDD. C'est durant celle-ci que des techniques de visualisation et de représentation des connaissances telles que les graphiques sont utilisées pour visualiser les données dans un format utile afin de faciliter la compréhension et l'interprétation des résultats obtenus suite à l'exploration de données.

2.3. Stratégies et Techniques du Data Mining

L'exploration des données a fourni une direction pour traiter de manière efficace une énorme quantité de données ayant une grande complexité. Il existe deux fins fondamentales du processus d'exploration de données, à savoir la prédiction et la description, chacune reposant sur une variété de stratégies et des techniques de conception différentes, comme le montre la figure 5.2. (Kautkar 2014) (Mukherjee et al. 2015).

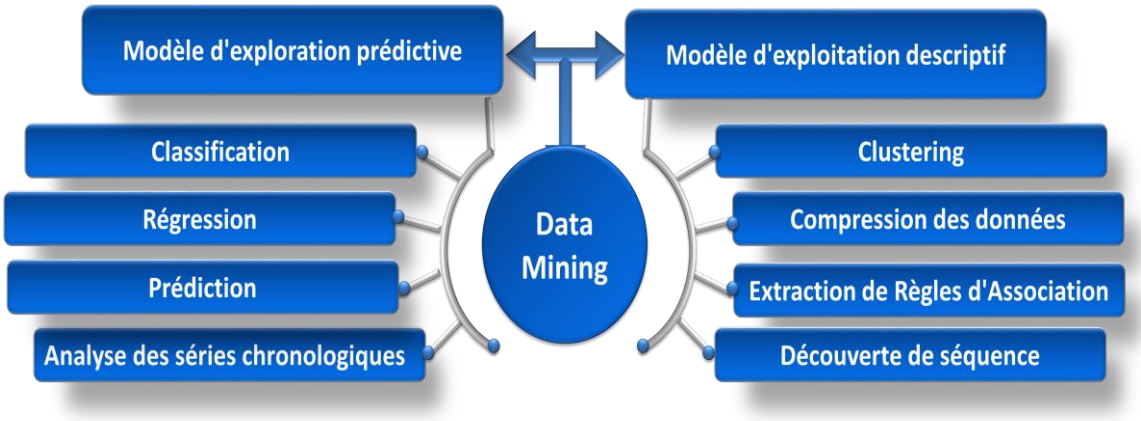


Figure 5. 2: Objectifs et Stratégies du Processus d'Exploration des Données.

1. Analyse Prédictive

L'analyse prédictivepermet de développer des modèles d'inductions et de classifications futures pour de nouvelle données en se basant sur les valeurs de certaines variables des ensembles de données et des connaissances actuelles ou connues à l'avance (historiques), afin de se projeter sur un état avant qu'il se produise (Karahoca et al. 2012) (Jain et Srivastava 2014). Cette stratégie d'exploitation utilise des fonctions d'apprentissage supervisé pour extraire et prédire les connaissances souhaitées en fournissant des informations sur« ce qui pourrait arriver » et « pourquoi cela pourrait arriver ». Plusieurs techniques prédictives sont utilisées en fonction de la nature changeante des données et des objectifs des tâches (Kautkar 2014) (Hussain 2017) (Mukherjee et al. 2015).

- ***Classification***

La classification, souvent appelée «apprentissage supervisé»,fait référence à un processus visant à développer le meilleur modèle ou à trouver la fonction la plus efficace qui permette de partitionner et de mettre en correspondance les données d'entrée en classes prédéfinies afin de mieux prédire la classe des nouvelles données.Les données d'entrée pour la classification, également appelées «données d'entraînement» ou «données d'apprentissage», sont associées avec des étiquettes de classe de l'ensemble. Le processus de la classification permet d'acquérir les meilleures connaissances possibles en explorant les données d'apprentissage pour former le modèle qui soit le plus à mêmed'attribuer des étiquettes de classe à des données non étiquetées (Karahoca et al. 2012) (Kautkar 2014). Il existe plusieurs modèles de classification dont nous citons les arbres de décision, les réseaux de neurones, les algorithmes génétiques, les classificateurs bayésiens, etc (Hussain 2017) (Mukherjee et al. 2015).

- ***Régression***

La régression est une méthode statistique généralement utilisée pour mapper des éléments des données afin d'analyser etprédire les tendances futures en fonction des valeurs de données passées et présentes ainsi que pour modéliser la relation entre une ou plusieurs variables (Kabilan et Jayaveeran 2015) (Karahoca et al. 2012) (Kaushal et al. 2015) (Mukherjee et al. 2015).

- ***Prédiction***

La prédiction est une technique permettant d'examiner les corrélations entre les variables dépendantes et indépendantes et de prédire les valeurs futures des variables de sortie inconnues ou manquantes. Cela signifie que la prédiction est utilisée pour prédire l'absence des valeurs de données numériques non disponibles plutôt que des étiquettes de classe (Kabilan et Jayaveeran 2015) (Kautkar 2014) (Mukherjee et al. 2015) (Sharma et al. 2014).

- ***Analyse des séries chronologiques***

En se basant sur des techniques statistiques, l'analyse des séries chronologiques permet de modéliser et de tracer un ensemble de points de données dépendants du temps. L'objectif principal du processus est de générer des prédictions et d'évaluer les tendances ou évènements futurs sur la base des événements passés connus. Exemple : l'analyse des conditions météorologiques (Kabilan et Jayaveeran 2015) (Mukherjee et al. 2015).

2. Analyse Descriptive

L'analyse descriptive se réfère à l'exploitation et l'inspection de l'ensemble des données et la description de leurs caractéristiques afin de mieux comprendre les événements passés ou récents. Dans les modèles descriptifs, l'exploration de données se concentre sur la découverte de modèles et d'associations intéressants qui relient les données et fournissent des synthèses utiles (Kautkar 2014) (Karahoca et al. 2012) (Jain et Srivastava 2014). Il existe plusieurs techniques d'exploration descriptive, nous citons parmi celles qui sont les plus populaires:

- ***Clustering***

Le Clustering, également connu sous le nom « classification non supervisée », est un processus descriptif qui permet la segmentation de l'espace de données selon leurs caractéristiques afin de former un ensemble de groupes appelées « clusters » et qui maximisent les similarités intra-clusters et minimisent les similarités inter- clusters. Il permet également d'identifier les régions denses et éparses dans l'espace de données et de découvrir des modèles de distribution et des corrélations intéressantes entre les attributs des données (Hussain 2017) (Kautkar 2014) (Karahoca et al. 2012) (Mukherjee et al. 2015) (Sharma et al. 2014). De nombreux algorithmes de Clustering ont été développés. Ils sont regroupés dans deux catégories principales : méthodes basées sur le partitionnement (K-Means, K-Medoids, PAM, etc.) et méthodes basées sur la hiérarchie (BIRCH, ROCK, HAC, CURE, etc.).

- ***Compression des données***

La compression des données (en anglais *Summurization*) est une technique qui permet de trouver une description compacte pour un sous-ensemble de données et de former l'abstraction des données. Elle comprend diverses méthodes telles que la moyenne, la moyenne pondérée et l'écart-type pour mesurer la dispersion des données (Karahoca et al. 2012) (Mukherjee et al. 2015) (Kabilan et Jayaveeran 2015).

- ***Extraction de Règles d'Association***

L'extraction de règles d'association est une méthode qui consiste à analyser les grands ensembles de données pour détecter des modèles et des relations fréquentes entre les éléments de données. La règle d'association utilise des expressions de la forme $X \rightarrow Y$, où X et Y sont des ensembles d'éléments, qui sont basées essentiellement sur deux concepts clés: *support* et *confiance*, pour déterminer l'utilité de la règle. Les règles d'association les plus utilisées sont l'algorithme Apriori, fp-tree, AprioriTid et l'algorithme hybride Apriori (Hussain 2017) (Kautkar 2014) (Karahoca et al. 2012) (Mukherjee et al. 2015) (Sharma et al. 2014).

- ***Découverte de séquence***

La découverte de séquence (en anglais *SequenceDiscovery*) est une technique analogue à laméthode des règles d'association, à l'exception que les relations et les associations entre les séquences de données sont établies en associant à chaque séquence de données sa propre chronologie d'événements organisés et liés par le temps.La découverte de ces séquences nécessite la découverte des associations séquentielles entre les éléments de la base de données des séquences et la sauvegarde des détails de chaque transaction ainsi les acteurs identifiés (Karahoca et al. 2012) (Mukherjee et al. 2015) (Kabilan et Jayaveeran 2015).

2.4. Applications du Data Mining

L'exploration de données joue un rôle important dans des domaines panoptiques et divers admettant une grande quantité de données intégrales et très sensibles. Ellea émergé en tant que nouvelle technologie puissante, d'où son intégration dans les activités quotidiennes et le fait qu'elle soit devenue courante. L'exploration de données a rapporté un gain positif pour les entreprises.Ces dernières sont devenues plus efficaces grâce à l'utilisation de cette technologie ayant un grand potentiel et pouvant les aider à se concentrer sur les informations les plus importantes parmi celles inscrites dans leurs grandes bases de données et de réduire ainsi leurs coûts. Parmi les applications de l'exploration de données, on retrouve: Web Mining, Text Mining (Knowledge Discovery in Text ou KDT), Multimedia data Mining, la détection des fraudes, l'analyse du marché, l'exploration de données éducatives, l'exploration de données omniprésente (UDM), Réseau et télécommunications et Spatial Data Mining (Ambulkar et Borkar 2012) (Hussain 2017) (Kautkar 2014).

3. Cloud Computing et Data Mining

Le Cloud Computing est apparu comme un paradigme puissant qui a réussi à dominer les services de réseau dans de nombreux domaines d'application et à transformer le secteur informatique de manière significative. Le SaaS constitue le roi des modèles de prestation de services Cloud. Il a été adopté de manière drastique par de nombreuses personnes et organisations au cours de la dernière décennie. Dans une mesure tout aussi importante, ces dernières années, les techniques et les outils d'exploration de données ont progressé et sont devenus incontournables. En tant que tendance future de la recherche, le couplage des algorithmes de Data Mining avec la technologie du Cloud Computing présente un avantage et un potentiel énormes. Il peut fournir des solutions pour surmonter certains problèmes posés par les deux paradigmes (Kaushal et al. 2015).

3.1. Cloud Computing basé sur Data Mining

Le Cloud Computing désigne la nouvelle tendance des services de l'Internet qui s'appuient sur des nuages de serveurs situés à distance pour gérer les tâches. Il s'agit d'un domaine essentiel sur lequel l'exploration de données doit se concentrer. L'exploration de données dans le Cloud Computing est le processus d'extraction d'informations structurées à partir de sources de données Web non structurées ou semi-structurées. L'exploration des données dans le Cloud Computing permet aux utilisateurs de récupérer des informations utiles à partir d'un entrepôt de données virtuellement intégré dans le but de prendre des décisions efficaces et prédire les tendances et les comportements futurs, ce qui réduit les coûts de l'infrastructure et du stockage. Avec cette intégration, on peut trouver les informations souhaitées sur le comportement, les habitudes, les centres d'intérêt et la situation géographique des clients en quelques clics de souris.

Le Cloud Computing fait référence à la fourniture de ressources informatiques sur l'Internet. Au lieu que le stockage de données, la maintenance et la mise à jour des matériels et logiciels soient faits localement par les utilisateurs, ils seront fournis par les serveurs Cloud en tant que service à ces utilisateurs via l'Internet. L'exploration de données dans le Cloud Computing permet aux organisations de centraliser la gestion des logiciels et du stockage des données, avec la garantie de la fourniture de services rentables, efficaces, fiables et sécurisés par les fournisseurs Cloud au profit de leurs utilisateurs finaux (Ambulkar et Borkar 2012) (Kabilan et Jayaveeran 2015) (Kaushal et al. 2015) (Dhote et Deshpande 2016) (Petre 2012).

Le Cloud Computing est un modèle évolué d'informatique distribuée. L'intégration et le développement des techniques automatiques d'extraction des modèles et des connaissances dans le Cloud Computing permettent d'avoir des Cloud intelligents en améliorant la qualité des services fournis aux utilisateurs. Les entreprises et les clients utilisent ces services pour obtenir des informations sur différents sujets et prendre des mesures basées sur les données présentées. Les travaux et les recherches effectués ont prouvé que les algorithmes du Data Mining jouent un rôle vital dans l'exploration des données dans un environnement de Cloud Computing, qui offre un potentiel remarquable d'analyse et d'extraction d'informations et de connaissances utiles dans différents domaines d'activités humaines: commerce, réseaux, économie, médecine, biologie, pharmacie, publicité, etc (Ambulkar et Borkar 2012) (Kabilan et Jayaveeran 2015).

3.1.1. Data Mining dans le modèle SaaS Cloud

L'utilisation de Data Mining consiste en l'application de techniques telles que les règles d'association, l'apprentissage machine et le Clustering, sur un ensemble de données brutes à des fins d'exploration. L'utilisation du Data Mining dans un Cloud SaaS rivette la technique qui pourrait prédire le comportement de l'utilisateur et reconnaitre son profil chaque fois qu'il interagit avec un service afin d'offrir une meilleure expérience personnalisée. Les données de sortie sont visualisées en termes d'interactivité avec l'application/le logiciel. Les données d'utilisation peuvent être représentées par des tableaux ou des graphiques relationnels qui seront utilisés par la suite par les entreprises ou les développeurs d'applications pour prendre des décisions éclairées et prévoir les tendances et les comportements futurs. L'intégration des techniques et des applications d'exploration de données est très nécessaire dans le modèle SaaS du Cloud Computing. Étant donné que les services SaaS sont fournis via l'Internet et que les données des utilisateurs sont stockées sur le site du fournisseur, ce dernier doit être capable d'assurer la fourniture des services flexibles, sécurisés et hautement disponibles pour répondre aux besoins de ses utilisateurs. Ces besoins sont généralement négociés par le biais d'un contrat de niveau de service conclu entre le demandeur et le fournisseur de service pour garantir les performances des services Cloud.

Les algorithmes de Data Mining permettent l'évaluation de la qualité des services logiciels SaaS potentiels sur le Cloud Computing en fabriquant différents modèles d'exploration en fonction des besoins du client. Ces algorithmes sont très utiles aux fournisseurs de services logiciels pour évaluer leurs propres services afin de régler certains problèmes remarquables dans les services SaaS et d'accroître leur disponibilité et performance dans l'environnement Cloud, en fonction des demandes et des besoins des utilisateurs du Cloud. Ils aident également les utilisateurs du Cloud à être plus agiles, d'accéder rapidement à la technologie et d'évaluer leurs propres satisfactions des services logiciels potentiels, disponibles dans l'environnement de Cloud Computing. Cependant, l'intégration de l'exploration de données dans les services SaaS du Cloud Computing devrait être assez solide pour être en mesure de supporter la variation et l'augmentation de la quantité de données produites et contribuera à une exploitation efficace de ces données (Jagli et Gupta 2013) (Kautkar 2014) (Parekh 2014) (Petre 2012).

3.1.2. Modélisation d'un Processus d'Exploration de Données SaaS

Un processus d'exploration de données SaaS peut être modélisé à travers un ensemble d'étapes successives (figure 5.3.). Ces étapes doivent être précédées par d'autres de prétraitement, qui sont:

- Extraction des données SaaS.
- Transformation des données SaaS.
- Application Des Algorithmes De Data Mining.
- Obtention Des Résultats d'Exploration.

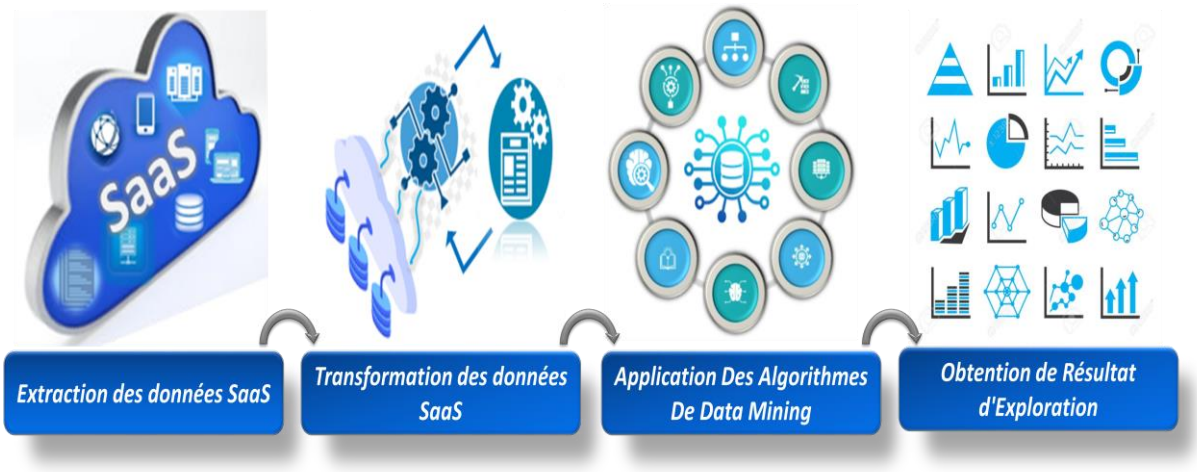


Figure 5. 3: Étapes du Processus d'Exploration de Données SaaS.

1) ***Extraction des données SaaS***
Les données SaaS nécessaires pour un tel traitement sont généralement distribuées un peu partout et ne sont stockées dans aucun type de référentiel de données. Par conséquence, les concepteurs doivent gérer et analyser efficacement des données distribuées et de taille énorme, ce qui constitue un défi pour eux. Ainsi, ils doivent être en mesure d'extraire et de recueillir seulement les données SaaS source nécessaires et de les séparer des données source réelles.

2) ***Transformation des données SaaS***
Cette étape s'appuie sur l'utilisation des méthodes appropriées pour traiter les incohérences et les bruits avec les données SaaS collectées et de les transformer dans un format standard. Après la phase d'extraction des données SaaS, un nettoyage est appliqué sur ces données afin de se débarrasser du bruit et des données non pertinentes. Etant donné que l'on est dans un environnement dynamique et évolutif, les données SaaS de flux sont de volume énorme et en évolution rapide et continue, ce qui implique beaucoup de difficulté durant leur stockage dans des entrepôts de données. Pour cela, il est essentiel de trouver les modèles intéressants parmi les données brutes en effectuant une analyse multidimensionnelle sur des mesures agrégées telles que la somme et la moyenne. Ensuite, les données résultantes doivent être assemblées dans des formats standards appropriés.

3) ***Application Des Algorithmes De Data Mining***
Durant cette phase, les données SaaS stockées dans des formats multidimensionnels sont chargées pour être utilisées comme entrées à divers algorithmes et méthodes de data Mining. Selon les besoins prédéfinis du modèle voulu, une ou des techniques d'exploration de données descriptives et/ou prédictives sont sélectionnées et appliquées par la suite sur les données SaaS. Après application de l'algorithme d'exploration de données, la sortie sera générée en termes de performances par les services SaaS fournis sur le Cloud par des fournisseurs de services.

4) ***Obtention de Résultats d'Exploration***

Enfin, les résultats obtenus peuvent être utilisés par les fournisseurs des services pour analyser le modèle développe et déterminer quelle méthode d'exploration de données est la plus efficace et la mieux adaptée à des données SaaS données en termes de performances de réponse. D'un autre côté, ils aideront les utilisateurs de services à prendre une décision adaptée à leurs besoins en matière de services logiciels.

3.2. Data Mining basée sur le Cloud Computing

Le développement rapide des technologies de l'information et de la capacité de stockage a conduit à l'accumulation d'une énorme quantité des données diversifiées. Dans un besoin réciproque à ce qui a été discuté dans la sous-section précédente, le traitement massif de ces données est devenu un problème important dans le domaine du Data Mining. Dans le passé, le processus d'exploration de données s'opérait sur une machine à haute performance ou un gros appareil informatique, ce qui est devenu inefficace. Par conséquent, la question à résoudre est de savoir comment améliorer le parallélisme et l'efficacité des algorithmes de l'exploration des données. Dans de telles circonstances, le Cloud Computing est un support puissant et un moyen efficace pour surmonter les problèmes de l'exploration de données grâce à son énorme capacité de traitement des données volumineuses, de stockage, de calcul et des changements élastiques. En incluant le Cloud Computing dans l'exploration de données, les grands problèmes de traitement et de stockage de données ainsi quede frais de possession des outils et des techniques d'exploration de données seront résolus.

Le Cloud Computing adopte le même principe de fourniture de logiciels et matériels informatiques pour fournir une tache d'exploration de données où ce dernier est fourni en tant que service sur l'Internet. Le système d'exploration de données basé sur le Cloud Computing « en anglais Data Mining based on Cloud Computing, abrégé DMC » est une approche construite sur le «Cloud» et axée sur les services, fournissant une interface de services d'exploration de données à une variété d'utilisateurs. Les utilisateurs peuvent utiliser les différents services via l'interface fournie par le système sans aucune nécessité de pré-connaissance sur le fonctionnement du système et d'inquiétude sur la capacité de calcul et de stockage de ce dernier. Ils doivent simplement sélectionner l'algorithme approprié pour traiter leurs données et l'exécuter pour obtenir les résultats de l'exploration de données. Dans une telle approche basée sur le Cloud, le système d'exploration de données basé sur le Cloud Computing peut adopter une méthode de paiement à l'utilisation, ce qui permet aux entreprises ou particuliers d'obtenir le service voulu directement et de se débarrasser des dépenses d'achat de logiciels et des obstacles qui les empêchent de bénéficier des techniques d'exploration de données (Kabilan et Jayaveeran 2015) (Kaushal et al. 2015) (Ying et al. 2016) (ZENG 2018).

Les services d'exploration de données basée sur le Cloud Computing peuvent aborder un procédé particulier dans le processus d'exploration de données, une technique d'exploration de données, des modèles distribuées d'exploration des données et d'un processus KDD. Les auteurs dans (Talia et Trunfio 2010) ont résumé quatre niveaux de service d'exploration de données dans le Cloud Computing, comme la montre la figure suivante:

Figure 5. 4: Les Niveaux des Services du DMC.

1) ***Etape KDD Unique*** *(en anglais Single KDD Step)**:*** à ce niveau, toutes les étapes incluses dans le processus KDD comprenant la sélection, la représentation et la visualisation qui sont exprimées en tant que services.
2) ***Tâches uniques d'exploration de données*** *(en anglais Single Data Mining Tasks)**:*** il inclut des services d'exploration de données séparés, comme la prédiction, la classification, le Clustering et la découverte des règles d'association.
3) ***Modèles d'exploration de données distribuées*** *(en anglais Distributed Data Mining patterns)**:*** à ce niveau, des méthodes d'exploration de données distribuées sont mises en œuvre, telles que l'apprentissage collectif et la classification parallèle en tant que service.
4) ***Processus KDD*** *(en anglais Process KDD)**:*** ce niveau comprend les tâches et les modèles précédents composés dans un processus de travail multi étapes.

Cette conception est incrémentale où la mise en œuvre des services dans un niveau s'appuie sur les services des niveaux qui le précède, ce qui permet une réutilisation des procédés, des techniques et des modèles déjà disponibles. Ce cadre peut permettre de développer des outils distribués d'exploration de données en tant que composition des services uniques accessibles à tout moment et de partout (Talia et Trunfio 2010).

En termes de dépense d'utilisation des techniques d'exploration de données, l'intégration du Cloud dans l'exploration de données offre un avantage pour les petites entreprises en leur permettant de disposer la possibilité de louer un service d'exploration de données dans le Cloud Computing à un frais minimal pour une analyse efficace de toutes les données de l'organisation qui étaient auparavant réservées seulement aux grandes entreprises (Kaushal et al. 2015).

3.2.1. Data Mining basé sur Cloud SaaS

Le modèle SaaS (Logiciel en tant que Service) est considéré comme étant le roi de tous les services Cloud. Le modèle SaaS permet de réduire les coûts en offrant des options de licence flexibles et en externalisant l'effort matériel. Dans le SaaS, les services sont gérés de façon centralisée et sont accessibles à distance sur le Web. Autrement dit, aucun composant logiciel n'est installé chez le client. Il se trouve sur le serveur d'un fournisseur de services logiciels qui s'occupe du matériel et des mises à jour logicielles et assure la maintenance technique. La fourniture des services SaaS est maintenue selon une architecture à multi-locataire qui implique une seule instance physique avec des clients hébergés dans des espaces logiques séparés, et selon une architecture où il peut y avoir de multiples

variations dans la façon dont une seule instance est réellement mise en œuvre et comment la multi-location est réellement réalisée (Ambulkar et Borkar 2012).

Etant donné que le SaaS fait référence aux logiciels informatiques fournis en tant que services sur l'Internet, dans le modèle SaaS du Cloud Computing, les logiciels d'exploration de données sont également fournis de cette façon. L'intégration du modèle SaaS dans l'exploration de données permet aux fournisseurs de définit des logiciels qui soient capables de traiter et de fournir une tache d'exploration de données aux utilisateurs en tant que service à la demande via l'Internet. Cela se reflète positivement sur les clients en termes de réduction des coûts d'utilisation des outils et des techniques d'exploration de données. D'une part, le client ne paie que pour les outils de Data Mining dont il a besoin, ce qui lui permet d'économiser beaucoup d'argent par rapport aux suites complexes de Data Mining qu'il n'utilise pas de manière exhaustive. D'un autre côté, il ne paie que les coûts générés par l'utilisation du Cloud. Il n'a pas besoin de maintenir une infrastructure matérielle, il peut appliquer le data Mining simplement via son navigateur. Cela réduit les obstacles qui empêchent les petites entreprises de tirer profit de l'exploration de données (Ambulkar et Borkar 2012) (Petre 2012).

4. Clustering

Le Clustering, une tâche ayant été pratiquée par les humains des milliers d'années durant, a été entièrement automatisé ces dernières décennies. Cela a été possible grâce aux avancements des technologies de calcul. L'apparition de l'analyse remonte vers les années 1930. Elle s'est développée et est devenue un outil majeur dans les années 1960 et 1970. Elle est devenue, à ce jour, un instrument capable de résoudre des problèmes informatiques complexes. Le terme « Clustering » désigne, dans son sens le plus général, la méthodologie qui consiste à répartir des éléments de données dans des groupes selon certaines caractéristiques communes. Dans cette section, nous établissons les bases de l'étude du Clustering.

4.1. Définition

Le Clustering, connu également sous les termes « analyse en cluster », «classification non supervisée», « partitionnement», «segmentation» ou «regroupement», est considéré comme étant l'une des principales techniques d'analyse exploratoire et descriptive des données et de découverte de connaissances. Il est utilisé principalement pour étudier la structure interne portant sur de grands ensembles de données complexes dont la structure sous-jacente est inconnue (Ayram et Karkkainen 2006).

Le Clustering fait référence à un éventail de méthodes qui cherchent à regrouper les instances d'un jeu de données en sous-ensembles de manière à ce que les instances ayant des caractéristiques similaires soient regroupées dans le même groupe (ou cluster), tandis que les instances dissemblables (ou dissimilaires) soient regroupées dans des groupes différents. Les méthodes typiques de regroupement utilisent une fonction de similarité, généralement basée sur la distance, pour comparer diverses instances de données. Dans ce cadre, les clusters sont établis de sorte à ce que les instances de données au sein d'un cluster aient des valeurs de distance minimales et que lesinstances de données entre différents clusters aient des valeurs de distance maximales.

En tant que tâche totalement non supervisée, le Clustering calcule la similarité entre les instances de données sans avoir d'informations sur leur distribution correcte (également connue sous le nom de la vérité de base) (Hassani et Seidl 2017). En d'autres termes, le Clustering vise à partitionner un ensemble de données non étiquetées en un certain nombre de clusters disjoints, en se basant sur des caractéristiques spécifiées. Dans ce contexte, chaque cluster est considéré comme étant une collection

de données qui ont une forte ressemblance entre elles et une faible ressemblance avec les données appartenant aux autres clusters (Ayram et Karkkainen 2006). Par conséquent, un cluster peut être traité comme une classe implicite dans le jeu de données. En ce sens, le Clustering est parfois appelé classification automatique, pouvant trouver automatiquement les groupes précédemment inconnus au sein d'un ensemble de données (Han et al. 2012). Il permet d'identifier les régions denses et éparses dans l'espace de données et de découvrir des modèles de distribution et des corrélations intéressantes entre les attributs des données. Cela explique la segmentation de l'espace des données, ce qui résulte la formation d'un ensemble de clusters où les similitudes intra-classe sont maximisées et les similitudes inter-classe sont réduites au minimum en fonction des propriétés de données (Hussain 2017) (Kautkar 2014) (Karahoca et al. 2012) (Mukherjee et al. 2015) (Sharma et al. 2014).

Loin d'être un simple processus de calcul qui trouve un regroupement unique et définitif pour un ensemble des données, le Clustering permet d'organiser les instances de données en une représentation efficace qui caractérise la population échantillonnée. Il est perçu comme un support pour les chercheurs car leur offrant une compréhension qualitative et quantitative des grands ensembles de données multi-variées (Ayram et Karkkainen 2006). Le Clustering est récemment devenu un sujet très abordé dans la recherche sur l'exploration de données en raison de la croissance des quantités de données collectées dans les bases de données multidimensionnelles. Il a fait l'objet d'un développement vigoureux en étendant ses racines dans de nombreux domaines d'application tels que la biologie, la médecine, la reconnaissance des formes d'images, la sécurité, la veille économique et la recherche sur le Web (Han et al. 2012).

4.2. Distance et Similarité

La similarité ou la dis-similarité est un concept fondamental utilisé par les algorithmes de Clustering afin de les aider à naviguer dans l'espace des données pour former des clusters. En fait, le Clustering est effectué sur la base d'une mesure de similarité définie sur l'espace des caractéristiques permettant de regrouper les éléments de données similaires dans les mêmes clusters, tout en plaçant les points de données dissimilaires dans des clusters différents (Shirkhorshidi et al. 2015). En ce sens, la similarité est une quantité qui reflète la force de la relation entre les instances de données, alors que la dis-similarité traite la mesure de la divergence entre ces instances. La mesure de similarité est généralement fondée sur une fonction de distance qui joue un rôle crucial dans les techniques de Clustering. Une fonction de distance, également connue sous le nom de fonction métrique, peut être définie comme la distance entre les instances d'un ensemble de donnéesservant à articuler la proximité entre celles-ci (Irani et al. 2016).

Soit X l'ensemble de N éléments de données tel que X= $\{x_i(a_1, a_2, \dots, a_Q) \in R^Q | i \in N\}$. Considérons x_b et $x_c$deux éléments de données distincts de X. Formellement, la distance $d(x_b,x_c)$ entre $x_b$et $x_c$est considéré comme une fonction à deux arguments satisfaisant les conditions suivantes :

1. Non-négativité: $d(x_b, x_c) \geq 0$
2. Symétrie: $d(x_b,x_c) = d(x_c,x_b)$
3. Inégalité triangulaire: $d(x_b,x_c) + d(x_c, x_d) \geq d(x_b, x_d)$ pour n'importe quels points x_b , $x_c$et x_d
4. $d(x_b,x_c)=0$, seulement si$x_b = x_c$

Sur la base de cette proximité, les techniques de Clustering peuvent décider d'affecter deux instances de données au même cluster ou à des clusters différents, où la mesure de distance donne de faibles valeurs lorsque $x_b$et x_c sont similaires et de grandes valeurs lorsque $x_b$et $x_c$sont dissemblables. De nombreuses mesures de distance ont été proposées dans la littérature en tant que mesures de proximité

pour le problème de Clustering. Le choix de l'une d'entre elles dépend des données étudiées et des objectifs variés des utilisateurs. Nous présentons dans les sous-sections suivantes un aperçu des principales mesures de distance couramment utilisées:

- ***Distance Euclidienne***

La distance euclidienne, également appelée «norme L_2» ou «distance L_2», est la méthode la plus répandue pour calculer la distance entre des éléments de données quantitatives dans un espace de caractéristique à Q-dimension (ou multidimensionnelle). Elle *représente la distance la plus courte entre deux points de données. La distance Euclidienne* détermine la racine carrée de la somme des carrés des différences entre les coordonnées d'une paire de données (Irani et al. 2016). On appelle distance euclidienne entre X_b et X_c la distance *obtenue par la fonction mathématique suivante:*

$$d(X_b, X_c) = \sqrt{\sum_{t=1}^{Q} (X_{b_t} - X_{c_t})^2} \quad (1)$$

- ***Distance Euclidienne Pondérée***

La distance Euclidienne pondérée « en anglais Weighted Euclidean Distance » est une version modifiée de la distance euclidiennequi peut être utilisée lorsque l'importance relative de chaque attribut est disponible (Shirkhorshidi et al. 2015). Autrement dit, elle permet d'accorder un poids à chaque attribut en fonction de son importance perçue. La distance euclidienne pondérée entre X_b et $X_c$peut être calculée comme suit :

$$d(X_b, X_c) = \sqrt{\sum_{t=1}^{Q} W_t \, (X_{b_t} - X_{c_t})^2} \quad (2)$$

Où $W_t$est le poids donné à la t^{ème} composante.

- ***La distance de Manhattan***

La distance de Manhattan (également appelée «norme L_1», «distance L_1» ou « city block distance ») peut être considérée comme une version approximative et moins coûteuse de la distance euclidienne. Elle représente une métrique de distance qui détermine la somme des différences absolues entre les coordonnées d'une paire de données (Amelio et Tagarelli 2019). Étant donné les deux points de données X_b et X_cà Q-dimensions, la distance de Manhattan est définie comme suit :

$$d(X_b, X_c) = \sum_{t=1}^{Q} \left| X_{b_t} - X_{c_t} \right| \quad (3)$$

- ***Distance de Minkowski***

La distance de Minkowski est la forme généralisée de la distance euclidienne et de la distance de Manhattan (Amelio et Tagarelli 2019). La distance de Minkowski entre deux points de données X_b et X_cà Q-dimensions est donnée par la formule suivante:

$$d(X_b, X_c) = \left[\sum_{t=1}^{Q} (X_{b_t} - X_{c_t})^{P} \right]^{\frac{1}{P}} \quad (4)$$

Où P est un nombre entier positifqui représente l'ordre de la norme. Dans l'équation (4) donnée ci-dessus, la distance Minkowski avec un ordre 1 et 2 (p=1, p=2) représente respectivement la distance de

Manhattan la distance Euclidienne. La distance de Tchebychev est une variante de la distance de Minkowski où p=∞ (en prenant une limite).

- ***Distance cosinus***

La similarité cosinusoïdale pour le Clustering mesure le cosinus de l'angle entre deux vecteurs projetés dans un espace multidimensionnel. Plus l'angle est petite, plus la similarité en cosinus est élevée.Étant donnéesX_b et X_c deux vecteurs à Q dimensionset θ l'angle entre ces deux vecteurs, la similarité en cosinus entre X_b et X_c est donnée par leur produit scalaire divisé par le produit de leurs normes (Shirkhorshidi et al. 2015). La distance cosinus entre X_b et $X_c$est égale à 1 moins le cosinus de θ et est obtenue comme suit:

$$d(X_b, X_c) = 1 - \cos(\theta) = 1 - \frac{X_b . X_c}{\|X_b\| . \|X_c\|} = 1 - \frac{\sum_{t=1}^{Q}(X_{b_t} - X_{c_t})}{\sqrt{\sum_{t=1}^{Q}(X_{b_t})^2}\sqrt{\sum_{t=1}^{Q}(X_{c_t})^2}} \quad (5)$$

- ***Distance de Jaccard***

La distance de Jaccard est utilisée pour identifier la similarité entre deux groupes du jeu de données. Elle représente le rapport entre le cardinal de l'intersection des deux groupes de données et le cardinal de leur union, où le cardinal est dénoté comme la taille ou le nombre d'éléments contenus dans l'ensemble résultant de l'intersection ou de l'union (Shirkhorshidi et al. 2015). Soit A et B deux groupes de données dont chacun rassemble une collection d'éléments de l'ensemble de données original, la distance de Jaccard est calculée via l'équation suivante:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

4.3. Approches de Clustering

Les approches de Clustering peuvent être subdivisées selon trois types de Clustering: Clustering *Dur*, Clustering *Flou* et Clustering *Doux*. Le *Clustering Dur* (en anglais, *Hard ou Crisp-Clustering)* permet de former une partition dure en divisant l'ensemble de *N* éléments de données en un nombre *K* de clusters disjointsde sorte que chaque élément de données appartienneà un et un seul cluster qui est jugé le plus proche en termes d'un critère de similarité défini.Formellement, soit $D = \{x_i\}_{i=1}^{N}$ un ensemble de *N* éléments de données dans R^n (espace euclidien à n dimensions) où X_i est le $i^{ème}$ vecteur de caractéristique; X_{ij} est la $j^{ème}$ caractéristique de X_i. La structure du regroupement durde *D* est représentée comme un ensemble *C*de *K* clusters avec,$C = \{C_1, C_2, \ldots, C_K\}$, tel que 2≤ K<N. La K-partition dure de *D* est un *K* sous ensemble de *D* de sorte que l'intersection de l'ensemble des clusters soit vide, c'est à dire$\forall a, b \in Ravec\ 1 \leq \underset{a \neq b}{a, b} \leq k$; $C_a \neq \emptyset; C_b \neq \emptyset;\ C_a \cap C_b = \emptyset$. En outre, l'union de l'ensemble des clusters permet la reproduction de l'ensemble original de données *D*, $\bigcup_{s=1}^{k} C_s = D$. Grace à sa facilité d'interprétation, le Clustering *Dur* fait partie des approches de Clustering les plus fréquentes. Néanmoins, le besoin de donner plus de flexibilité aux clusters peut s'avérer essentiel dans certains cas réels. Il peut arriver, dans un premier lieu, que certaines instances de données se distinguent des autres instances de manière trop significative. Le fait de leur assigner un cluster peut ainsi perturber le processus de Clustering. Il arrive parfois que ces instances soient rejetées et qu'aucun cluster ne leur soit attribué dans le résultat final. Il s'agit là de *« Clustering Dur Partiel »*, ce qui signifie que chaque instance de données n'appartient à aucun ou à un cluster (Hamidouche et Idjeraoui 2013). En deuxième lieu, il est parfois difficile de regrouper, de façon stricte, les instances d'un jeu de données complexes où les clusters se chevauchent (Aggarwal et Reddy 2013). En effet, la

nature floue des problèmes pratiques a conduit au développement de la catégorie de *Clustering Flou* (en anglais, *Fuzzy Clustering*), qui permet à chaque élément de l'ensemble de données d'appartenir à tous les clusters simultanément en introduisant le concept de degré d'appartenance. Cette idée a trouvé des terrains fertiles dans le domaine de la classification. Elle a été proposée en 1965 par Zadeh. Elle consiste à représenter la similarité entre chaque élément de données et les différents clusters en utilisant une fonction d'appartenance dont les valeurs varient entre zéro et un. Autrement dit, chaque élément de données est associé par un ensemble de degrés d'appartenance où chaque degré indique la force d'association entre cet élément de données et un cluster particulier. Dans ce contexte, on considère *C* la partition obtenue d'un processus de *Fuzzy Clustering ;* C= $\{C_1, C_2, \ldots, C_K\}$ tel que 2≤ K<N. La K-partition floue de *D* caractérise l'appartenance de chaque élément de données X_i à touts les clusters $\{C_1, C_2, \ldots, C_K\}$ en utilisant une fonction d'appartenance u_{ia} qui mesure le degré d'appartenance de l'élément de donnée X_i à chaque cluster $C_s \subseteq$C, et dont la valeur se situe entre zéro et un $u_{is} \epsilon [0,1]$. De plus, la somme des degrés d'appartenance pour chaque élément de données doit être égale à un $\sum_{s=1}^{k} u_{is} = 1$. L'union des K clusters résultant permet la reproduction de *D* de sorte que $\forall a, b \in R \; avec \; 1 \leq \begin{matrix} \text{a, b} \\ a \neq b \end{matrix} \leq$ k; $C_a \neq \emptyset$; $C_b \neq \emptyset$; $\bigcup_{s=1}^{k} C_s = D$, tandis que leurs intersections ne donnent pas forcément l'ensemble vide, c'est-à-dire $C_a \cap C_b = \emptyset$ n'est pas obligatoire (Bezdek et al. 1984). Il est notamment à noter que le développement de méthodes de *Fuzzy Clustering* a connu un progrès significatif.Il s'agit d'un concept très important dans le regroupement de données. Cependant, il a été estimé qu'en dépit du ce fait, un flou complet pourrait ne pas être souhaitable dans certains cas.Par conséquence, une autre catégorie d'algorithmes de Clustering a été développée. Connue sous le terme *Clustering Doux* ou «*Soft* ou *Semi-Fuzzy Clustering*», elle permet aux données d'appartenir à certains clusters mais pas nécessairement à tous. Cette idée a été introduite par S. Z. Selim et M. A. Ismail pour limiter le *Fuzzy Clustering* en restreignant l'appartenance des éléments de données à certains clusters plutôt qu'à tous (Behera et Chakravarty 2015) (Bezdek et al. 1984) (Kamel et Selim 1991) (Patel et Yadav 2016) (Selim et Ismail 1984). Par conséquence, on peut convertir une partition résultante *d'un Clustering Flou* ou *Doux* en partition dure en affectant chaque élément de données au cluster ayant le plus grand degré d'appartenance (Behera et Chakravarty 2015). Réciproquement, on peut considérer la partition résultante d'un *Clustering Dur* comme étant un cas spécial de *Clustering Flou* où chaque élément de données sera attribué un degré d'appartenance de 1 au cluster auquel appartient et un 0 pour les autres clusters:

$$u_s(X_i) = u_{is} = \begin{Bmatrix} 1; & X_i \in C_s \\ 0; & otherwise \end{Bmatrix} \quad (7)$$

4.4. Méthodes de Clustering

Le Clustering désigne un processus consistant à regrouper les objets d'un jeu de données en différents ensembles appelés « clusters ». Selon les spécialistes du domaine, il existe plusieurs définitions de la notion de cluster, de ce que constitue un cluster et de la relation entre les clusters. Pour cette raison, une littérature très riche sur le Clustering sous ses différentes approches, à savoir dure, floue et douce, s'est développée au cours des dernières décennies et plusieurs algorithmes ont été proposés (Chafika 2006) (Rokach et Maimon 2005). Il a été proposé par C.Farley et A.E.Raftery (Fraley et Raftery 1998) de diviser les méthodes de Clustering en deux grandes et principales catégories : les méthodes hiérarchiques et les méthodes de partitionnement. Quant à J.Han et M.Kamber (Han et Kamber 2001), ils ont suggéré de les classer en trois autres catégories majeures : les méthodes basées sur la densité, celles basées sur les motifs et celles basées sur la grille. Pour chacune de ces catégories, il est possible de trouver une multitude de sous-catégories et d'algorithmes différents permettant de déterminer les clusters au sein d'un jeu de données. En adoptant la catégorisation de C.Farley et A.E.Raftery, cette

sous-section décrit les deux classes fondamentales de Clustering, à savoir le Clustering par partitionnement et le Clustering hiérarchique. Elle aborde également les algorithmes sous-jacents de ces deux classes auxquels nous nous intéressons dans cette étude, notamment l'algorithme de Classification ascendante hiérarchique dure et l'algorithme Thresholded fuzzy C-Means.

4.4.1. Clustering par Partitionnement

Le Clustering par partitionnement est la version la plus simple et la plus fondamentale de l'analyse de cluster quireflète une famille de méthodes de regroupement flexibles, basées sur la relocalisation itérative des instances de données entre les clusters. Il vise à découvrir les clusters présents dans le jeu de données en optimisant une fonction objective spécifique et en améliorant de manière itérative la qualité des partitions.

Les méthodes de Clustering par partitionnement créent un partitionnement des instances de données d'un seul niveau en divisant généralement le jeu de données en un certain nombre de clusters disjoints. Ces méthodes requièrent généralement que le nombre de clusters soit prédéfini par l'utilisateur. Les techniques de Clustering par partitionnement construisent dans un premier lieu un partitionnement initial. Ensuite, elles relocalisent itérativement les instances de données en les déplaçant d'un cluster à un autre dans le but d'améliorer la qualité du partitionnement. En fait, la majorité des méthodes de partitionnement est formée pour optimiser un critère de partitionnement objectif, tel qu'une fonction de dis-similarité basée sur la distance, de sorte que les instances de données d'un cluster soient «similaires ou proches» les unes des autres et «dissemblables ou éloignées» des instances des autres clusters (Ayram et Karkkainen 2006) (Han et al. 2012) (Rokach et Maimon 2005).

Etant donné un ensemble de données X de N instances, $X = \{x_1, x_2, \dots, x_N\}$ et K est le nombre désiré de clusters, une approche de Clustering par partitionnement construit typiquement K partitions de données ($K \leq N$), où chaque partition représente un cluster qui doit contenir au moins une instance de données. Les techniques basiques de Clustering par partitionnement organisent les instances d'un ensemble de données en K partitions en adoptant généralement une séparation exclusive des clusters, c'est-à-dire que chaque instance de données doit appartenir exactement à un seul cluster. Cette exigence peut être allégée dans les techniques de partitionnement floues dans lesquelles une instance de donnée peut appartenir à plus d'un groupe (Chafika 2006) (Han et al. 2012).

Il existe de nombreuses méthodes de Clustering par partitionnement proposées dans la littérature, y compris K-Means, K-Medoids, K-Modes, PAM, CLARA, CLARANS, FCM, etc. Le premier algorithme de Clustering partitionnel qui sera abordé dans cette section est l'algorithme de Clustering *K-Means*. Il s'agit de l'un des algorithmes de Clustering les plus simples et les plus efficaces pour le Clustering de données. Ensuite, nous discuterons deux variantes importantes de l'algorithme *K-Means*, à savoir *Fuzzy K-Means* et *Thresholded Fuzzy C-Means*.

- ***K-Moyennes :***

L'algorithme *K-Moyennes* « en anglais *K-Means*» est l'algorithme de Clustering partitionnel le plus populaire. Il a été utilisé avec succès dans les applications scientifiques et industrielles. Il a été initialement mis au point par E.W.Forgy en 1965 (Forgy 1965) et J.McQueen en 1967 (MacQueen 1967), et est également connue sous le nom d'algorithme des centres mobiles. L'algorithme K-Means est une technique itérative permettant de partitionner N instances de données situées dans un espace multidimensionnel en K clusters non uniformes, où K est un paramètre d'entrée qui représente le nombre de clusters fixé à priori. Dans cette méthode, un cluster est représenté par son centre de gravité qui peut être considéré comme une description récapitulative de toutes les instances de données contenues dans le cluster. À cet égard, la forme spécifique de cette description dépendra du type des

données qui sont regroupées. Dans le cas d'attributs numériques, le centroïde correspond à une moyenne de tous les objets situés à l'intérieur du cluster et ses coordonnées sont définies comme la moyenne arithmétique pour chaque dimension séparément de toutes les instances de données appartenant à ce cluster (Hamidouche et Idjeraoui 2013) (MacQueen 1967).

L'algorithme K-Means commence par la sélection arbitraire de *K* centroïdes initiaux dont chacun représente le centre d'un cluster et qui doivent être placés dans des emplacements différents.Ensuite, il attribut chaque instance de données au cluster dont le centroïde est le plus proche en termes de fonction de distance particulière choisie.Cette étape génère un ensemble de *K* clusters, dont chacun regroupe les points de données les plus proches de son centroïde. Dans ce contexte, un large éventail de mesures de proximité peut être utilisé dans l'algorithme K-Means lors du calcul du centroïde le plus proche, dont les plus courantes sont présentées ci-dessus (la sous-section 4.2.). De façon générale, la distance euclidienne est le choix le plus populaire pour l'algorithme K-Means. Une fois les clusters formés, les centroïdes des clusters sont mis à jour en calculant les *K* nouveaux centroïdes à partir de la moyenne des points de données de chaque cluster. Le processus d'attribution des points de données et de réajustement des centres est répété de manière itérative jusqu'à ce que les centroïdes ne changent plus leur position ou jusqu'à ce qu'un autre critère de convergence soit atteint.

Etant donné un ensemble de données$X$ de N éléments, X= $\{x_i \epsilon R^q | \ i \epsilon N\}$, nous désignons le regroupement obtenu après l'application de l'algorithme K-Means par C, C= $\{C_j \subseteq X | \ j \epsilon K\}$. L'algorithme K-Means est essentiellement un problème d'optimisation permettant de trouver une structure de Clustering/les *K* centroïdes qui minimise une fonction objective. La fonction objective la plus connue est la somme des erreurs au carré (en anglais Sum of Squared Error, abrégé SSE) qui est définie dans l'équation suivante (Aggarwal et Reddy 2013) (Amelio et Tagarelli 2019) (Kokate et al. 2018):

$$SSE(C) = \sum_{j=1}^{K} \sum_{i=1}^{N_j} ||x_{ij} - v_j||^2 \quad (8)$$

$$v_j = \frac{\sum_{i=1}^{N_j} x_{ij}}{N_j} \quad (9)$$

Où« K » est le nombre de clusters,

« N_j » est la taille du cluster C_j,

« x_{ij} » est le point de données x_i appartenant au cluster C_j ,

«v_j»correspond au centroïde du cluster C_j,

«|| x_i - v_j ||» est la distance euclidienne entre le $i^{ème}$ élément de données et le $j^{ème}$ centre de cluster.

L'algorithme Fuzzy C-Means procède comme suit:

Etape 1: fixer le paramètre d'entré «*k*» qui est le nombre de clusters;

Etape 2: sélectionner arbitrairement *K* points comme des centroïdes initiaux ; $C_1, C_2, \ldots, C_K$;

Etape 3: former les *K* cluster en assignant chaque point de données au centroïde le plus proche ;

Etape 4: recalculer le centroïde de chaque cluster nouvellement formé ;

Etape 5: répéter les étapes 3 et 4 jusqu'à satisfaction du critère d'arrêt.

❖ ***K-Moyennes Floues***

L'algorithme *K-Moyennes Floues* « en anglais *Fuzzy K-Means*, également connu sous le nom *Fuzzuy C-Means*, abrégé *FCM* », est l'une des techniques de Clustering floue les plus populaires, qui généralise la technique *K-Means* standard. Cette technique a été développée par Dunn en 1973 et améliorée par Bezdek en 1981. A la différence de K-Means standard qui permet de construire une partition nette avec K clusters, l'algorithme K-Means flou permet aux éléments de données d'appartenir à plusieurs clusters simultanément en introduisant la notion de degré d'appartenance, ce qui produit une partition floue de l'ensemble d'éléments de données (Hathaway et al. 1996). Cet algorithme fonctionne en attribuant un degré d'appartenance (u) à chaque élément de données afin de quantifier le taux de rattachement de l'élément à un cluster. Le degré d'appartenance est mesuré sur la base de la distance entre l'élément de données et les centres de clusters. Sa valeur varie de 0 à 1 ($u \epsilon [0,1]$) où plus u est grand et proche de 1, plus l'appartenance de l'élément au cluster est forte, tandis que la valeur 0 signifie qu'il n'est pas membre de ce cluster flou. De toute évidence, la somme des membres de chaque élément de données est égale à 1. On considère X= $\{x_i \epsilon R^q \mid i \epsilon N\}$, l'ensemble de N éléments qui seront associés à des clusters C= $\{C_j \subseteq X \mid j \epsilon K\}$ et u_{ij} le degré d'appartenance d'un élément x_i au $j^{ième}$ cluster. L'algorithme Fuzzy C-Means vise principalement à trouver la bonne partition qui minimise la fonction objective J (MSE, Mean Square Error) suivante:

$$J_{FCM}(U,V) = \sum_{i=1}^{N} \sum_{j=1}^{K} (u_{ij})^m \|x_i - v_j\|^2 \quad , 1 \leq m \leq \infty \tag{10}$$

Où: «*U*» est le fuzzy C-Partition de X ;

«*V*» est une partition floue de l'ensemble de données X formé par les clusters $\{v_1, v_2, \dots, v_k\}$;

«u_{ij}» est le degré d'appartenance de l'élément x_i au cluster j avec $u_{ij} \epsilon [0,1]$, $\sum_{j=1}^{K}(u_{ij}) = 1$;

«v_j» correspond au centre du cluster j.

Le paramètre «*m*», appelé degré de flou,est un poids qui détermine le degré auquel les membres partiels d'un groupe affectent le résultat du Clustering. L'initialisation appropriée de m est essentielle pour la fuzzification. Le paramètre «*m*» prend une valeur réelle supérieure ou égale à 1($m \geq 1$) de sorte que si on l'initialise à 1, le résultat sera une partition dure. De plus, l'augmentation de *m* tend à dégrader le degré d'appartenance vers l'état le plus flou (Miyamoto 1998). D'après (Bezdek et al. 1984), aucune preuve théorique ou de simulation distingue un *m* optimal. Pour la plupart des données, $1,5 \leq m \leq 3,0$ donne de bons résultats.

«$\| x_i - v_j \|$» est la distance euclidienne entre le $i^{ème}$ élément de données et le $j^{ème}$ centre de cluster. Elle est utilisé pour mesurer (exprimer) le degré de similitude.

Le partitionnement flou est réalisé par une optimisation itérative de la fonction objective présentée ci-dessus, avec la mise à jour de l'appartenance u_{ij} et des centres de cluster v_j à chaque itération en utilisant les formules suivantes:

$$v_j = \frac{\sum_{i=1}^{N} (u_{ij})^m * X_i}{\sum_{i=1}^{N} (u_{ij})^m} \tag{11}$$

$$u_{ij} = \left[\sum_{l=1}^{K} \left(\frac{\|X_i - v_j\|}{\|X_i - v_l\|} \right)^{\frac{2}{m-1}} \right]^{-1} \tag{12}$$

L'algorithme Fuzzy C-Means procède comme suit:

Etape 1: fixer les paramètres suivants:

- *«k»*, le nombre de clusters;
- *«ε»*, le seuil représentant l'erreur de convergence (par exemple : ε= 0.001) ;
- *«m»*, le degré de flou, généralement pris égal à 2 (m=2).

Etape 2: initialiser la matrice degrés d'appartenances U^0 par des valeurs aléatoires u_{ij} dans l'intervalle [0, 1] ($u_{ij}\ \epsilon[0,1]$)de sorte que $\sum_{j=1}^{k} u_{ij} = 1$.

Etape 3: dans chaque itération "r" de l'algorithme, calculer les centres des clusters avec U^r en utilisant l'équation (11).

Etape 4:mettre à jour la matrice degrés d'appartenances U^r, U^{r+1} par la relation (12).

Etape 5: arrêter si $\|U^{r+1} - U^r\| \leq \varepsilon$, sinon répéter les étapes 3 à 4 jusqu'à satisfaction du critère d'arrêt.

❖ ***C-Moyennes Floues à Seuil***

Une généralisation de l'algorithme Fuzzy C-Means standard a été proposée par M.S. Kamel et S.Z. Selim en 1991dans le but de réduire le degré de flou de la partition finale. Cette approche est connue dans la littérature sous le nom *C-Moyennes Floues à Seuil* « en anglais *Thresholded Fuzzy C-Means*, abrégé *TFCM*». Elle permet principalement d'utiliser les itérations de base de l'algorithme de Clustering FCM et d'appliquer un seuillage sur la partition finale en fonction des degrés d'appartenance pour limiter le flou des résultats. Elle permet également de générer des clusters par un processus itératif en minimisant une fonction objective et d'obtenir une partition semi-floue de l'ensemble de données dans laquelle chaque élément de données peut appartenir à plus d'un cluster mais pas nécessairement à tous, avec des valeurs d'appartenance variantes. Autrement dit, lorsqu'un degré d'appartenance u_{ij} du vecteur v_i à la classe j est inférieur à une valeur seuil « **τ** », il est forcé à zéro et les autres degrés d'appartenance sont ensuite normalisés de telle sorte que leur somme soit égale à un. Dans la pratique, cette approche est mise en œuvre en ajoutant les étapes suivantes à la fin de l'algorithme de Bezdek (Kamel et Selim 1991) :

- **calcul du seuil (global):** L'algorithme TFCM permet de calculer la valeur de ℧ de façon à ce que l'équation (1) soit satisfaite, en procédant par plusieurs méthodes (Kamel et Selim 1991) telles que la détermination du maximum de seuil **τ,** la combinaison de plusieurs restrictions semi-floues ou encore l'utilisation de seuils multiples.
- **seuillage :**

$$\text{If } u_{ij} < \tau, \text{then set } u_{ij} = 0, \ \ such\ as\ 1 \leq i \leq n\ and\ 1 \leq j \leq c \qquad (13)$$

- **normalisation, calcul de nouveaux degrés d'appartenance u_{ik}^*:**

$$\text{For each i, } 1 \leq i \leq n \ \text{ set } \quad u_{ij}{}^* = \frac{u_{ij}}{\sum_{j=1}^{k} u_{ij}} \text{ such as } 1 \leq j \leq k \qquad (14)$$

Ces étapes supplémentaires peuvent être vues comme une tâche de pré-défuzzification qui peut être jugée importante lorsque le résultat souhaité est une classification semi floue. Cette tâche est procédée d'une façon naturelle après la convergence de l'algorithme FCM en considérant pour chaque élément de données que les classes finales pour lesquelles les degrés d'appartenance sont suffisants par rapport à un seuil. L'algorithme Thresholded Fuzzy C-Means procède comme suit:

Début

Etape 1: fixer les paramètres suivants:

- «*k*», le nombre de clusters;
- «*ε*», le seuil représentant l'erreur de convergence (par exemple : ε= 0.001) ;
- «*m*», le degré de flou, généralement pris égal à 2 (m=2).

Etape 2: initialiser la matrice degrés d'appartenances U^0 par des valeurs aléatoires u_{ij} dans l'intervalle [0, 1] ($u_{ij}\,\epsilon[0,1]$)de sorte que $\sum_{j=1}^{k} u_{ij} = 1$.

Etape 3: dans chaque itération "r" de l'algorithme, calculer les centres des clusters avec U^r en utilisant l'équation (11).

Etape 4: mettre à jour la matrice degrés d'appartenances U^r, U^{r+1} par la relation (12).

Etape 5: arrêter si $\|U^{r+1} - U^r\| \leq \varepsilon$, sinon répéter les étapes 3 à 4 jusqu'à satisfaction du critère d'arrêt.

Etape 6:

- Calculer un seuil global τ en utilisant l'une des méthodes de seuillage citées précédemment.
- Seuillage des valeurs des degrés d'appartenance en utilisant la relation (13).
- Normaliser les valeurs de degré d'appartenance seuillées en utilisant la formule (14).

Fin

4.4.2. Clustering Hiérarchique

Le Clustering hiérarchique (en anglais *hierarchical Clustering*) est l'une des approches les plus importantes du Data Mining et des statistiques, ayant pour objectif d'identifier des groupes d'objets similaires dans un jeu de données. Il fait référence à une famille de méthodes d'analyse de cluster qui cherche à construire une hiérarchie de clusters ou un arbre de clusters, c'est-à-dire une structure de type arborescente basée sur la hiérarchie, connue sous le nom de dendrogramme. Ce dernier représente principalement les relations entre les données et décrit l'ordre dans lequel les clusters sont construits (Amelio et Tagarelli 2019) (Chafika 2006). Il permet de choisir automatiquement le bon nombre de clusters en divisant l'arbre à différents niveaux afin d'obtenir différentes solutions de Clustering pour le même jeu de données sans avoir à ré-exécuter l'algorithme de Clustering (Aggarwal et Reddy 2013). Fondamentalement, les méthodes du Clustering hiérarchique peuvent être subdivisées en fonction de la manière dont la décomposition hiérarchique est formée, en deux types, à savoir *agglomératives* et *divisives*.

- ***Clustering hiérarchique agglomératif,*** également connu sous le nom d'«*approche ascendante*» ou *de « Bottom-up approach »*en anglais, représente initialement chaque instance de données par un cluster singleton spécifique. Ensuite, il agglomère (vue de bas en haut) successivement des paires de clustersen fonction de leur proximité, jusqu'à ce que tous les clusters soient regroupés en un seul cluster (le niveau le plus élevé de la hiérarchie) contenant tous les points de données ou qu'une condition d'arrêt soit remplie (par exemple, la construction de *K* clusters). Dans cette approche ascendante, le processus de Clustering se progresse à chaque étape vers le haut et génère une hiérarchie de clusters représentée par un dendrogramme à partir duquel le Clustering final est obtenu en coupant le dendrogramme à un niveau spécifique (Amelio et Tagarelli 2019) (Kokate et al. 2018).
- ***Clustering hiérarchique divisif,*** également connu sous le nom d'*«approche descendante»* ou *«Top-down appoach»* en anglais, rassemble initialement toutes les instances de données dans un cluster unique. Ensuite, au fur et à mesure de la progression de l'algorithme, les clusters sont successivement divisés (vue de haut en bas) en fonction d'une mesure de similarité entre leurs

instances de données. Cette division se fait en deux nouveaux clusters choisis pour maximiser la distance entre les clusters. Le processus de Clustering via l'approche descendante se poursuit jusqu'à ce que chaque instance de données devienne un seul cluster ou qu'une condition d'arrêt donnée par l'utilisateur soit atteinte (par exemple, la structure de cluster souhaitée soit obtenue) (Kokate et al. 2018) (Rokach et Maimon 2005).

Les techniques hiérarchique divisives sont moins communes. On se concentre dans notre étude sur les techniques hiérarchiques agglomératives.

❖ ***Technique de Classification Ascendante Hiérarchique***

L'algorithme de Classification Ascendante Hiérarchique « en anglais *Hierarchical Ascending Classification,* abrégé *HAC* » appartient à la famille des algorithmes de Clustering hiérarchiques agglomératifs, permettant de construire une hiérarchie entière des objets sous la forme d'un "arbre" dans un ordre ascendant. L'algorithme HAC vise à regrouper, en fonction d'un critère de similarité, un ensemble de *N* objets décrits par des caractéristiques (des variables) et représentés sous forme des vecteurs,de sorte à ce que les objets regroupés au sein d'un même cluster (homogénéité intra-clusters) soient les plus semblables possible,et que les clusters soient les plus dissemblables possibles (hétérogénéité inter-clusters). Il considère initialement chaque objet de données comme étant un cluster individuel, et à chaque étape, fusionne la paire des clusters les plus proches selon une mesure de similarité pour former une nouvelle classe. Le processus est itéré jusqu'à ce que tous les objets se trouvent dans un seul cluster. Cette classification produit un arbre binaire de classification appelé dendrogramme. Ce dernier permet de visualiser et décrire l'ensemble de partitions imbriquées et cohérentes, qui sont autant de solutions potentielles. Le dendrogramme peut être coupé à différents niveaux pour obtenir des clusters significatifs. Le choix de niveau de coupure dépend généralement soit aux contraintes des utilisateurs où ils sont en mesure de fixer un nombre de clusters souhaités k, soit aux critères plus objectifs. Le principe de l'algorithme de classification hiérarchique ascendante standard est simple:

Soit$X$ un ensemble de N instances de données à regrouper, $X = \{x_1, x_2, \dots, x_n\}$.

- Considérer chaque instance de données comme étant un cluster unique. L'ensemble de ces clusters forme les N clusters initiaux. Ensuite, former la matrice de distance qui peut être obtenue en calculant la distance entre chaque paire de clusters ;
- Fusionner les clusters les plus proches en termes de leurs distances et formez un nouveau cluster ;
- Mettre à jour la matrice de distances en remplaçant les deux clusters regroupés par le nouveau et en calculant sa distance avec chacun des autres clusters ;
- Itérer les deux étapes précédentes jusqu'à l'obtentiond'un seul cluster ou l'atteinte du nombre de clusters k préfixé par l'utilisateur.

L'opération clé du Clustering hiérarchique agglomératif est de définir une mesure de similarité appelée «critère d'agrégation», utilisée pour former une K partition optimale de l'ensemble de données. Selon un critère d'agrégation particulier, HAC mesure la ressemblance entre les individus par le biais de la notion de « *distance* » (voir la section 4.2.) entre les instances de données afin de former des clusters. De nombreux critères d'agrégation ont été proposés dans la littérature. Les plus connus sont: *critère du lien simple, critère du lien complet, critère du lien moyen, critère de Ward* (Abdellaoui 2014) (Han et al. 2012) (Tabet et Ziani 2013). Ce dernier constitue la méthode d'agrégation la plus courante dans la classification hiérarchique à laquelle on s'intéresse dans cette thèse.

Critère du Lien Simple :

La méthode du lien simple, également appelée *critère du saut minimum* ou *Single-linkage criteria* en anglais, est l'une des méthodes les plus utilisées. Elle considère que la distance entre deux clusters est égale à la distance la plus courte «minimale» entre toute instance de données d'un cluster et toute instance de données de l'autre cluster. En utilisant le critère du lien simple, l'algorithme de classification hiérarchique ascendante combine à chaque étapeles deux clusters ayant la plus petite distance du lien simple. La distance minimale entre les instances de données appartenant aux clusters $C_j$et $C_l$est calculée avec la formule suivante (Chafika 2006) :

$$d(C_j, C_l) = \min_{x_r \in C_j\,, x_s \in C_l} d(x_r, x_s) \tag{15}$$

Critère du Lien Complet :

La méthode du lien complet, également appelée *méthode du diamètre*, *critère du saut maximum* ou *Complete-Linkage criteria* en anglais, considère que la distanceentre deux clusters est égale à la distance la plus longue (distance maximale) entre toute instance de données d'un cluster et toute instance de données de l'autre cluster (Rokach et Maimon 2005). En utilisant le critère du lien complet, l'algorithme de classification hiérarchique ascendante fusionne à chaque étape les deux clusters ayant la plus petite distance du lien complet. La distance maximale entre les instances de données appartenant aux clusters $C_j$et $C_l$est calculée avec la formule suivante (Chafika 2006):

$$d(C_j, C_l) = \max_{x_r \in C_j\,, x_s \in C_l} d(x_r, x_s) \tag{16}$$

Critère du Lien Moyen

La méthode du lien moyen, également appelée *méthode de variance minimale* ou *Average-Linkage criteria* en anglais, considère que la distance entre deux clusters est égale à la distance moyenne entre toutes les instances de données d'un cluster et toutes les instances de données de l'autre cluster (Rokach et Maimon 2005). En utilisant le critère du lien moyen, l'algorithme de classification hiérarchique ascendante fusionne à chaque étape les deux clusters ayant la plus petite distance du lien moyen. La distance moyenne entre les instances de données appartenant aux clusters $C_j$et C_l, possédant respectivement la taille N_{C_j}et N_{C_l}, est obtenue par la formule suivante (Chafika 2006):

$$d(C_J, C_L) = \frac{1}{N_{C_j} \times N_{C_l}} \times \sum_{r=1}^{N_{C_j}} \sum_{s=1}^{N_{C_l}} d(x_r\,, x_s) \quad x_r \in C_j\,, x_s \in C_l \tag{17}$$

Le critère de Ward

La méthode de Ward, également appelée *distance de Ward* ou *Ward'scriteria* en anglais, consiste à utiliser un algorithme qui permet à chaque étape d'agréger deux clusters de sorte que l'augmentation de l'inertie intra-classe soit minimale ou l'augmentation de l'inertie interclasse soit maximale. Ces deux derniers critères sont équivalents d'après le théorème de Hygens ($Inertie\ totale = Inertie\ inter - classe + Inertie\ intra - classe$). Dans le critère de Ward, la distance entre deux clusters $C_j$et C_l est celle de leurs barycentres au carré, pondéré par les effectifs des deux clusters. En se basant sur la distance euclidienne, le critère de Ward est donné par la formule suivante:

$$D(C_j, C_l) = \frac{e_{C_j} * e_{C_l}}{e_{C_j} + e_{C_l}} * \left\| g_{C_j} - g_{C_l} \right\|^2 \quad (18)$$

Où g_{C_j} est le centre de gravité du cluster C_j ;

g_{C_l} est le centre de gravité du cluster C_l ;

e_{C_j} est le nombre d'éléments de données dans le cluster C_j ;

e_{C_l} est le nombre d'éléments de données dans le cluster C_l ;

$\left\| g_{C_j} - g_{C_l} \right\|^2$ est la distance euclidienne au carré entre les centres de gravité des deux clusters C_j et C_l .

Nous considérons alors le critère d'agrégation de Ward comme étant une mesure de la perte d'inertie inter-clusters lors du regroupement des deux clusters j et l. Par conséquent, l'algorithme HAC permet à chaque itération de regrouper la paire de clusters qui provoque une perte minimale de l'inertie inter-clusters (Abdellaoui 2014) (Tabet et Ziani 2013). Il s'agit donc d'une optimisation pas-à-pas, qui ne dépend d'aucune initialisation arbitraire. L'algorithme de classification ascendante avec le critère de Ward permet de générer des clusters plus petits dans un ordre instructif pour l'affichage des données. En effet, à chaque étape de la mise en cluster, ce critère induit une minimisation de la perte de variance interclasse. En outre, l'interprétation du saut de Ward est plus propre que celle obtenue par les autres méthodes d'agrégation.

❖ ***Discussion***

Etant donné que l'algorithme *TFCM* fait partie de la famille des algorithmes de partitionnement *K-Means* et bâtit principalement sur *FCM*, il leur permet d'hériter les mêmes propriétés que ces deux algorithmes. En fait, le *TFCM* est un algorithme fiable et efficace. Il se caractérise par sa facilité d'interprétation, sa simplicité de mise en œuvre, son adaptabilité aux données éparses et sa vitesse de convergence. Il garantit particulièrement une diminution de la fonction objective d'une itération à l'autre.Cependant, le *TFCM* hérite également les faiblesses des algorithmes *K-Means* et *FCM*, telles que : (i) la sensibilité à l'initialisation, (ii) le long temps de calcul, (iii) la sensibilité au bruit et aux valeurs aberrantes pour les clusters de données de tailles et de volumes différents. La principale lacune de l'algorithme *TFCM* réside dans sa phase d'initialisation où il nécessite que le nombre de clusters, la matrice de partition floue et les centroïdes des clusters soient initialisés à l'avance et d'une manière appropriée en raison de leurs grandes influences sur l'évolution de l'algorithme et le résultat attendu. La sélection aléatoire des paramètres d'entrées fait que le processus itératif tombe facilement dans des optima locaux. En outre, il pourrait converger vers le même résultat, mais probablement avec un nombre différent d'itérations, ce qui peut contribuer à des performances médiocres en ralentissant la vitesse et en augmentant le temps de calcul. À cet égard, les algorithmes de Clustering hiérarchique ont été développés principalement pour surmonter certains des inconvénients associés aux méthodes de Clustering partitionnelles. Le Clustering hiérarchique ascendant est l'une des techniques les plus recommandées du Clustering hiérarchique en raison de son utilité avantageuse. Il surpasse plusieurs autres approches du fait qu'il ne nécessite pas la connaissance préalable du nombre de clusters et qu'il ne dispose pas de fonction d'initialisation. En raison de la nature heuristique de la *HAC*, le problème rencontré généralement réside dans le choix d'une métrique de similarité pour qu'elle soit la plus proche de la métrique utilisée par les individus. En outre, les décisions de fusion ou de division prises à un niveau donné de la hiérarchie ne peuvent pas être annulées, ce quiconduit à la réduction de la flexibilité. En plus, la *HAC* permet de générer des partitions finales dures, ce qui limite son application correcte sur certains ensembles de données de problèmes réels. Dans certains cas, les concepteurs des

applications s'efforcent à maintenir un environnement réaliste pour traiter leurs problèmes, ce qui rend indésirable pour euxla nature des résultats durs de ces algorithmes (Aggarwal et Reddy 2013) (Grover 2014) (Kamel et Selim 1991).

Afin de surmonter les inconvénients des deux algorithmes *TFCM* et *HAC* présentés précédemment et de bénéficier de leurs avantages, on peut profiter de la technique qui permet d'améliorer la partition obtenue d'un type de Clustering spécifié avec une méthode de deuxième type de Clustering (Zhang et al. 1996). Pour ce faire, nous proposons d'améliorer la qualité du Clustering par une hybridation successive de la technique hiérarchique dure *HAC* et l'algorithme de partitionnement flou *TFCM*. L'hybridation successive des deux algorithmes permet de générer des clusters par un processus itératif en minimisant une fonction objective, et d'obtenir une partition floue de l'ensemble de données dans laquelle chaque élément de donnée peut appartenir à plus d'un cluster mais pas nécessairement à tous, avec des valeurs d'appartenance variantes.

4.5. Techniques de Validation du Clustering

La validation du Clustering fait référence à l'évaluation quantitative des résultats issus du processus de Clustering, qui est une partie cruciale du choix de l'algorithme de Clustering et du regroupement le plus performant pour un ensemble de données d'entrée. Il s'agit d'un processus autonome qui ne fait pas partie de la tâche de Clustering mais qui est plutôt effectué après la génération du résultat final du Clustering. La validation du Clustering se déroule principalement selon deux techniques, à savoir la validation externe et la validation interne. Chacune de ces deux techniques possède des indices représentatifs appelés indices de validité. Dans ce qui suit, nous présentons une description de chaqueméthode de validation des clusters ainsi que leurs indices représentatifs (Hamidouche et Idjeraoui 2013) (Halkidi et al. 2002) (Hassani et Seidl 2017).

4.5.1. Validation Externe

La validation externefait référence à une méthode pouvant servir à tester et valider différentes techniques de Clustering en utilisant des critères qui ne sont pas inhérents à l'ensemble de données. Elle mesure la qualité des résultats du Clustering par rapport à une information externe ou un résultat de référence qui est considéré comme étant la base réelle connue à priori. Il peut s'agir de connaissances préalables ou spécifiées par un expert sur les clusters, telles qu'une structure de cluster connue de l'ensemble de données ou obtenue par un autre processus de Clustering, ou des connaissances préalables sur les données telles que les véritables étiquettes de cluster pour chaque instance du jeu de données (Zaki et Meira 2020).

Soit $D = \{x_i\}_{i=1}^{N}$ un ensemble de données constitué de N points dans un espace à Q dimensions, partitionné réellement en un ensemble de J clusters $R=\{R_1, R_2,..., R_J\}$. On dénote $C=\{C_1, C_2,..., C_K\}$, le regroupement de l'ensemble de données D en K clusters obtenu via un processus de Clustering. Les indices de validité externe permettent de mesurer d'une certaine manière la similitude entre les deux groupements R et C. Nous présentons dans cette section les indices les plus utilisés (Dudoit et al., 2002).

- ***Matrice de Confusion***

La matrice de confusion, également connue sous le nom de *tableau de contingence*, est un tableau croisé $N \times N$qui résume toutes les informations pertinentes pour évaluer les performances et les clusters résultantd'un algorithme ou d'un modèle de regroupement (Chafika 2006) (Palacio-Niño et Berzal 2019) (Zaki et Meira 2020). La construction et l'assurance du fonctionnement d'une matrice de confusion nécessite de déterminer et de comprendre les valeurs de quatre terminologies principales:

- ✓ **True Positive (TP):** Nombre de paires de données trouvées dans le même cluster, dans C et dans R à la fois.
- ✓ **TrueNegative (TN) :** Nombre de paires de données trouvées dans de différents clusters, dans C et dans R à la fois.
- ✓ **False Positive (FP) :** Nombre de paires de données trouvées dans le même cluster dans C mais dans des clusters différents dans R.
- ✓ **False Negative (FN) :** Nombre de paires de données trouvées dans de différents clusters dans C mais dans le même cluster dans R.

- ***La précision***

La précision (en anglais *Precision*) fait référence à la fraction d'instances correctement classées dans le même cluster (vraie positive), divisée par le nombre total d'instances classées positives. Elle indique le degré de confiance accordé au modèle de Clustering lorsqu'il classe une instance comme étant positive.Elle peut être calculée avec la formule suivante (Grandini et al. 2020) (Palacio-Niño et Berzal 2019):

$$Precision = \frac{TP}{TP + FP} \tag{19}$$

- ***Le rappel***

Le rappel (en anglais, *Recall*) fait référence à la fraction des instances vraie positive (TP) divisée par le nombre total d'instances classées réellement positives dans l'ensemble de données. Celui-ci équivaut à la somme des vrais positifs (TP) et des faux négatifs (FN). Egalement appelé le « taux de vrais positifs » et parfois la « sensibilité », le rappel est calculé par la formule suivante (Grandini et al. 2020):

$$Recall = \frac{TP}{TP + FN} \tag{20}$$

- ***Le score F1***

Le score F1 (en anglais, F1-score ou F-measure) combine les mesures de la précision et durappel de la recherched'information sous le concept de moyenne harmonique [CHA 06]. Il est utilisé afin d'évaluer et de comparer la qualité et les performances de Clustering en tenant compte des clusterscorrectes connus pour un jeu de données.Le score F1 correspond à une moyenne pondérée de la précision et du rappel. Sa valeur est maximale lorsque le rappel et la précision sont équivalents. Les valeurs du score F1 sont comprises dans l'intervalle [0,1] et les valeurs les plus grandes indiquent une meilleure qualité de Clustering (Zaki et Meira 2020). La F-mesure est calculée par la formule suivante:

$$F - measure = \frac{2}{(Recall^{-1} + Precision^{-1})} = 2 \times \left(\frac{Recall \times Precision}{Recall + Precision}\right) \tag{21}$$

- ***Pureté***

La pureté (en anglais, *purity*) quantifie la mesure dans laquelle un cluster contient des entités provenant d'une seule partition. En d'autres termes, elle mesure le rapport entre les entités qui se trouvent dans le cluster ayant la même classe réelle. La pureté de chaque cluster$C_k \in C$est définie comme étant le pourcentage du type de données prédominant selon la classe réelle connue$\mathrm{R_j} \in R$, et est calculée comme suit (Chafika 2006) (Rendón et al. 2011) (Zaki et Meira 2020).

$$\text{Purity}\ (C_k) = \underset{\mathrm{R_j} \in R}{max} \frac{N_{jk}}{N_k} \tag{22}$$

Où $N_k$est la taille du cluster$C_k$etN_{jk}est le nombre des points de données de la classe réelle$\mathrm{R_j}$dans ce clusterC_k. La pureté globale du Clustering C «Purity (C)»est alors définie comme étant une somme pondérée des puretés des clusters individuels.Elle est donnée comme suit:

$$\text{Purity}\ (C_k) = \frac{1}{N}\sum_{k=1}^{K} \underset{\mathrm{R_j} \in R}{max} \quad N_{jk} \tag{23}$$

- ***Coefficient de Jaccard***

Le coefficient de Jaccard (en anglais, Jaccard Coefficient) mesure la fraction des instances vraies positives (TP), après avoir ignoré les vraies négatives (TN) (Zaki et Meira 2020). Il évalue la similarité entre les clusters détectés *C* par rapport à la partition *R* fournie (Palacio-Niño et Berzal 2019). Les valeurs du coefficient de Jaccard sont comprises dans l'intervalle [0,1] et pour un Clustering C parfait, le coefficient de Jaccard a la valeur 1. Le coefficient de Jaccard est défini comme suit :

$$Jaccard = \frac{TP}{TP + FN + FP} \tag{24}$$

- ***Indice de Rand***

L'indice de Rand (en anglais *Rand Index,* abrégé *RI*) mesure la fraction de vrais positifs (*TP*) et de vrais négatifs (*TN*) sur le nombre total de données *N*.Il est équivalent à l'exactitude (en anglais, *Accuracy*) dans un contexte d'apprentissage automatique supervisé (Palacio-Niño et Berzal 2019). Les valeurs de l'indice de Rand sont comprises entre 0 et 1 dont les valeurs élevées sont souhaitables pour un meilleur Clustering (Zaki et Meira 2020). L'indice de Rand est défini comme suit :

$$Rand = \frac{TP + TN}{TP + TN + FP + FN} = \frac{TP + TN}{N} \tag{25}$$

- ***Entropie***

L'entropie (en anglais Entropy, abrégé E) est une mesure de qualité qui reflète la façon dont les membres des *J* classes réelles*R* sont distribués dans chaque cluster $C_k \in C$. L'entropie globale pour un ensemble de clusters *C* est calculée comme la somme pondérée des entropies de tous les clusters. C'est une mesure plus complète que la pureté, car elle tient compte de la distribution de toutes les classes dans chaque cluster (Chafika 2006) (Rendón et al. 2011). L'entropie globale est calculée parl'équation suivante :

$$E\ (C) = -\sum_{k=1}^{K} \frac{N_k}{N} \sum_{R_j \in R} \frac{N_{kj}}{N_k} \log(\frac{N_{kj}}{N_k}) \tag{26}$$

Où N_k est la taille du cluster *k*, N_{kj}est le nombre d'instances de la classe $R_j$dans le cluster C_k,et *N* est le nombre total des points de données.

4.5.2. Validation Interne

La validation interne se réfère à une méthode d'évaluation des résultats d'un processus de Clustering en utilisant uniquement les quantités et les caractéristiques inhérentes à l'ensemble de données (Halkidi et al. 2002). Elle mesure la qualité du Clustering en s'appuyant sur la structure des clusters

trouvés et les relations entre eux, sans avoir recours à des informations externes (Tseng et al., 2005). Cette méthode est beaucoup plus efficace et réaliste dans plusieurs scénarios du monde réel, présentant une production croissante et continue de volumes de données. C'est le cas dans les applications récentes de Cloud Computing où il est difficile de prétendre à la validité ou à la disponibilité d'une connaissance complète des bases réelles. Différentes mesures de validité interne ont été proposées dans la littérature. Elles sont basées principalement sur les concepts de la compacité et/ou la séparation afin de mesurer la relation de proximité étroite entre les éléments (objets) d'un cluster et révéler le degré d'isolement ou de distinction d'un cluster des autres respectivement (Bensaid et al. 1996) (Hassani et Seidl 2017) (Rendón et al. 2011). Ces deux concepts fondamentaux reflètent la bonnetée de la qualité du Clustering qui devrait ressembler à une grande compacité et une séparation importante (Cui et al. 2017). Dans cette section, nous présentons les mesures de validité interne les plus courantes.

Soit $D = \{x_i\}_{i=1}^{N}$ un ensemble de données constitué de *N* points dans un espace à *Q* dimensions. On dénote $C=\{C_1, C_2,..., C_K\}$, le regroupement de l'ensemble de données *D* en *K* clusters obtenu via un processus de Clustering.

- **Coefficient de Partition:**

Le coefficient de partition (en anglais *Partition Coefficient*, abrégé *PC*) est considéré comme étant le premier indice de validité proposé par Bezdek dans (Wang et Zhang 2007) pour mesurer la quantité de «chevauchement» entre les clusters. L'indice *PC* se base principalement sur la minimisation de l'intersection floue par paire dans la matrice de partition U. Il est défini comme suit:

$$PC(_c)= \frac{1}{N}\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^2 \quad (27)$$

Où N est le nombre de clusters, u_{ij} est l'appartenance du point de données j dans le cluster i. Les valeurs d'index *PC*varient dans la plage de [1/c, 1] et plus l'indice est proche de l'unité, plus la qualité du Clustering est meilleure. Par ailleurs, une valeur proche de 1/c indique qu'il n'y a pas de tendance de regroupement dans l'ensemble de données ou que l'algorithme de Clustering n'a pas réussi à le révéler.

- **L'entropie de partition**

L'entropie de partition (en anglais *Partition Entropy*, abrégé *PE*) est un indice basé sur l'entropie simple proposée par (Wang et Zhang 2007). Elle mesure le flou de la partition de cluster uniquement. Elle est définie comme suit:

$$PE= \frac{1}{N}\left(\sum_{i=1}^{c}\sum_{j=1}^{N} u_{ij} log_b(u_{ij})\right) \quad (28)$$

Où *b* représente une base de logarithme. L'indice *PE* est calculé pour les valeurs de N supérieures à 1 et ses valeurs sont obtenues dans la plage $[0, \log_b(c)]$. Contrairement à l'indice *PC*, des valeurs *PE* plus petites montrent la présence de clusters bien séparés tandis que les structures de cluster deviennent floues si les valeurs *PE* approchent de la limite supérieure de la plage.

- **L'indice de Partition:**

L'indice de partition (en anglais, *Partition Index* ou *SC*) est le rapport de la somme de la compacité et de la séparation des clusters. Il s'agit d'une somme de mesures de validité de cluster individuelles, normalisées par division par la cardinalité floue de chaque cluster (Kaya 2005).

$$SC(c) = \sum_{i=1}^{c} \frac{\sum_{j=1}^{N}(u_{ij})^m \left\|x_j - v_i\right\|^2}{N_i \sum_{k=1}^{c}\|v_k - v_i\|^2} \quad (29)$$

Où N_i est le nombre d'échantillons dans le $i^{ème}$ cluster. SC est utile pour comparer différentes partitions ayant un nombre égal de clusters. Une valeur inférieure de *SC* indique une meilleure partition.

- **Indice de séparation:**

A l'inverse de l'indice de partition (*SC*), l'indice de séparation utilise une séparation à distance minimale pour la validité de la partition (Kaya 2005).

$$S(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^2\left\|x_j - v_i\right\|^2}{N min_{i,k}\|v_k - v_i\|^2} \quad (30)$$

- **L'indice de Xie et Beni:**

L'indice de Xie et Beni (en anglais Xie and Beni's Index, abrégé *XB*) est l'un des indices de validité de Clustering floue les plus utilisé. Il a été proposé par Xie et Beni en 1991 avec le paramètre m fixé à 2. Il vise à quantifier le rapport de la variation totale au sein des clusters et la séparation des clusters. Cet indice a été développé en tant que fonction de rapport de la validité de regroupement, mesurant la compacité du regroupement flou à travers la distance des objets (échantillons) dans un cluster de leurs centres de cluster et la séparation des clusters représentée par la distance entre les centres des clusters (Wang et Zhang 2007). En considérant une partition floue de l'ensemble de données X = {x_j; j = 1, ..., n} avec v_i (i = 1,…, n_c}, les centres de chaque cluster et u_{ij} l'appartenance du point de données j par rapport au cluster i, l'indice *XB* peut être calculé comme suit:

$$XB(c) = \frac{\sum_{i=1}^{c}\sum_{j=1}^{N}(u_{ij})^m\left\|x_j - v_i\right\|^2}{N min_{i,j}\left\|x_j - v_i\right\|^2} \quad (31)$$

Des valeurs plus petites de l'indice *XB* indiquent une meilleure qualité de Clustering, cela veut dire des grappes plus compactes et bien séparées. Cependant, l'indice *XB* diminue de façon monotone à mesure que c s'approche de n (nombre clusters n_c devient très grand et proche de n).

- **L'indice de Dunn:**

L'indice de Dunn (en anglais, Dunn Index ou *DI*) a été introduit par J. C. Dunn en 1974 comme une métrique pour évaluer les algorithmes de Clustering. Il définit le rapport entre la distance minimale intra-cluster et la distance maximale inter-cluster afin d'identifier les ensembles de clusters qui sont compacts, avec une petite variance entre les objets du cluster, et bien séparés, où les moyennes des différents clusters sont suffisamment éloignées, par rapport au cluster à l'intérieure variance (Wang et Zhang 2007). L'indice DI est donné par l'équation suivante:

$$DI_c = \min_{i\in c}\left\{\min_{j\in c, i\neq j}\left\{\frac{d(C_i, C_j)}{\max_{k\in C} diam\,(C_k)}\right\}\right\} = \min_{i\in c}\left\{\min_{j\in c, i\neq j}\left\{\frac{\min_{x\in C_i, y\in C_j} d(x,y)}{\max_{k\in C}\{\max_{x,y\in C} d(x,y)\}}\right\}\right\} \quad (32)$$

Où $\boldsymbol{d(C_i, C_j)}$ est la fonction de dissimilarité entre deux clusters C_i et C_j, définie par $\boldsymbol{d(C_i, C_j) = min_{x\in C_i, y\in C_j} d(x,y)}$ qui dénote la distance minimale qui sépare deux éléments xi et yj (objets de différents clusters) classés séparément, et $\boldsymbol{diam\,(C_k)}$ le diamètre d'un cluster, qui peut être considéré comme une mesure de la dispersion des clusters et défini comme $\boldsymbol{diam(C_k) = max_{x,y\in C_k} d(x,y)}$ où $\boldsymbol{max_{x,y\in C_k} d(x,y)}$ dénote la distance maximale qui sépare deux éléments classés ensemble (objets du même cluster). L'indice de Dunn est limité à l'intervalle [0, 1] et doit être maximisé pour indiquer un meilleur Clustering (la présence de grappes compactes et bien séparées).

- **L'indice de Davies-Bouldin**

L'indice de Davies-Bouldin (en anglais, Davies-Bouldin Index ou *DBI*) est une mesure de similarité entre les clusters qui est définie à partir d'une mesure de dispersion d'un cluster et d'une mesure de dissimilarité entre deux clusters. Cet indice traite chaque cluster individuellement et cherche à mesurer la similitude moyenne entre chaque cluster C_i, i = 1,…, N_c et le cluster qui lui est le plus proche (similaire) (Davies et Bouldin 1979). C'est la moyenne du rapport maximal entre la distance d'un point au centre de son cluster et la distance entre deux centres de clusters. Il est formulé de la façon suivante:

$$DBI = \frac{1}{N}\sum_{i=1,i\neq j}^{n} \max\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \quad (33)$$

Où N est le nombre de clusters, σ_i est la distance moyenne de tous les objets du cluster i à leur centre de cluster c_i, σ_j est la distance moyenne de tous les objets du cluster j à leur centre de cluster c_j, et d (c_i, c_j) est la distance des centres de cluster c_i et c_j. Les petites valeurs de DBI correspondent à des clusters compacts et dont les centres sont éloignés les uns des autres. La meilleure partition est donc celle qui minimise la moyenne de la valeur calculée pour chaque classe. En d'autres termes, la meilleure partition est celle qui minimise la similarité entre les classes.

5. Clustering pour la Disponibilité dans le Cloud Computing

Étant donné que la disponibilité permanente, l'accessibilité et la gestion efficace des services dans les environnements de Cloud Computing sont d'une importance vitale autant pour les clients que pour les fournisseurs de services, l'intégration des techniques de Clustering dans les environnements distribués orientés services, notamment le modèle SaaS, est devenue nécessaire afin de maintenir une performance raisonnable (Shahapure et Jayarekha 2015) (Shahriar et al. 2011). Dans cette section, nous présentons les travaux de recherche importants appliquées pour la disponibilité dans le Cloud Computing, particulièrement dans le modèle SaaS.

G.Malathy et al. ont proposé dans (Malathy at al. 2013) une approche de Clustering de ressources permettant de former des clusters en s'appuyant sur l'identification de la distribution de l'utilisation des ressources pour un groupe de nœuds présentant des modèles d'utilisation des ressources similaires. Cette approche vise à améliorer les performances de l'environnement Cloud en termes d'utilisation des ressources. Elle utilise deux techniques complémentaires, à savoir: (i) les histogrammes d'utilisation des ressources, utilisés pour fournir des informations statistiques sur les capacités des ressources, et (ii) l'algorithme Espérance-Maximisation (en anglais Expectation-Maximization algorithm, souvent abrégé EM), utilisé pour regrouper les ressources et obtenir une représentation compacte d'un ensemble de cloudlets similaires afin d'atteindre l'évolutivité. Les résultats expérimentaux montrent que l'approche proposée est capable de fournir une grande fiabilité pour la disponibilité et la découverte des ressources. Cependant, elle souffre d'une faible efficacité et d'un trafic réseau élevé lorsque le nombre de clusters devient important.

L'article (Chavan et Kaveri 2014) propose une architecture basée sur le regroupement des machines virtuelles dans les centres de données Cloud en termes de divers attributs, y compris la RAM, le type de système d'exploitation, la configuration matérielle, etc. Le Clustering des ressources applique la méthode de Clustering K-Means qui aide les machines virtuelles à se reconfigurer et facilite la planification des ressources. Cela conduira à une meilleure expérience utilisateur avec une plus grande disponibilité des ressources et une meilleure évolutivité.

Dans (Saravanakumar et al. 2021) ont proposé un système qui se concentre sur le regroupement des machines virtuelles en tenant compte de la taille de la tâche demandée et du niveau de la bande

passante. L'objectif principal du Clustering de machines virtuelles proposé est de faire correspondre la tâche à la machine virtuelle appropriée pour être exécutée avec la bande passante afin d'obtenir une haute disponibilité et de maintenir une haute fiabilité dans le Cloud Computing. L'algorithme proposé prouve sa performance en permettant de réduire le temps d'exécution des tâches tout en améliorant l'efficacité.

Wu et al. (Wu et al. 2014) ont proposé une stratégie de catégorisation des ressources de services Cloud. La recherche a présenté une amélioration de l'algorithme de classification naïve bayésienne original en introduisant des poids pour calculer la similarité des différentes caractéristiques dans l'ensemble de données des ressources. En outre, les auteurs ont mis en œuvre un modèle de programmation parallèle utilisant Hadoop combiné à MapReduce pour mettre en œuvre la parallélisation de l'algorithme amélioré. Les résultats expérimentaux montrent que la stratégie proposée est efficace dans les environnements Cloud dynamiques à grande échelle et offre une meilleure performance de classification des ressources via la parallélisation. Cependant, cette recherche ne tient pas compte des applications et ne classifie que les ressources de différents types.

Yoori et al. ont proposé dans (Oh et Kim 2019) une méthode de regroupement dynamique de ressources similaires en appliquant l'algorithme des cartes auto-organisatrices (en anglais, Self-Organizing Map, abrégé SOM) et l'algorithme K-Means dans des environnements Cloud Computing hybride. Sur la base des clusters résultants, une méthode de recommandation de ressources rentable est appliquée aux utilisateurs du Cloud,permettant derefléter les caractéristiques des applications. Les résultats expérimentaux prouvent l'efficacité des méthodes proposées en termes de temps d'exécution, de disponibilité et de coûts d'utilisation des ressources.

Dans (Durgadevi et Srinivasan 2016), une nouvelle technique de découverte optimale et d'allocation dynamique des ressources est proposée. La technique proposée utilise l'algorithme « Modified Hierarchical Agglomerative Clustering, abrégé MHAC » pour la découverte des ressources et la construction de l'arbre. Ensuite, les ressources sont allouées à l'aide de l'algorithme hybride de recherche des coucous et de colonies d'abeilles artificielles (en anglais Hybrid Artificial Bee Colony Cuckoo Search, abrégé HABCCS). Dans ce contexte, la colonie d'abeilles artificielle est utilisée pour optimiser le chemin de construction de l'arbre et la recherche du coucou est utilisée pour modifier l'algorithme de la colonie d'abeilles artificielle. D'après les résultats passionnants, il est clair que cette technique permet d'améliorer la disponibilité des ressources souhaitées qui sont allouées de la manière la plus efficace avec un temps de calcul minimum.

Meenakshi et al (Meenakshi et al. 2019) ont proposé une technique efficace de fourniture de ressources dans le Cloud en utilisant l'algorithme de regroupement K-Means et l'optimisation Gray Wolf (en anglais Gray Wolf Optimisation, souvent abrégé GWO). La technique proposée utilise GWO pour la priorisation et le K-Means pour analyser les mesures de QoS afin d'allouer les ressources du Cloud de la manière la plus optimale possible et de répondre aux demandes des utilisateurs. Les résultats expérimentaux prouvent la performance de l'approche proposée en termes de précision de Clustering, permettant de fournir une QoS de service satisfaisante en termes de disponibilité, d'équilibrage de charge, d'utilisation de la mémoire et de temps d'exécution.

Ces travaux de recherche ont montré que les approches existantes de regroupement des ressources et des services ont tendance à se concentrer sur la catégorisation de petits ensembles de données et que leurs performances diminuent dans les environnements à grande échelle. Le tableau 5.1 résume les approches ci-dessus en mettant l'accent sur leurs avantages et leurs inconvénients. En outre, il présente les principaux critères de performance considérés, à savoir : (a) la disponibilité, (b) l'évolutivité, (c) la

réduction des coûts, (d) le temps d'exécution, (e) la consommation de bande passante, (f) l'équilibrage des charges, (g) la précision.

Tableau 5. 1: Aperçu des principaux travaux fondés sur les techniques de Clustering pour améliorer la disponibilité dans le Cloud Computing.

Référence	Algorithmes utilisés	Avantages	Inconvénients	Principales caractéristiques considérées						
				a	b	c	d	e	f	g
(Malathy at al. 2013)	Expectation Maximization "EM" clustering algorithm	Disponibilité et tolérance aux pannes accrues. Succès et précision élevés pour la découverte des ressources. Amélioration de l'évolutivité et de l'utilisation des ressources.	Faible efficacité. Trafic réseau élevé. Frais généraux élevés lorsque le nombre de clusters devient important.	☑	☑	☑	☒	☒	☒	☑
(Chavan et Kaveri 2014)	K-Means clustering algorithm	Haute disponibilité et évolutivité. Amélioration de l'équilibrage des charges, du coût et de l'utilisation des ressources. Minimisation du temps de réponse et de l'utilisation de la bande passante.	L'algorithme nécessite une initialisation appropriée du nombre de clusters.	☑	☑	☑	☑	☑	☑	☒
(Saravanakumar et al. 2021)	Clustering Algorithms (not specified)	Maintien d'une haute disponibilité, fiabilité, latence et efficacité. Réduction du temps d'exécution des tâches et de la consommation de bande passante.	non spécifié	☑	☑	☑	☑	☑	☒	☒
(Wu et al. 2014)	Parallel Bayesian classification algorithm	Amélioration de la précision et des performances de la classification des ressources. Haute disponibilité et efficacité dans les environnements Cloud hétérogènes dynamiques à grande échelle.	Ne tient pas compte des applications et ne classe que les ressources de différents types.	☑	☑	☑	☑	☒	☒	☑
(Oh et Kim 2019)	SOM self-organizing maps algorithm and k-means algorithm	Amélioration de la disponibilité et de l'efficacité. Réduction du temps d'exécution. Optimisation des coûts d'utilisation des ressources et de la consommation énergétique des VM.	Ne s'adapte pas aux environnements Cloud de taille importante	☑	☑	☑	☑	☑	☒	☒
(Durgadevi et Srinivasan 2016)	Modified Hierarchical Agglomerative Clustering (MHAC) algorithm, Hybrid artificial bee colony and cuckoo search (HABCCS) algorithm	Amélioration de la disponibilité et de l'efficacité. Réduction des coûts, de l'utilisation de la mémoire et du temps d'exécution.	non spécifié	☑	☑	☑	☑	☑	☒	☒
(Meenakshi et al. 2019)	K-means clustering and gray wolf optimization (GWO) partitioning technique	Précision et efficacité élevées du Clustering. Amélioration de l'équilibrage des charges, de la disponibilité, de l'utilisation de la mémoire et du temps d'exécution.	L'analyse des charges de travail dans le Cloud est moins efficace.	☑	☒	☒	☑	☒	☑	☑

À mesure que le modèle SaaS gagne en ampleur, l'évaluation et l'amélioration de la qualité des services SaaS fournis deviennent de plus en plus indispensables. Dans cette optique, les chercheurs ont récemment envisagé l'utilisation de différentes techniques de regroupement dans le modèle SaaS.

V.Kanagalakshmi et R.Gnanaselvam ont présenté dans (Kanagalakshmi et Gnanaselvam 2017) un modèle d'estimation de la qualité des services logiciels SaaS dans le Cloud Computing, basé sur des techniques de Clustering. Le modèle d'évaluation proposé comporte différentes étapes à suivre pour les données SaaS, à savoir 1) Extraire les données SaaS, 2) Transformer les données SaaS, 3) Charger au format RDBMS/MDBMS, 4) Appliquer l'algorithme de Clustering, 5) Obtenir le cluster approprié. Le modèle de mise en cluster proposé aide les fournisseurs de services à augmenter la disponibilité et l'évolutivité des services SaaS. Il aide également les utilisateurs du Cloud à évaluer

les services logiciels potentiels disponibles dans l'environnement du Cloud. Pour un objectif similaire, les mêmes étapes ont été suivies par D.Jagli et A.Gupta dans (Jagli et Gupta 2013), bien qu'à l'étape 4, les auteurs aient utilisé les algorithmes de Clustering hiérarchique dur les plus populaires.

D.Jagli et al. ont proposé dans (Jagli et al. 2014) une approche pour l'évaluation de SaaS basée sur l'algorithme Constraint Based Clustering (abrégé CBC) pour l'exploration de données. L'approche proposée prend en compte les exigences des clients ainsi que les attributs de qualité des services SaaS. Elle commence par la création de clusters en se basant sur l'algorithme K-Means. En fonction des contraintes des utilisateurs, chaque cluster est ensuite transformé en micro-cluster. Puis, les contraintes spécifiées par les utilisateurs sont utilisées pour identifier les impasses.Si l'impasse est trouvée, alors le cluster final se forme, sinon "break le micro cluster" et répéter l'opération jusqu'à ce qu'une impasse soit trouvée.

D.Jagli et al. ont présenté dans (Jagli et al. 2017) divers algorithmes de Clustering (K-Means Clustering, B-Cluster, Hierachical) pour évaluer la qualité du SaaS. Les données SaaS ont été collectées à partir d'AWS et analysées à l'aide de l'outil R. L'objectif de cette recherche est de fournir un système de prise de décision avec des solutions optimales pour les utilisateurs et les fournisseurs de services Cloud afin de pouvoir sélectionner rapidement et facilement des services logiciels potentiels en fonction des exigences spécifiées. Les résultats expérimentaux montrent que le temps pris par l'algorithme de Clustering K-Means était beaucoup moins important par rapport aux autres méthodes de Clustering. Pour atteindre les objectifs ci-dessus, les mêmes auteurs ont utilisé dans (Jagli et al. 2018(a)) l'algorithme de Clustering "Partitioning Around Medoids" (abrégé PAM). L'approche proposée commence par la collecte de données d'attributs SaaS sur lesquelles des techniques de prétraitement sont appliquées. Ensuite, elle procède àune analyse de cluster qui permet de définir le nombre optimal de clusters en utilisant la méthode du coude elbow method et d'appliquer l'algorithme PAM pour former les clusters de données. Les résultats expérimentaux montrent l'efficacité et la robustesse de l'algorithme PAM par rapport à K-Means en présence de bruit et de valeurs aberrantes.

R.Kamalraj et al. ont proposé dans (Kamalraj et al. 2012) une approche basée sur des techniques de Data Mining pour créer des groupes de produits SaaS. Cette approche vise à minimiser le temps nécessaire pour trouver les produits requis afin de répondre aux attentes des clients, de faire peu d'efforts pour les associer à de nouveaux produits et d'augmenter les gains commerciaux. Elle permet de collecter les données et de les regrouper en une structure utilisée comme entrée pour l'algorithme de Clustering. Ensuite, les règles d'association (en anglais Association Rules) sont utilisées pour trouver les relations entre les différents éléments de données des clusters disponibles afin de présenter les nouveaux produits aux clients. Dans cette approche, les techniques du Data Mining sont appliquées à l'analyse du panier du marché (en anglais Market Basket Analysis) du Cloud Computing, permettant d'analyser les groupes de services, d'identifier les services fréquemment utilisés par les clients et d'associer les nouveaux produits en tant que version d'"évaluation gratuite", ce qui peut améliorer les bénéfices commerciaux en attirant les clients avec ses nouvelles fonctionnalités.

D.Jagli et al. ont proposé dans (Jagli et al. 2018(b)) un modèle d'évaluation de la qualité des SaaS dans l'environnement du Cloud Computing, basé sur l'algorithme de Clustering Estimation et Maximisation (en anglais Estimation and Maximization, abrégé EM). L'approche commence par collecter les données SaaS sur lesquelles des techniques de prétraitement sont appliquées. Elle utilise ensuite l'algorithme de regroupement EM pour former des clusters de données. La dernière étape consiste à visualiser les données de sortie, ce qui permet d'interpréter et d'évaluer les attributs.

Dans une étude similaire (Jagli et al. 2018(c)), les mêmes auteurs ont proposé un modèle de qualité pour les services SaaS appelé Software as a Service Quality model (SAASQUAL). Le modèle proposé vise à évaluer les services SaaS et à différencier la qualité de leurs fournisseurs en se basant non seulement sur l'algorithme EM, mais aussi sur différents attributs et métriques qui mesurent la qualité des logiciels. L'algorithme EM démontre son efficacité pour les ensembles de données incomplets et prouve sa capacité à aider les utilisateurs du Cloud à sélectionner les meilleurs services SaaS ainsi que les fournisseurs SaaS à améliorer leurs produits pour fidéliser leurs clients dans un monde compétitif.

Dans (Yusoh et Tang 2010), Z. I. M. Yusoh et M. Tang ont proposé un algorithme génétique basé sur les pénalités pour le problème du placement du SaaS-Cloud composite, qui prend en compte non seulement le placement des composants logiciels SaaS, mais aussi le placement des données SaaS. Le problème peut être classé comme un problème d'optimisation dont le but est d'optimiser les performances du SaaS en fonction de son temps d'exécution estimé qui dépend du temps passé à chercher la solution de placement pour chaque configuration. Cette proposition constitue la première tentative de placement de SaaS avec ses données sur les serveurs du fournisseur de Cloud. Dans (Yusoh et Tang 2012 (a)), les auteurs ont développé un algorithme de Clustering génétique (GGA) pour le regroupement de plusieurs SaaS composites dans des clusters Cloud afin de traiter la question du placement des SaaS composites et de l'optimisation de leurs ressources. L'approche proposée permet de minimiser l'utilisation des ressources SaaS sans violer leurs SLA en reconfigurant l'emplacement des composants de l'application tout en maintenant les performances de l'application, en équilibrant la distribution thermique entre les serveurs et en minimisant la consommation de données. Un algorithme qui oscille entre les deux derniers travaux a été proposé dans (Yusoh et Tang 2012 (b)). Il ressemble à celui présenté dans (Yusoh et Tang 2012 (a)), à la différence de l'introduction de la notion de pénalité utilisée dans l'algorithme proposé dans (Yusoh et Tang 2010), sachant qu'il ne considère pas le placement des données SaaS. Ces recherches démontrent la faisabilité et l'évolutivité de l'AG.

Le tableau 5.2 résume les travaux de recherche ci-dessus en mettant l'accent sur leurs avantages et leurs inconvénients. En tant qu'observation de recherche, il convient de noter que la recherche qui traite de l'intégration de l'exploration de données dans le modèle SaaS est encore limitée. Les différents travaux présentés ci-dessus se concentrent principalement sur l'évaluation de la qualité des services SaaS plutôt que sur l'amélioration des solutions SaaS fournies. Bien qu'ils ne traitent pas de l'aspect de l'incorporation des techniques de Clustering pour améliorer la disponibilité des services SaaS-Cloud basés sur les réseaux P2P, nous considérons qu'il pourrait s'agir d'une étape importante vers une plateforme de fourniture de services Cloud hautement disponible basée sur le P2P.

Tableau 5. 2: Comparaison des algorithmes de Data Mining implémentés dans SaaS

Référence	Cadre utilisé	Algorithmes Utilisés	Type de données gérées	Objectifs	Avantages	Limitations	Évaluation axée sur			Attributs de Qualité						
							Perspective Utilisateur	Perspective Fournisseur	Attributs de Qualité	Efficacité	Optimisation des Ressources	Traitement du Bruit	Disponibilité	Évolutivité	Réutilisabilité
(Jagli et Gupta 2013)	Environnement Cloud SaaS	Différentes méthodes de Clustering	Non structurées, structurées	Évaluation du SaaS dans le Cloud.	Utile aux utilisateurs et aux fournisseurs pour mesurer la qualité des services SaaS.	Ne pas prendre en compte les attributs de qualité du logiciel.	✓	✓							
(Jagli et al. 2014)		Constraint Based Clustering (CBC), K-Means	structurées	Évaluation multicritères de la qualité potentielle du SaaS.	Amélioration de la qualité du service et réduction des coûts, plus grande satisfaction et fidélité des clients.	Les mesures de qualité des clusters sont moins efficaces.	✓	✓	✓	✓					
(Jagli et al. 2017)		K-Means, Hierarchical, BiCluster	structurées	Optimisation de la prise de décision (sélection/provision) sur les services SaaS qualifiés.	Grande rapidité et flexibilité de la spécification de qualité SaaS.	La détermination des paramètres d'entrée des algorithmes nécessite une connaissance du domaine.	✓	✓							
(Jagli et al. 2018(a))		Partitioning Around Medoids (Pam) Algorithm	structurées	Évaluer la qualité de tout produit SaaS avec des attributs de qualité identifiés dans le Cloud.	Haute performance et robustesse en présence de bruit.	Faible analyse de grands ensembles de données, moins de stabilité.			✓	✓		✓	✓		✓
(Kamalraj et al. 2012)		Association Rule Learning	structurées	Analyser le panier de marché SaaS et identifier le comportement des clients pour augmenter les gains commerciaux.	Réduction du temps et des efforts consacrés à la publicité des nouveaux produits SaaS, satisfaction élevée des utilisateurs, augmentation des retours financiers.	Faible performance d'algorithmes spécifiques pour le Clustering et l'association.	✓	✓							
(Jagli et al. 2018(b))		Expectation–Maximization algorithm	structurées	Évaluation de la qualité du SaaS pour sélectionner le meilleur service en fonction des besoins des utilisateurs.	Efficacité et estimation élevée pour les ensembles de données incomplets.	Taux de convergence lent lorsque la dimension des données augmente.			✓	✓		✓	✓	✓	✓
(Jagli et al. 2018(c))		Expectation–Maximization algorithm	structurées	Évaluer la sélection et la fourniture de SaaS en fonction des besoins des utilisateurs à l'aide d'un modèle de qualité SaaS.	Automatisation de l'évaluation complète de la qualité du SaaS.	Convergence lente et généralement vers les optima locaux.	✓	✓	✓	✓			✓	✓	✓
(Yusoh et Tang 2010)		Penalty-based Genetic Algorithm	structurées	Placement efficace des composants SaaS en tenant compte du stockage de leurs données dans le Cloud.	Optimisation des performances et de l'utilisation des ressources SaaS. Excellente évolutivité.	Centralisation. Augmentation parallèle du temps de calcul en fonction de la taille du réseau.	✓		✓	✓		✓	✓	✓	✓
(Yusoh et Tang 2012(a))		A Penalty-based Grouping Genetic Algorithm	structurées	Optimisation de la gestion dynamique de l'allocation des ressources pour le SaaS composite dans le Cloud.	Performances et évolutivité élevées, coût réduit des ressources utilisées, migration minimale des VM.	Temps de calcul de l'algorithme long.			✓	✓				✓	✓
(Yusoh et Tang 2012(b))		Grouping Genetic Algorithm	structurées	Maintenir les performances du SaaS composite par une gestion dynamique et optimale de l'allocation des ressources dans les centres de données en Cloud.	Efficacité et performance élevées, en minimisant l'utilisation des ressources SaaS.	Temps de calcul long.	✓		✓	✓	✓			✓	✓

6. Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur l'utilisation du Data Mining, principalement sa célèbre technique « Clustering », pour améliorer la qualité et la disponibilité des services de Cloud, en particulier le modèle SaaS. Tout d'abord, nous avons fait un tour d'horizon des principaux concepts relatifs au Data Mining. Ensuite, nous avons expliqué comment le Data Mining et le Cloud Computing, en particulier le modèle SaaS, peuvent être sollicités réciproquement et coopérer en vue de tirer parti des avantages de chacun. Dans ce contexte, nous avons fourni une vision générale du processus d'intégration du Data Mining dans le modèle SaaS-Cloud Computing afin d'améliorer la qualité de ses services. Suite à quoi, nous avons présenté les notions et les concepts de base du Clustering où nous avons introduit sa définition et ses principales mesures de distance et de similarité. Par ailleurs, nous avons détaillé les approches, les techniques et les algorithmes de Clustering choisis pour être utilisés dans notre travail de recherche. Par la suite, nous avons établi les techniques de mesure de performance les plus répandues dans les recherches dans ce domaine. Enfin, nous avons passé en revue des solutions et travaux de recherche importants portant sur l'utilisation des algorithmes de Clustering pour la disponibilité des services dans le Cloud Computing, particulièrement dans le modèle SaaS.

Partie 2 : Contribution

Chapitre 6: Un Système Cloud-SaaS Basé sur un Réseau P2P Hybride pour le Traitement des Services à Distance

Sommaire

1. Introduction
2. Solution Proposée
   - 2.1. Modèle en Couches
     - 2.1.1. Couche Cloud
     - 2.1.2. Couche des Serveurs Cloud Partiels
     - 2.1.3. Couche Clients
   - 2.2. Description Détaillée
     - 2.2.1. Inscription et Connexion
     - 2.2.2. Hébergement des Services
     - 2.2.3. Recherche d'un Service
     - 2.2.4. Invocation de Service
3. Evaluation et Analyse des Résultats
   - 3.1. Environnement Expérimental
   - 3.2. Résultat et Discussion
4. Conclusion

1. Introduction

Dans ce chapitre, nous présentons le système proposé, qui consiste en une nouvelle architecture Cloud qui utilise un réseau P2P hybride « H-P2P » comme infrastructure pour fournir des services sous forme de logiciel en tant que service «SaaS» aux pairs qui n'en disposent pas. Il permet aux pairs d'utiliser et d'exécuter des services SaaS qui sont initialement fournis par le Cloud mais hébergés et exécutés par les pairs eux-mêmes. Ces services sont exécutés par invocation à l'aide de la technologie Remote Method Invocation « RMI » sur les pairs qui les hébergent. A la différence des définitions de services existantes, nous définissons nos services sous forme de classes (ou fonctions) java, en exploitant les techniques de l'introspection (Doudoux 2019) qui nous permettent de développer les tâches de création, d'hébergement et d'utilisation de services. L'hébergement d'un service par un pair du réseau se fait directement en téléchargeant sa classe java, sans nécessité d'un traitement ou précondition, ce qui permet au système d'être utilisé par un plus large public. La suite de ce chapitre est organisée comme suit: nous commençons par présenter une description complète et détaillée de notre système proposé. Suite à quoi, nous décrivons le cadre expérimental et nous présentons les résultats de la simulation obtenus. Ce travail a été l'objet de notre article de conférence « Hybrid P2P Network-Based Remote Treatment SaaS Cloud Computing, 2019 ».

2. Solution Proposée

Le système proposé consiste en un Cloud qui utilise un réseau Pair-à-Pair comme infrastructure pour fournir des services sous forme SaaS à des pairs qui ne les disposent pas. Ces services sont exécutés par invocation sur les pairs qui les hébergent. Il convient de noter que le réseau Pair-à-Pair hybride est semblable à l'architecture du Cloud Computing, ce qui a favorisé notre choix de l'utiliser comme infrastructure pour notre Cloud.

Notre Cloud est un Cloud distribué dynamique qui possède une infrastructure minimale composée d'un nœud central qui aura le rôle d'*administrateur* et un ensemble de nœuds appelés *super-pairs* (en anglais super-peers) qui agissent comme des *Serveurs Cloud Partiels* (en anglais *Partial Cloud Server*, abrégé *PCS*) distribués géographiquement. Les *PCSs* effectuent les tâches d'indexation et de traitement des demandes de service, et sont connectés entre eux par une couche réseau. L'infrastructure réelle et fonctionnelle sera représentée par un ensemble de nœuds formant le réseau P2P (incluant les *PCSs*) jouant le rôle de *Clients*. Un client peut avoir deux rôles différents. C'est un *Client Participant* lorsqu'il héberge des services et les exécute pour le compte d'autres clients qui auront dans ce cas le rôle d'*Utilisateur*. Lorsqu'un client rejoint le réseau, il se connecte au *PCS* le plus proche géographiquement, qui fait office de serveur central pour tous les clients qui y sont connectés, formant un cluster. L'administrateur se charge de l'approvisionnement des services. Ces derniers sont représentés par des méthodes contenues dans des classes java téléchargeables, que chaque pair peut héberger ou utiliser à distance. Les *PCSs* maintiennent la trace de leurs clients connectés et les services qu'ils hébergent. Lorsqu'un utilisateur demande un service, le *PCS* auquel il est connecté reçoit la demande, effectue une recherche et lui renvoie les informations des participants hébergeant le service demandé. Le client utilisateur choisit par la suite un client participant et invoque le service dans celui-ci. Une fois la demande traitée par le participant, ce dernier renvoie le résultat du service au client utilisateur demandeur. Le modèle en couches du système et son architecture globale sont illustrés successivement dans la figure 6.1 et la figure 6.2.

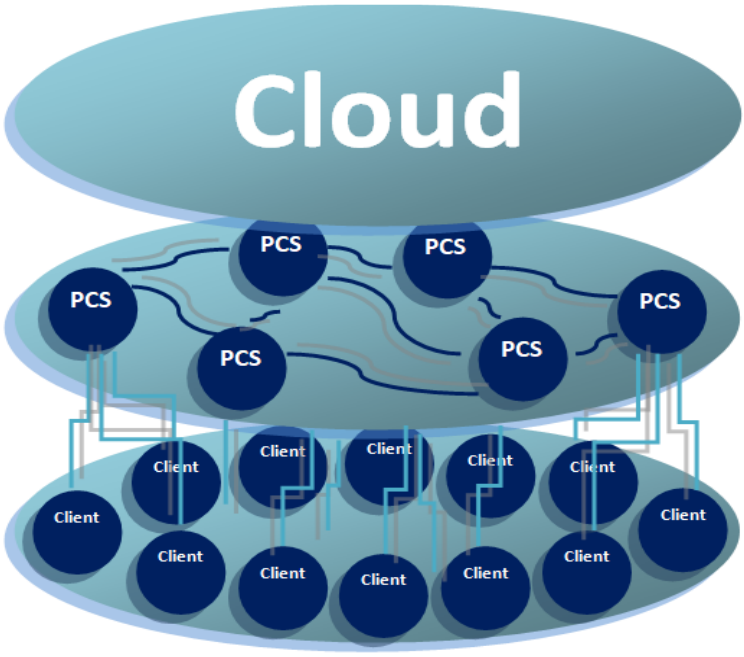


Figure 6. 1: Modèle en Couches du Système Proposé.

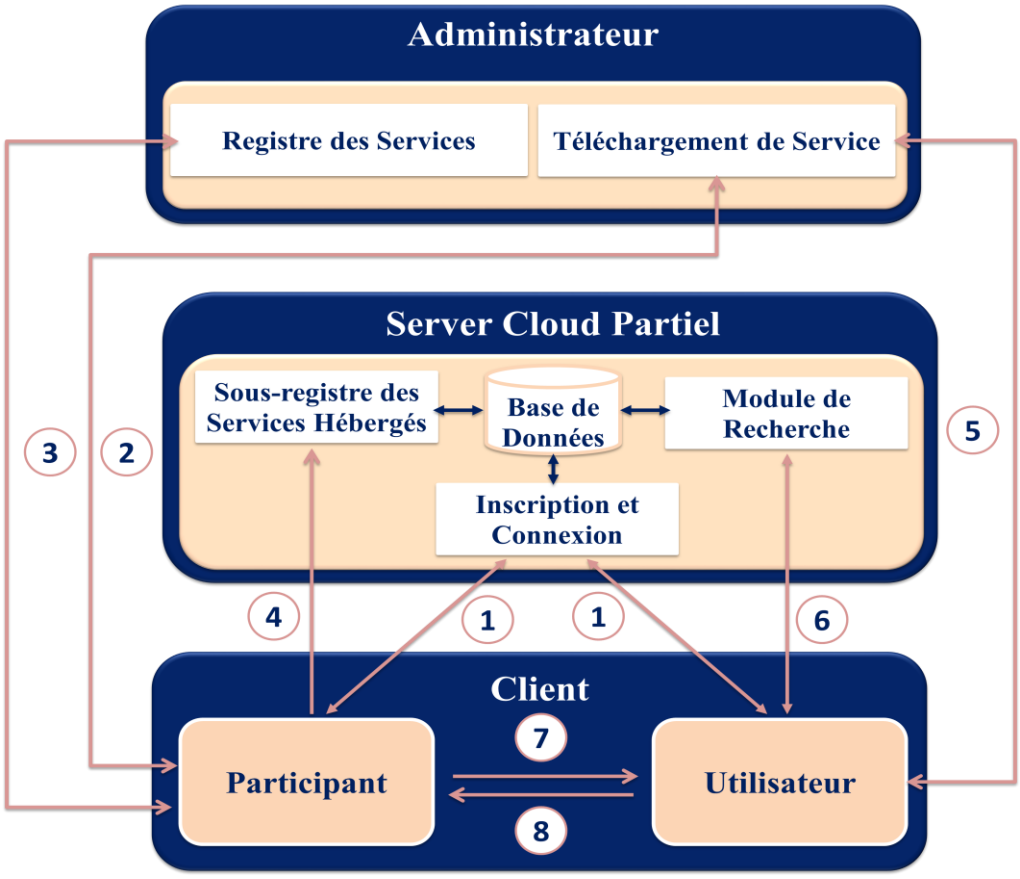


Figure 6. 2: Architecture Globale du Système.

2.1. Modèle en Couches

Le système proposé se compose de trois couches interdépendantes comme le montre la figure 6.1.

2.1.1. Couche Cloud

La *Couche Cloud* (en anglais *Cloud Layer*) permet principalement d'exposer la liste des services offerts par le système aux clients à des fins d'hébergement ou d'utilisation. L'acteur principal de cette couche est l'administrateur central qui se charge de l'alimentation du système de services ainsi que de leur gestion (l'approvisionnement). L'administrateur est le seul responsable de l'ajout, la suppression et la mise à jour de la gamme de services. Une interface permet d'afficher l'ensemble de services disponibles aux clients pour leurs besoins d'hébergement ou de recherche. En d'autres termes, elle permet aux participants d'héberger un ensemble de services offerts qu'ils vont exécuter à la demande des utilisateurs. Ces services sont conservés dans un répertoire spécifique qui détermine la liste des services offerts par le système. Les services offerts par notre système sont des programmes qui fournissent un traitement particulier. Un service est une déclaration d'une méthode Java où les entrées des services sont les arguments de la méthode et la sortie du service (rendering) est le résultat retourné par la méthode. La *Couche Cloud* ne s'occupe ni du traitement des demandes ni de l'exécution des services. Ces tâches sont réalisées successivement par la *Couche des Serveurs Cloud Partiels* et par les pairs participants. La mise à jour d'un service par l'administrateur nécessite la modification et la recompilation de la classe qui le contient.

2.1.2. Couche des Serveurs Cloud Partiels

La *Couche des Serveurs Cloud Partiels* (en anglais *Partial Cloud Servers Layer*, abrégé *PCSs Layer*) est composée d'un ensemble de serveurs Cloud partiels, interconnectés entre eux. Elle est responsable du traitement des demandes des services, de la gestion et de la coordination entre les clients. Chaque *PCS* maintient continuellement la trace de ses clients connectés et de leurs services hébergés. Pour cette tâche, on utilise une base de données SQL qui regroupe des informations sur chaque client et sur ses services hébergés. Le *PCS* joue le rôle d'un serveur d'index pour les utilisateurs qui y sont connectés. C'est le seul responsable du traitement de leurs demandes de services. À la réception d'une requête de recherche d'un service de la part d'un utilisateur, le *PCS* lance une recherche locale au niveau de son cluster et renvoie comme résultat à l'utilisateur demandeur la liste des pairs participants qui détiennent le service demandé. Si la recherche au niveau de son cluster est infructueuse, il communique avec ses *PCSs* voisins afin de localiser le service souhaité dans les autres clusters.

2.1.3. Couche Clients

L'application client est principalement conçue pour permettre aux clients de rechercher, d'héberger ou d'exécuter à distance les services offerts par le Cloud. Elle doit être installée chez tous les pairs du réseau. Après l'inscription au système, le client peut choisir d'être un participant ou un utilisateur. Le participant est un client effectif dans le système. Il joue le rôle d'un hébergeur et fournisseur de service en même temps. Il choisi un ensemble de services qu'il veut héberger en les sélectionnant à partir du nœud central. En revanche, l'utilisateur est un client consommateur qui cherche à exploiter ces services pour son intérêt. Cette couche exploite le principe des réseaux P2P pour permettre d'effectuer une connexion directe entre les clients lors de l'invocation de services sans aucune intervention d'un tiers.

2.2. Description Détaillée

Les différentes taches effectuées entre les trois couches sont décrites ci-dessous:

2.2.1. Inscription et Connexion

Cette étape (opération 1, figure 6.2) permet aux clients (participants/utilisateurs) de se connecter au système avec un identifiant (nom d'utilisateur) et un mot de passe pour s'identifier, sachant qu'ils doivent s'inscrire préalablement en créant un compte. L'authentification du client est assurée par le

PCS auquel il s'y est identifié la dernière fois. Si l'authentification a été faite par un PCS autre que celui auquel il s'est connecté, ce dernier PCS rapatrie les informations du client chez lui. L'objectif principal est de permettre aux *PCSs* de connaitre les informations de leurs clients afin de pouvoir localiser les services qui y sont hébergés ainsi que de maintenir la trace des clients connectés au système.

2.2.2. Hébergement des Services

Cette tâche (opération 2, 3 et 4, figure 6.2) permet aux participants connectés de sélectionner et de télécharger un ensemble de services parmi les services offerts par le Cloud (tâche d'hébergement). La sélection des services à héberger est réalisée via une interface. Celle-ci permet aux participants de télécharger dynamiquement les classes Java compilées qui contiennent les services représentés par une méthode Java. Après l'hébergement, le participant effectue une analyse des services hébergés pour détecter les paramètres (entrée/sortie) de chaque service. Il crée un fichier *manifest* en format *XML* qui sera utilisé ultérieurement pour la génération de l'interface associée au service. Le participant informe son *PCS* des services qui ont été récemment hébergés. À son tour, le serveur Cloud partiel met à jour sa base de données. Pour réaliser cette tâche, l'administrateur dispose d'un module *Registre des Services* « en anglais *ServiceRegitry* » qui permet au participant d'avoir la liste des services offerts par le Cloud et d'un module *Téléchargement de Service* « en anglais *ServiceUpload* » qui permet de télécharger les services vers le pair participant. De son côté, le participant dispose d'un module *Demande de la Liste des Services* « en anglais *RequestServiceList* » qui lui permet d'obtenir la liste des services offerts par le Cloud, d'un module *Sélection de Service* « en anglais *ServiceSelection* » qui lui permet de choisir les services à héberger et de télécharger les classes correspondantes, d'un module *Analyse de Service* « en anglais *ServiceAnalysis*» pour créer les fichiers *manifest* du service et d'un module *Informations sur les Services Hébergés* « en anglais *HostedServiceInformation*» qui lui permet d'informer son *PCS* de ses services hébergés. Le *PCS* dispose d'un module *Sous-registre des Services Hébergés* « en anglais *HostedServiceSubRegistry*» qui lui permet de garder la trace des services hébergés par les participants (figure 6.3). Le schéma présenté dans la figure 6.4 explique la procédure d'hébergement.

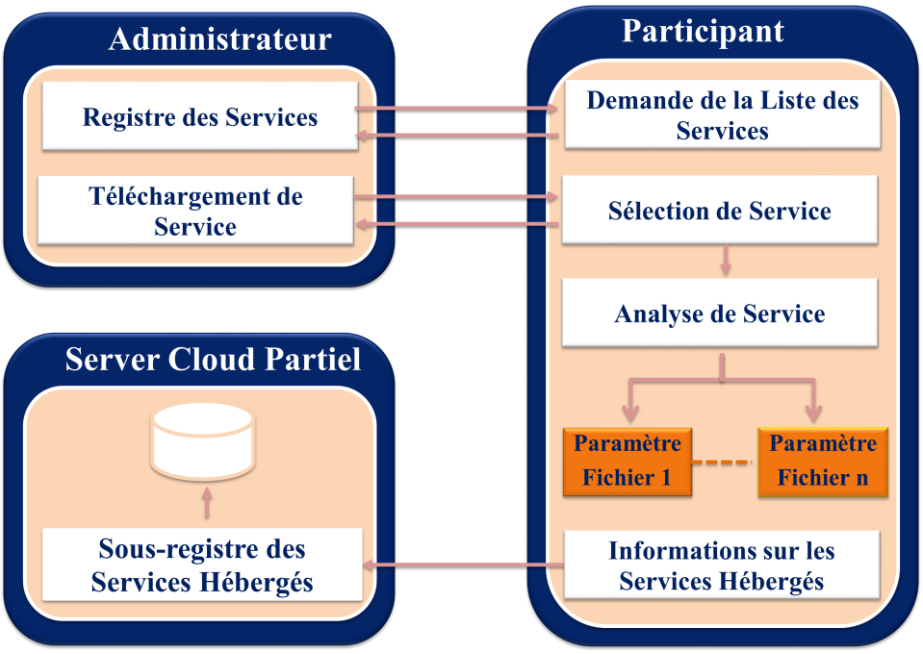


Figure 6. 3: Etape d'Hébergement.

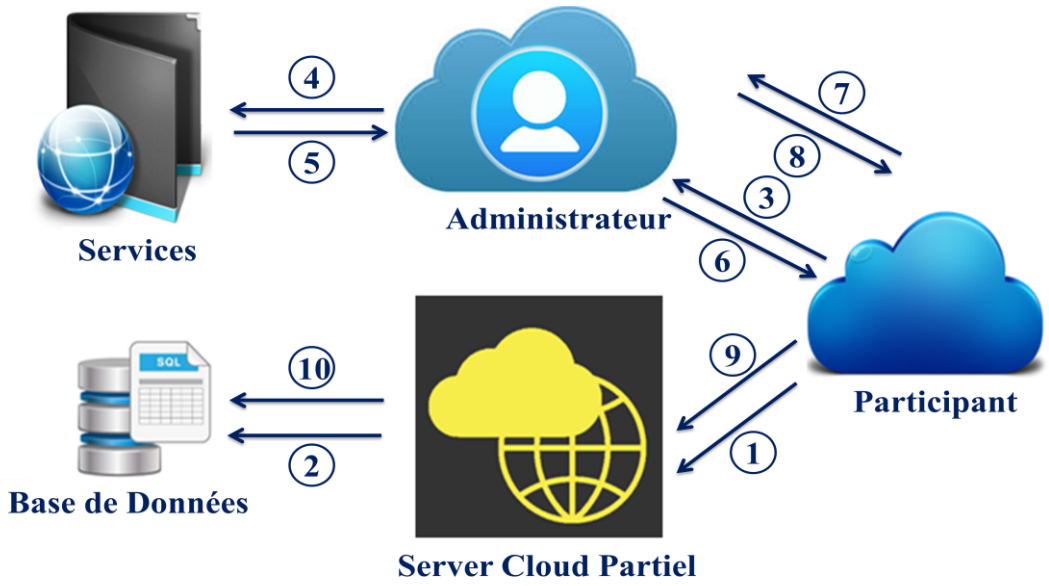


Figure 6. 4: Procédure d'Hébergement.

1. Le participant s'identifie en se connectant au système.
2. Le PCS met à jour sa base de données des clients connectés.
3. Le participant demande à l'administrateur la liste des services offerts par le système selon un critère.
4. L'administrateur consulte le répertoire des services.
5. L'administrateur génère une liste des services disponibles selon le critère.
6. L'administrateur retourne la liste des services au participant.
7. Le participant choisit les services qu'il veut héberger et envoie une demande à l'administrateur.
8. Le participant télécharge les classes contenant les services sélectionnés.
9. Le participant informe son PCS des nouveaux services.
10. Le PCS met à jour la base de données.

2.2.3. Recherche d'un Service

Cette étape (opération 5 et 6, figure 6.2) permet aux utilisateurs connectés de bénéficier des services offerts par le Cloud. Elle commence par une recherche au niveau de la couche *Cloud* afin de choisir un service parmi les services offerts par le système, et s'achève par une recherche au niveau de la couche des *Serveurs Cloud Partiel* pour localiser le service. Dans la première phase, l'utilisateur dispose d'un module *Demande de la Liste des Services* « en anglais *RequestServiceList* » qui lui permet d'avoir la liste des services offerts par le système et d'en choisir un, et un module *Recherche de Service* « en anglais *ServiceSearch* » qui lui permet d'envoyer une requête de recherche du service sélectionné à son *PCS* et d'obtenir une liste des participants hébergeant le service. De son côté, le *PCS* dispose d'un module *Réception de la Demande de Service* « en anglais *ReceptionRequestService* » qui lui permet de recevoir la demande de recherche et d'un module *Recherche* « an anglais *Search* » qui lui permet de lancer une recherche dans sa base de données locale pour le service demandé et d'établir une liste de participants hébergeant le service (figure 6.5). Le module *Réception de la Demande de Service* retourne le résultat à l'utilisateur. Si la recherche locale n'aboutit pas, le module *Recherche* permet au

PCS de propager la requête de recherche aux *PCSs* voisins par le mécanisme d'inondation (Pourebrahimi et al. 2005). La recherche est arrêtée si la liste des participants retournée par l'un des voisins n'est pas vide ou bien le TTL associé à la requête de recherche atteint sa valeur maximale. Le schéma présenté dans la figure 6.6 explique la procédure de recherche.

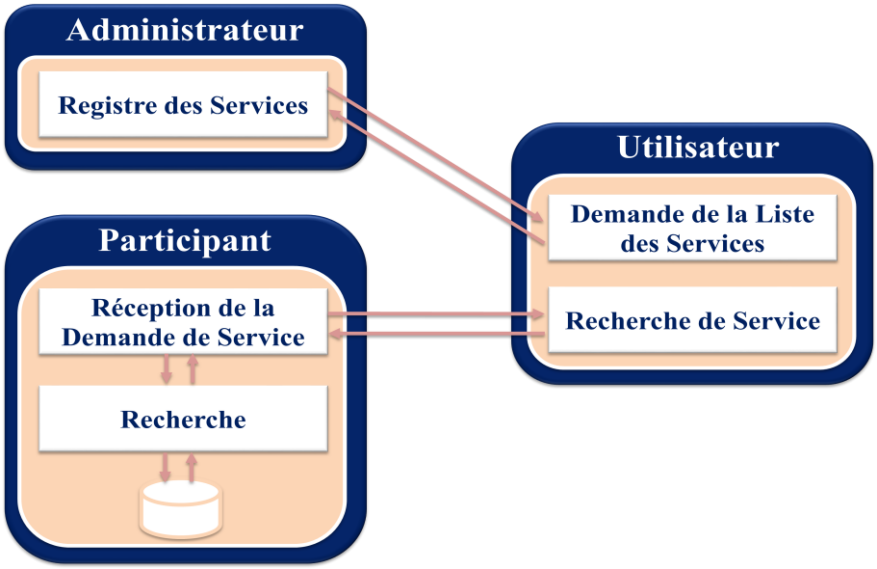


Figure 6. 5: Etape de Recherche.

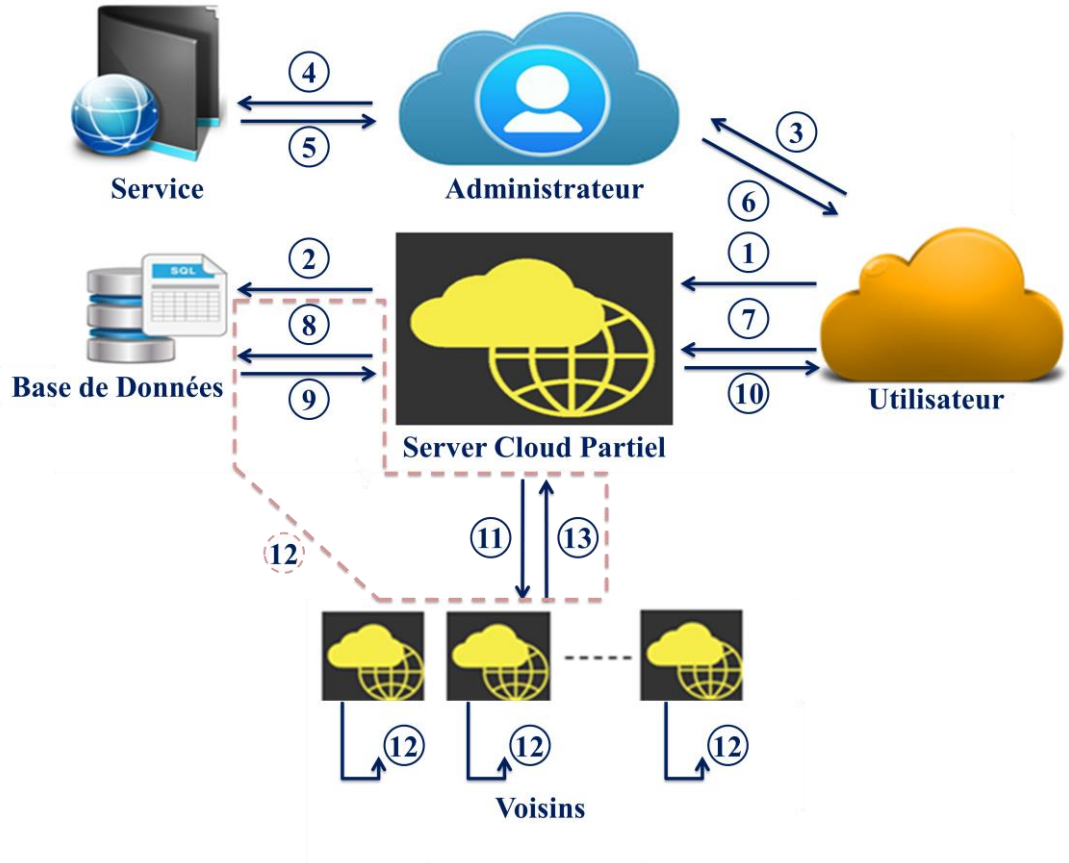


Figure 6. 6: Procédure de Recherche.

1. Le participant s'identifie en se connectant au système.
2. Le *PCS* met à jour sa base de données des clients connectés.
3. Le client demande la liste des services disponibles selon des critères.
4. L'administrateur consulte le répertoire des services.
5. L'administrateur génère une liste des services disponibles selon les critères.
6. L'administrateur retourne la liste des services à l'utilisateur.
7. L'utilisateur choisit un service et envoie une demande de recherche de service à son propre *PCS*.

8, 9. Le *PCS* consulte la base de données pour obtenir la liste de ses participants hébergeant ce service.

Si la liste des participants n'est pas vide:

10. Le PCS retourne cette liste à l'utilisateur.

Sinon :

11. Etant donné qu'aucun *PCS* ne dispose d'informations des services hébergés par les clusters voisins, chaque *PCS* inonde la requête de recherche de service vers ses voisins et incrémente la valeur du TTL de la demande de recherche de service reçue.
12. Chaque voisin (*PCS*) exécute les étapes 8 et 9, si l'ensemble de résultats est vide et que la valeur maximale du TTL n'est pas encore atteinte, il exécute l'étape 11 sinon l'étape 13.
13. La recherche est arrêtée et chaque voisin retourne sa liste des participants. Puis, on passe à l'étape 10.

2.2.4. Invocation de Service

Cette étape (opération 7 et 8, figure 6.2) permet de mettre l'utilisateur et le participant en connexion directe. L'utilisateur initie une demande d'exécution d'un service de son choix en utilisant un module *Demande de Service* «en anglais *ServiceRequest*» qui lui permet d'envoyer une demande de service au participant. Le participant, de son côté, dispose d'un module *Réception d'une Demande de Service* «en anglais *ReceivingServiceRequest*» qui lui permet de recevoir la demande, de notifier le module *Informations sur les Paramètres* «en anglais *ParameterInformation*» qui retourne le fichier *manifest* (définition des paramètres du service demandé) à l'utilisateur via le module *Creation Automatique de l'Interface de Service* « en anglais *AutomaticCreationOfServiceInterface* » qui permet de créer automatiquement une interface au niveau de l'utilisateur pour lui permettre d'introduire les arguments nécessaires au service, et d'invoquer le service à distance en utilisant java RMI (Remote Method Invocation) sur le participant et de recevoir le résultat du traitement sur la même interface. Le participant dispose d'un module *Invocation de Service* « en anglais *ServiceInvocation*» qui lui permet de recevoir les paramètres du service de la part de l'utilisateur, d'exécuter le service et de retourner le résultat à l'utilisateur (figure 6.7). Le schéma présenté dans la figure 6.8 explique la procédure d'invocation:

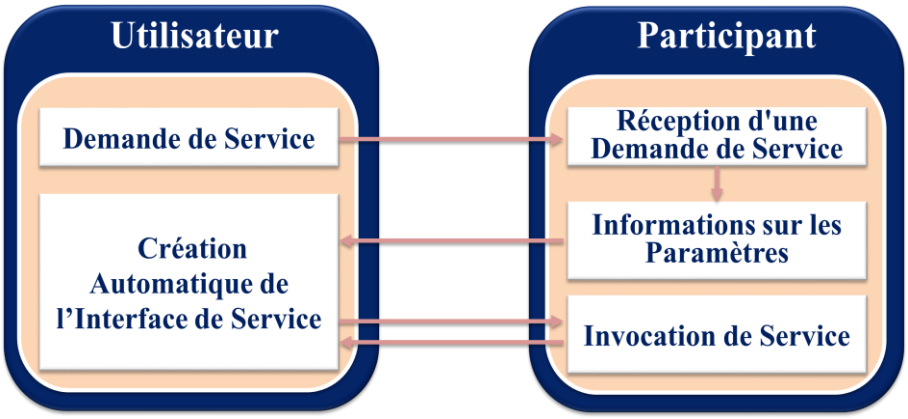


Figure 6. 7: Etape d'Invocation.

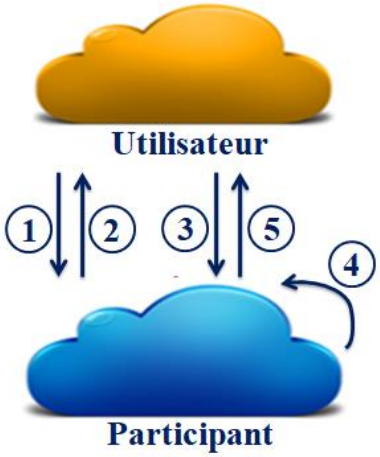


Figure 6. 8: Procédure d'Invocation.

1. L'utilisateur envoie au participant une demande pour un service sélectionné.
2. Le participant renvoie la liste des paramètres demandés à l'utilisateur (le fichier manifeste).
3. L'utilisateur invoque le service avec les paramètres nécessaire sur le participant.
4. Le participant exécute le service avec les arguments reçus de l'utilisateur.
5. Le participant retourne le résultat de l'exécution à l'utilisateur.

3. Evaluation et Analyse des Résultats

Dans cette section, nous présentons l'environnement expérimental et la méthodologie utilisée pour évaluer l'architecture présentée dans la Section 2.

3.1. Environnement Expérimental

Nous avons mis en œuvre un prototype Java du système Cloud-SaaS basé sur un réseau P2P décrit dans la section précédente, en utilisant la technologie Java RMI pour la gestion des communications à distance et le simulateur PeerSim comme cadre expérimental.

PeerSim: Le simulateur PeerSim est un simulateur P2P écrit en Java, partiellement développé dans le cadre du projet BISON et publié sous licence GPL open source. Il a été conçu principalement

pour être évolutif et dynamique afin de simuler des réseaux P2P à grande échelle et soutenir leurs dynamismes. PeerSim est un simulateur générique qui est en mesure de simuler des superpositions structurées et non structurées de grande taille de nœuds, à savoir des millions de nœuds de réseau. Il est adapté également pour des scénarios dynamiques tels que les désabonnements (départs) et les jointures des pairs, ainsi que d'autres types de défaillance. Le simulateur offre à la fois *un modèle de simulation basé sur les cycles* et *un modèle de simulation basé sur des événements*, en utilisant des moteurs de simulation distincts. Dans le *modèle basé sur le cycle*, tous les nœuds sont choisis au hasard et chaque protocole de nœud est invoqué à son tour à chaque cycle. Tandis que dans *le modèle basé sur les événements*, un ensemble de messages ou d'événements est programmé dans le temps et les protocoles des nœuds sont invoqués en fonction de l'ordre temporel de livraison des messages. Le réseau dans PeerSim est modélisé comme une liste de nœuds dont chacun a un identifiant fixe, généré de manière incrémentielle sous forme d'entier par le simulateur ou personnalisé par des mécanismes définis par les utilisateurs et une pile de protocoles définissant son comportement accessible via l'interface du nœud. La simulation dans PeerSim a des initialiseurs et des contrôles qui sont programmés pour être exécutés par le moteur du simulateur de façon périodique accordable par le fichier de configuration dont les initialiseurs sont exécutés avant la simulation, tandis que les contrôles sont exécutés pendant la simulation. Ces composants implémentent l'interface Control et peuvent modifier (ajouter de nouveaux nœuds ou détruire ceux existants) ou superviser les différents nœuds lors de la simulation et collecter des statistiques qui peuvent être formatées et envoyées vers une sortie standard pour tirer des conclusions significatives sur le réseau (Jesi 2005) (Ebrahim et al. 2014) (Naicken et al. 2006) (Srati et al. 2017).

Le prototype proposé a été évalué à l'aide d'un grand nombre de simulations utilisant le modèle basé sur le cycle de PeerSim, dans lequel un fichier de configuration est utilisé dans la première étape de la simulation. Ce fichier consiste en un simple fichier ASCII, composé essentiellement de paires clé-valeur et contenant tous les paramètres de simulation pour tous les objets impliqués dans l'expérience (Jesi 2005). Pour nos expériences, nous avons utilisé des configurations de test en combinant les paramètres présentés dans le tableau 6.1:

Tableau 6. 1: Paramètres de Configuration.

Taille du Réseau	1000, 5000, 10000, 30000, 50000 nœuds
Nombre de *PCS*	30
Nombre maximal des voisins pour chaque *PCS*	5
TTL (Time To Live)	7
Technique de Communication entre les *PCSs*	Inondation (en anglais, Flooding)

Ensuite, le simulateur configure le réseau en initialisant les nœuds du réseau et les protocoles en eux. Les nœuds du réseau et les protocoles sont créés par clonage grâce à la méthode de clonage *clone ()* de la classe *Node*. Autrement dit, une seule instance est construite à l'aide du constructeur de l'objet qui sert de prototype, et tous les nœuds du réseau sont clonés à partir de ce prototype. La phase d'initialisation est effectuée par les objets de contrôle dont l'exécution est planifiée uniquement au début de chaque expérience. Après l'initialisation, le moteur piloté par cycles appelle tous les composants (protocoles et contrôles) une fois par cycle, jusqu'à la fin de la simulation (Jesi 2005). Les résultats de simulation sont effectués sur un PC DELL G5 15 5587 avec un processeur Intel Core i7 de 2,2 GHz ; Turbo 4,1 GHz / 9, 32 Go de mémoire et un disque dur de 1 To + 256 SSD exécutant sur

Windows10. Pour adapter notre solution au simulateur PeerSim, nous avons défini un ensemble de classes qui décrivent le comportement des principaux acteurs de l'expérience:

Les Clients: Ils sont représentés par une classe qui contient les différents modules (méthodes) qui leur permettent d'effectuer les différentes tâches (communication avec leurs *PCSs* et les autres clients, hébergement de services, recherche de services et demande de service). Cette classe implémente l'interface *CDProtocol* qui hérite de la classe Protocol et possède la méthode *nextCycle()* où nous avons défini les différentes tâches qu'un pair doit accomplir durant son déroulement. Dans notre solution, nous avons prévu quatre scénarios possibles avec une probabilité égale par scénario: 1) hébergement de services, 2) recherche de services, 3) suppression de services, 4) le pair ne fait rien. Chaque scénario a une probabilité de 0.25. Cette classe utilise la structure de données suivante:

- ***Liste locale de services hébergés,*** contenant tous les services hébergés spécifiques à chaque client.

Les Serveurs Cloud Partiels: Ils sont représentés par une classe qui possède les modules d'indexation et de recherche et les outils nécessaires pour la communication avec les voisins. Cette classe implémente l'interface Control qui possède *execute()* comme méthode principale pour informer chaque client du réseau et de l'adresse de son *PCS*. Ce Control n'est appelé qu'une seule fois lors de l'étape d'initialisation. Cette classe utilise la structure de données suivante:

- ***Liste partielle des services hébergés,*** contenant des informations sur tous les services hébergés localement par tous les clients du *PCS* ainsi que leurs identifiants.

Les Observateurs: Ils sont représentés par des classes qui nous permettent d'extraire des mesures (statistiques) nécessaires pour analyser et comprendre le comportement de notre réseau. Durant chaque cycle, les mesures sont collectées et analysées à la fin de la simulation. Les expérimentations portent sur cinq indicateurs de performance:

1) *Recherche Réussie (*en anglais *Success Search)*, la recherche d'un service demandé prend une valeur positive si le serveur Cloud partiel trouve au moins un participant qui héberge le service demandé.
2) *Résultat Réussi* (en anglais *Success Result)*, reflète le succès de résultat retourné par les participants après exécution des services demandés.
3) *Nombre de Nœuds Ressources (*en anglais *Number of Resource Nodes)*, c'est le nombre de nœuds hébergeant le service demandé.
4) *Résultat Echoué (*en anglais *Failure Result)*, qui signifie la défaillance des nœuds participant à fournir correctement le service demandé.
5) *Recherche Echouée (*en anglais *Failure Search)*, la recherche d'un service demandé échoue si les serveurs Cloud partiels ne trouvent aucun fournisseur de service.

Le Nœud Central: Pour mieux concrétiser l'environnement du Cloud et son nœud central, nous avons créé une classe de type *Control*, accessible par tous les pairs, qui est lancée une seule fois durant la phase d'initialisation de la simulation et qui simule le nœud central. Cette classe utilise la structure de données suivante:

- ***Liste globale des services,*** contenant les catégories, sous catégories, noms, entrées et sortie des services. Elle sert de référence aux clients pour choisir des services qu'ils vont héberger ou demander.

3.2. Résultat et Discussion

Nous avons réalisé cinq scénarios de simulation en fonction de la taille du réseau P2P (taille : 1000, 5000, 10000, 30000, 30000, 50000), les résultats à discuter portent sur les métriques introduites précédemment.

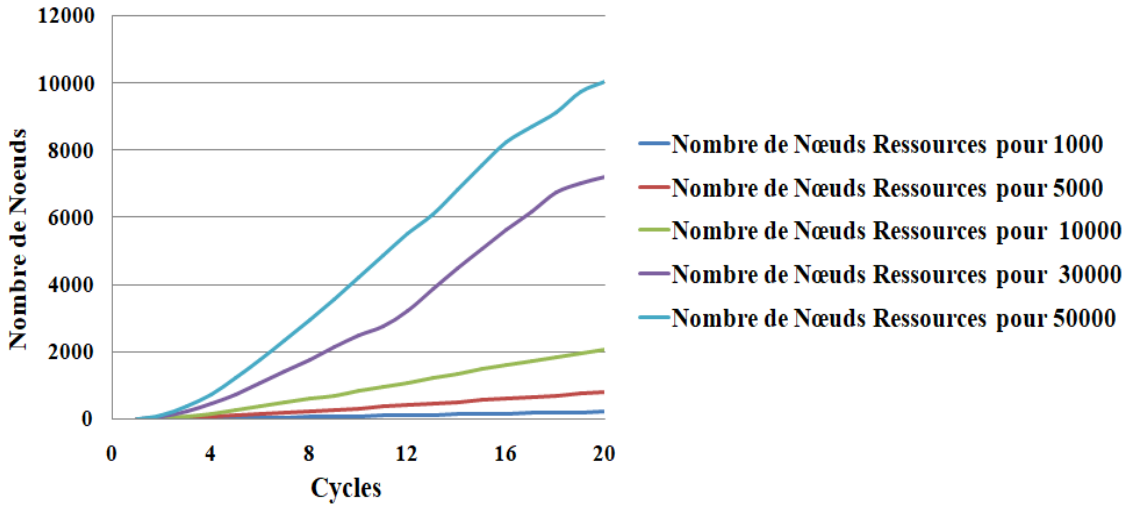


Figure 6. 9: Nombre de Nœuds Ressources.

La figure 6.9 représente la variation du nombre de nœuds ayant les services demandés par cycle pour les 5 différents scénarios de simulation. En observant le graphique de la figure 6.9, nous pouvons constater que le nombre de nœuds disposant les services demandés a augmenté de manière significative au fil du temps. Cela est dû au fait que les clients hébergent activement les services et deviennent des participants au système. Par conséquent, cette croissance aura un impact positif sur le processus de recherche.

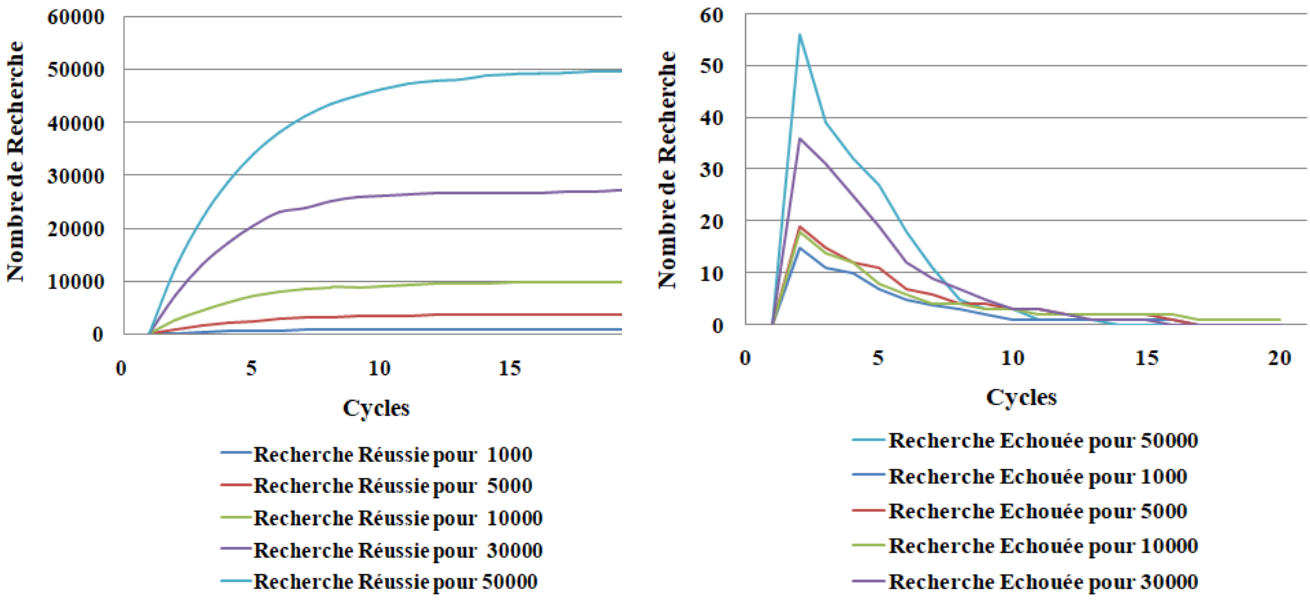


Figure 6. 10: Recherche Réussie Vs. Recherche Echouée.

La figure 6.10 montre la variation des deux métriques *Recherche Réussite* «en anglais *Success Search* » *et Recherche Echouée* «en anglais *Failure Search* » dans les différents scénarios de simulation. D'après les statistiques, nous observons tout d'abord que la *Recherche Réussie* est supérieure à *Recherche Echouée* dans tous les scénarios et pendant toute la durée de la simulation. De plus, les résultats montrent que le facteur de la *Recherche Réussie* augmente constamment au fil du temps, ce qui est une conséquence évidente puisque le nombre de nœuds qui ont la ressource augmente simultanément (figure 6.9). Ce facteur commence à se stabiliser à partir du cycle 15. Tandis que, le facteur *Recherche Echouée* augmente légèrement au début de la simulation car à ce stade il y a moins de participants, puis il diminue à partir du 3ème cycle et tend vers zéro au fil du temps.

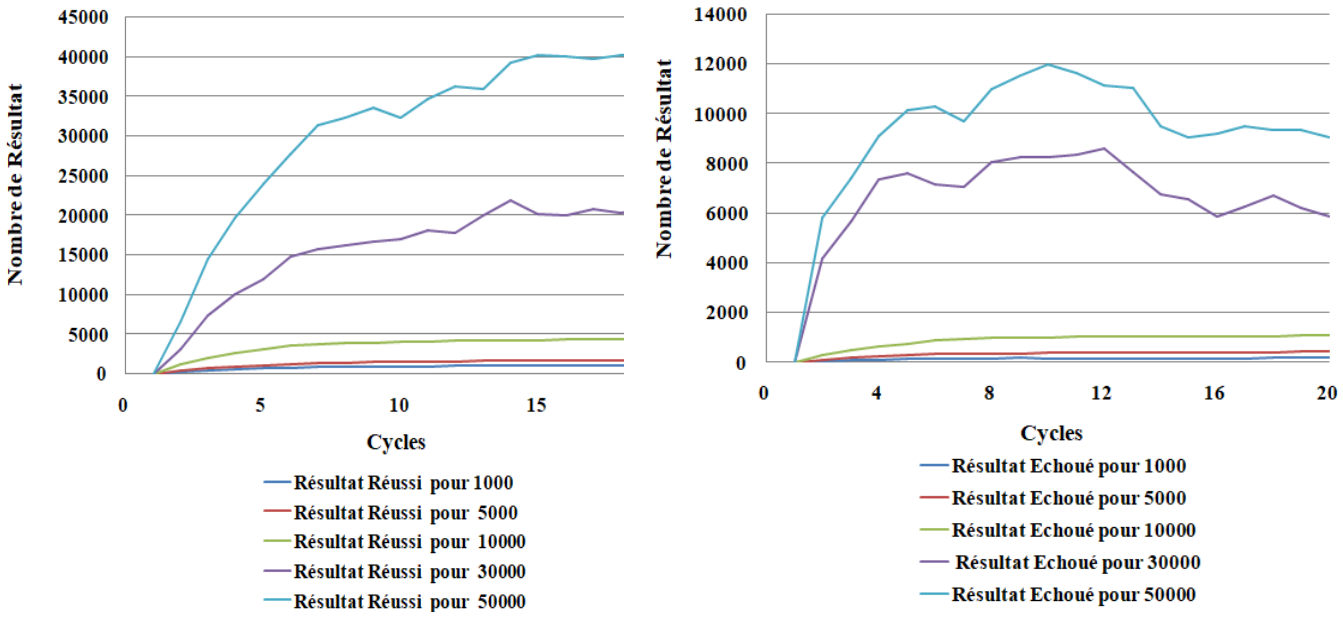


Figure 6. 11: Résultat Réussi Vs. Résultat Echoué.

La figure 6.11 montre la variation des deux métriques *Résultat Réussi* « en anglais *Success Result* » et *Résultat Echoué* «en anglais *Failure Result* » dans les différents scénarios de simulation. Ces facteurs sont les facteurs les plus importants qui reflètent la capacité du système à assurer la fonction d'approvisionnement en ressources à la demande du Cloud Computing. D'après ces statistiques, nous pouvons observer que le *Résultat Réussi* augmente continuellement pour atteindre le maximum dans le 15ème cycle puis stagne jusqu'à la fin de la simulation. Ce changement est associé à une variation du *Résultat Echoué*. Ce dernier s'accroit au fil du temps jusqu'au 9ème cycle, puis diminue progressivement, suite à quoi il se stabilise jusqu'à la fin de la simulation. De plus, nous pouvons clairement voir que les résultats du *Résultat Réussi* restent supérieurs à ceux du *Résultat Echoué.*

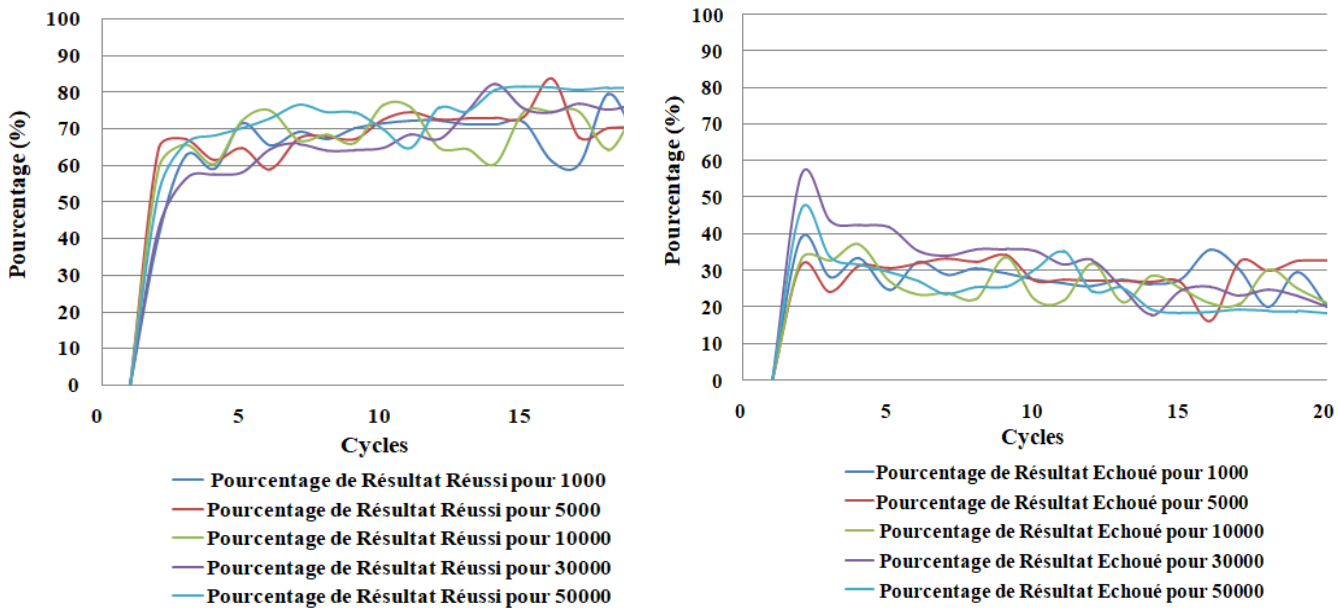


Figure 6. 12: Pourcentage du Résultat Réussi Vs. Pourcentage du Résultat Echoué.

La figure 6.12 représente l'évolution du *Pourcentage du Résultat Réussi* «en anglais *Percentage Success Result* » et du *Pourcentage du Résultat Echoué* «en anglais *Percentage Failure Result* » par rapport aux requêtes effectuées dans les différents scénarios de simulation. Durant les trois premiers cycles, nous observons une augmentation significative du *Pourcentage du Résultat Réussi,* atteignant un taux de 60%. Cette variation est associée à une progression rapide du *Pourcentage du Résultat Echoué* qui atteint 40%. A partir du 3ème cycle, le *Pourcentage du Résultat Réussi* continue à croître lentement atteignant une valeur maximale de 80%, puis se stabilise jusqu'à la fin de la simulation. Tandis que le *Pourcentage du Résultat Echoué* diminue au fil du temps, jusqu'à une stagnation qui tend vers une valeur de 20%.

4. Conclusion

Dans ce travail, nous avons proposé un Cloud qui utilise un réseau Pair-à-Pair hybride comme infrastructure pour fournir (une exécution à distance) des logiciels en tant que service « SaaS ». Des Serveurs Cloud Partiels sont disponibles et sont responsables de l'indexation et de la coordination, permettant aux clients utilisateurs d'effectuer des demandes de services et de se connecter avec des clients participants. Dans le réseau, chaque client peut être un participant ou un utilisateur du système. Parmi les points importants de notre solution figure la distribution du Cloud sur une infrastructure distribuée qui est formée par les clients. Au lieu que le traitement soit fait par le Cloud, il sera délégué à ces clients. Les résultats expérimentaux obtenus ont montré que le système proposé peut atteindre un taux de réussite élevé, allant jusqu'à 80%. Ces résultats sont encourageants et prouvent qu'une architecture Cloud distribuée et dynamique basée sur un réseau Pair-à-Pair pour la fourniture de services pourrait être une solution fiable aux problèmes d'infrastructure.

Chapitre 7: Clustering Hybride pour Améliorer la Disponibilité des Services dans un Cloud-SaaS basé sur P2P

Sommaire

1. Introduction
2. Approche Proposée
   - 2.1. Rappel du Modèle de Système
   - 2.2. Méthodologie
     - 2.2.1. Préparation des Données
       - 2.2.1.1. Structure de Données des Profils des Clients
       - 2.2.1.2. Normalisation des Caractéristiques
     - 2.2.2. Formulation de la Couche Virtuelle
       - 2.2.2.1. Clustering des Données des Profils des Clients
       - 2.2.2.2. Formation de Nœuds Virtuels et de la Couche Virtuelle
     - 2.2.3. Mesure de la Disponibilité
3. Evaluation et Analyse des Résultats
   - 3.1. Montage Expérimental
   - 3.2. Critères d'Evaluation
   - 3.3. Résultat et Discussion
4. Discussion Comparative
5. Conclusion

1. Introduction

Dans ce chapitre, nous étendons le système de base de notre première contribution, en proposant une nouvelle approche destinée à répondre aux exigences de disponibilité du service et à maintenir les performances dans un environnement dynamique « Cloud-SaaS basé sur un réseau Pair-à-Pair hybride pour le traitement des services à distance ». Comme le système proposé est construit sur un réseau Pair-à-Pair, sa performance dépend fortement de sa capacité à maintenir la disponibilité du service, qui est liée à la disponibilité des pairs du réseau. Afin d'obtenir, de leur part, une estimation de la disponibilité des services, nous avons proposé une approche qui améliore la disponibilité des services en surmontant les problèmes de défaillances du système et de dynamique des pairs. L'approche proposée consiste tout d'abord à analyser le comportement individuel des clients et à former des données de profil pour chacun d'entre eux. Ces données de profil seront utilisées pour classer et agréger les clients du système en fonction de leurs similarités de profil à l'aide d'un mécanisme de clustering hybride, qui vise à fournir une infrastructure virtuelle et optimale pour organiser les pairs du système en clusters distincts, où chaque cluster est représenté par un nœud virtuel formant ensemble une couche virtuelle. Comme le choix des techniques de clustering dépend de la nature des données analysées, nous proposons d'améliorer la qualité du clustering des profils des clients en combinant dans «un processus d'hybridation séquentielle » deux algorithmes de clustering «Dur» et «Doux» bien connus, à savoir la «Classification Ascendante Hiérarchique» et l'algorithme «C-moyennes Floues à Seuil». La couche virtuelle résultante permet non seulement la distribution des pairs fournisseurs mais aussi la formation de zones condensées de chaque service qui intéresse un ensemble de pairs voisins, ce qui contribue grandement à améliorer la probabilité de disponibilité de ces services dans des régions spécifiques de la couche virtuelle au moment de la demande. En outre, nous avons proposé un modèle pour mesurer la disponibilité des services, qui se base principalement sur l'utilisation de la couche virtuelle du système et prend en compte différentes entités à différents niveaux du système. Ce chapitre est articulé comme suit: une description approfondie de l'approche proposée pour améliorer la disponibilité des services est présentée dans la deuxième section. Nous décrivons le cadre expérimental et nous présentons les résultats obtenus dans la section 3. Enfin, une étude comparative est discutée dans la section 4. Ce travail a été l'objet de notre article de journal « Hybrid fuzzy clustering to improve services availability in P2P-based SaaS-Cloud, 2022 ».

2. Approche Proposée

Dans cette section, nous donnons d'abord une description récapitulative de notre système de base « Cloud-SaaS basé sur un réseau P2P hybride pour le traitement des services à distance » afin de mieux rappeler le modèle de réseau sur lequel notre solution de disponibilité sera exécutée. Ensuite, nous décrivons la structure de métadonnées utilisée pour représenter les clients du système. Enfin, nous présentons notre approche proposée qui vise à améliorer la disponibilité des services et à maintenir un système Cloud-SaaS basé sur un réseau P2P fiable et performant.

2.1. Rappel du Modèle de Système

Nous avons proposé un système consistant en une nouvelle architecture Cloud. Afin de fournir des services sous forme SaaS aux pairs qui n'en disposent pas, cette architecture utilise le réseau P2P hybride (H-P2P) comme infrastructure. Le dit système permet aux pairs d'utiliser et d'exécuter des services SaaS qui sont hébergés et exécutés par les pairs eux-mêmes et initialement fournis par le Cloud. Ces services sont exécutés par invocation sur les pairs qui les hébergent, et ce, à l'aide de la

technologie Remote Method Invocation (RMI). Cette architecture consiste en un environnement dynamique distribué qui dispose d'une infrastructure minimale couvrant quatre types d'entités, dont: 1) L'*Administrateur*, responsable de l'alimentation du système en services et de leur approvisionnement; 2) Les *Serveurs Cloud Partiels* «*PCSs*» qui sont connectés sur une couche réseau et qui sont responsables de l'indexation des services. Les clients peuvent être: 3) Des *Clients Participants* qui hébergent, fournissent et exécutent à distance des services pour le compte des *Cients Utilisateurs*, 4) ces derniers peuvent exploiter ces services. Ensemble, le *PCS* et les clients, qui sont géographiquement distribués, forment successivement l'ensemble des super pairs et des pairs normaux, constituant ainsi le réseau H-P2P. Par conséquent, les pairs normaux sont regroupés en clusters gérés par des super-pairs formant la couche réseau réelle du système. En outre, les clients exploitent le concept des réseaux P2P pour permettre une connexion directe entre eux lorsqu'un service est invoqué sans l'intervention d'un tiers.

2.2. Méthodologie

L'approche proposée tente de rallier la dynamique et la vélocité des comportements des pairs (utilisateurs/participants) dans la couche réseau réelle du système de prestation de services afin de collecter leurs informations. Les informations relatives aux comportements individuels forment des données de profil représentées sous la forme d'une matrice de vecteurs de profils de clients. Un mécanisme de mise en cluster utilise cette matrice et fournit une infrastructure virtuelle optimale en réorganisant les pairs dans le réseau. En d'autres termes, le mécanisme de mise en cluster proposé effectue un nouveau regroupement des pairs du réseau, outre leur regroupement dans la couche réseau réelle. Ces pairs sont dispatchés, selon les similarités de leurs profils, dans plusieurs clusters formant une *Couche Virtuelle* (en anglais *Virtual Layer*, abrégé *VL*). Chaque cluster est représenté par un *Nœud Virtuel* (en anglais *Virtual Node*, abrégé (VN_{C_j})). Cette couche permet non seulement la distribution des fournisseurs de services, mais aussi la formation de zones condensées de chaque service qui intéressent un ensemble de pairs qui les hébergent ou les utilisent. Pour y parvenir, nous proposons d'améliorer la qualité du clustering des profils de clients en combinant, dans un processus d'hybridation séquentielle, deux algorithmes bien connus des deux familles de clustering « *Dur* » et « *Doux* », à savoir *Classification Ascendante Hierarchique* «en anglais *Hierarchical Ascending Classification*, abrégé *HAC* » et *C-Moyennes Floues à Seuil* «en anglais *Thresholded Fuzzy C-Means*, abrégé *TFCM* ». En effet, l'hybridation proposée répond largement à la nature des données manipulées, permettant d'associer automatiquement chaque vecteur du profil du client au nombre approprié de clusters. Ces derniers sont les plus similaires, sans connaissance à priori du numéro de clusters et du nombre de clusters auxquels il doit appartenir. Dans un premier temps, l'algorithme *HAC* utilise les vecteurs de profils desclients et génère une partition dure qui minimise l'inertie interclasse et maximise l'inertie intraclasse. Ensuite, cette partition dure, avec ses informations associées, est utilisée dans l'étape d'initialisation de l'algorithme *TFCM* au lieu d'une initialisation aléatoire. Dans notre approche, le *TFCM* est un outil de base pour construire une hiérarchie semi-floue à partir de profils des clients et un algorithme complémentaire pour le *HAC* permettant l'affectation des clients à des clusters jugés suffisamment proches. Nous suggérons l'utilisation de la technique de normalisation Z-Score pour prétraiter la variabilité des ensembles de données et assurer ainsi le fonctionnement précis des fonctions objectives et améliorer la performance des résultats du clustering. Par conséquent, la couche virtuelle forme des zones condensées de services hébergés par des pairs appartenant au même cluster, ce qui peut améliorer considérablement la probabilité de disponibilité de ces services dans des régions spécifiques au moment de la demande. Dans notre système de prestation de services, l'évaluation de la disponibilité des services proposée est basée principalement sur l'utilisation de la couche virtuelle pour calculer la probabilité de disponibilité des services, en tenant compte de la

probabilité de disponibilité des nœuds virtuels et donc de celle de disponibilité des fournisseurs de services dans la couche réseau réelle. Afin de mesurer la disponibilité de ces entités, nous utilisons différents types de métriques de disponibilité issus de la littérature: disponibilité basée sur le temps, disponibilité basée sur l'activité, disponibilité basée sur la présence de k-of-n. Notre proposition a été structurée en trois étapes majeures illustrées dans le schéma général de la figure 7.1 et détaillées séparément dans les sous-sections suivantes.

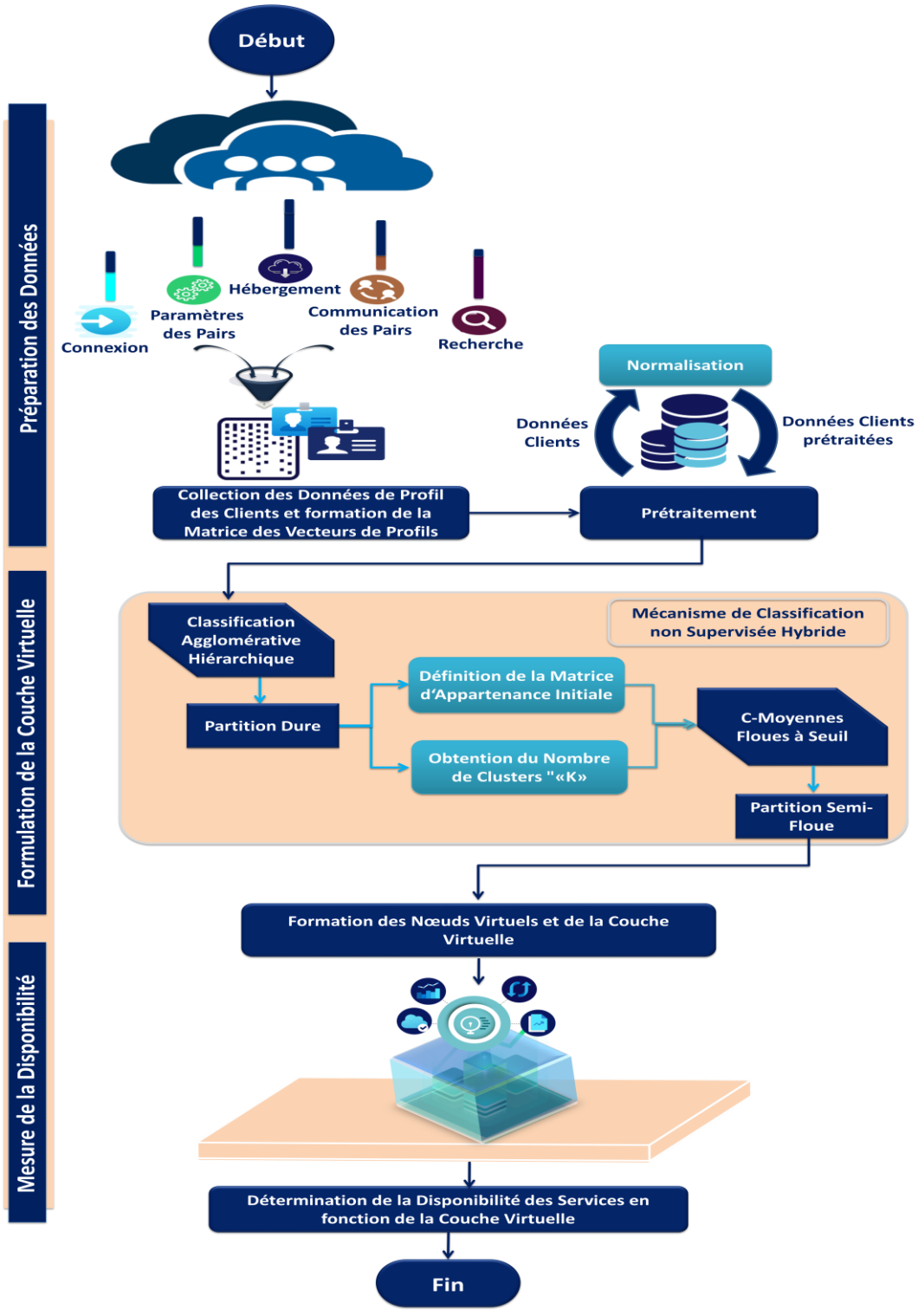


Figure 7. 1: Schéma Global de l'Approche Proposée.

2.2.1. Préparation des Données

2.2.1.1. Structure de Données des Profils des Clients

Les informations des profils des clients dans le système de prestation de services Cloud-SaaS basé sur H-P2P sont conservées dans une matrice 2D. Un profil client est un ensemble d'informations qui détermine son propre comportement dans le réseau. Nous désignons par P l'ensemble des N pairs constituant le système, $P = \{p_1, p_2, \dots, p_n\}$. La matrice des vecteurs de Profils (en anglais *Vector Profil Matrix*, abrégé *VPM*) $VPM(N \times Q)$ est un ensemble de N vecteurs de profil PV_{p_i}et Q variables (comme indiqué sur la figure 7.2).

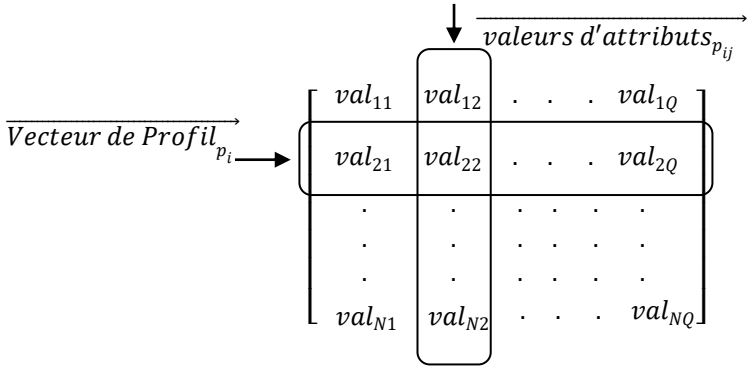


Figure 7. 2: Matrice de Vecteurs des Profils des Clients.

Chaque vecteur de profil PV_{p_i} représente le profil d'un client p_i et consiste en un ensemble de q variables, ou attributs, décrivant son comportement. Dans ce travail, ces attributs sont collectés périodiquement à partir du système fonctionnant dans un environnement entièrement dynamique. Plusieurs fonctionnalités caractérisant le comportement des pairs dans au sein du système Cloud basé sur un réseau P2P hybride pour le traitement des services SaaS à distance sont définies:

- *Services Hébergés :* ce critère fournit une indication de tous les services hébergés par un pair.
- *Intérêt envers les services :* est une matrice de préférence « Client-Service » inspirée du système de recommandation (Chen et al. 2018) qui fournit un vecteur de préférence pour chaque client. Le vecteur de préférence contient des valeurs d'évaluation qui reflètent l'intérêt d'un client pour les services hébergés, demandés ou exécutés.
- *Localisation géographique :* cet attribut décrit l'emplacement réel du pair. En effet, la localisation géographique s'est avérée être un facteur important dans l'optimisation des réseaux en termes de coûts de communication entre pairs.
- *Réputation :* La confiance est un concept complexe qui mesure le degré de fiabilité en termes d'incertitude et de subjectivité qu'un pair peut solliciter un autre pair pour effectuer une tâche donnée pour lui. La confiance, généralement appelée réputation, est considérée comme une mesure globale plus objective, formée sur la base d'informations ou d'observations sur le comportement passé (Koutrouli et Tsalgatidou 2006) (Shree et Basha 2014). Dans le système proposé, la réputation reflète le degré de fiabilité des pairs et est calculée en agrégeant un ensemble de caractéristiques.
  - ✓ *Caractéristiques basées sur le temps :* elles mesurent le comportement temporel des pairs : (a) temps de connexion, (b) durée de connexion et (c) disponibilité temporelle.
  - ✓ *Caractéristiques basées sur l'activité :* elles caractérisent le niveau d'activité des pairs et comprennent : (a) la disponibilité individuelle basée sur l'interaction (activité), (b) le nombre de demandes de service reçues, (c) le nombre de demandes de fourniture de service réussies,

(d) le nombre de demandes de fourniture de service échouées, (e) le nombre de demandes de recherche et enfin (f) la moyenne des services hébergés pendant une session.

Les caractéristiques ci-dessus peuvent varier d'une session à l'autre et durant une même session.

2.2.1.2. Normalisation des caractéristiques

La normalisation des données est définie comme le processus de mise à l'échelle de données brutes sans changer leur nature et vise à générer des clusters de haute qualité et à améliorer les performances (Visalakshi et Kuttiyannan 2009).

Plusieurs attributs des données de profil ont une plage très large, par exemple la valeur du « *nombre de demandes de service reçues* » peut varier dans la plage [0, 150]. Par conséquent, ces caractéristiques sont normalisées en utilisant la méthode de normalisation Z-score, qui est considérée comme l'une des méthodes de normalisation les plus puissantes qui donnera des résultats de Clustering plus précis et plus efficaces. La technique du Z-score consiste à mettre à l'échelle les valeurs d'un attribut *X* en fonction de la moyenne et de l'écart-type afin de s'assurer que la distribution des caractéristiques a une moyenne μ = 0 et un écart-type σ = 1 (voir équation.1).

$$Z(val_{ij}) = \frac{val_{ij} - \mu_j}{\sigma_j} \quad (1)$$

2.2.2. Formulation de la couche virtuelle

2.2.2.1. Clustering des Données des Profils des Clients

Le mécanisme de Clustering de données proposé combine les deux algorithmes *HAC* et *TFCM* afin de surmonter leurs limites et de bénéficier des avantages de chacun. Il commence par l'application de l'algorithme *HAC* sur les données. L'algorithme *HAC* prend en entrée une matrice de profils des clients formée par un ensemble de vecteurs de profil spécifiés à chaque pair du système $VP = \{VP_i(val_1, val_2, \dots, val_Q), VP_i \epsilon R^Q | i = 1, \dots, N\}$, où *N* est le nombre de pairs du réseau représentés par des vecteurs de profil de dimension *Q*. Chaque vecteur VP_i est un ensemble de valeurs d'attributs qui décrivent l'intérêt et le comportement d'un pair. La sortie de l'algorithme *HAC* est un ensemble de clusters C= $\{C_m \subset VPM \mid$ m=1,…,*K* $\mid K \leq N\}$, où *K* est le nombre de clusters dans la sortie et chaque cluster C_m est un ensemble de vecteurs de profil similaires. Afin de connaître le meilleur partitionnement, nous avons ajouté une structure de stockage de données nécessaire pour le processus automatisé de détection du meilleur partitionnement par le dendrogramme. Les clusters créés par l'algorithme *HAC* consistent en une partition dure des vecteurs d'entrée, ce qui signifie: $\forall m, s \in R \; avec \; 1 \leq \underset{m \neq s}{\text{m, s}} \leq K; \; C_m \neq \emptyset; C_s \neq \emptyset; \; C_m \cap C_s = \emptyset$. De plus, l'union de tous les clusters permet de reproduire la matrice de vecteurs de profils des clients, $\coprod_{m=1}^{K} Cm = PVM$. Ces clusters sont formés par les étapes suivantes (figure 7.3) :

Étape 1 : considérer chacun des vecteurs de profil VP_i, comme un cluster unique $\{VP_1\}$, $\{VP_2\}, \dots,$ $\{VP_N\}$. Une structure de stockage de données *S* est définie comme un ensemble de triplets *(r,* $Ward's\ Minimal\ Distance_r$, $Clusters_r$ *)*, où r est le cycle ou l'itération actuelle atteinte par l'algorithme, *Ward's Minimal Distance*$_r$ est la valeur minimale de la perte d'inertie interclasse parmi les valeurs des distances calculées entre tous les vecteurs de profil des pairs, *Cluster*$_r$ représente les clusters résultants durant un cycle (*r*), *i.e. S(0)*←(*0, 0,* $\{\{VP_1\}, \{VP_2\}, \dots, \{VP_N\}\}$). Mesurer la similarité entre chaque paire de clusters en utilisant le critère de Ward avec la distance euclidienne et obtenir la matrice de distance $D = (d_{ig})\ N \times N$.

Étape 2 : former un nouveau cluster VP_{ig} en regroupant VP_i et VP_g ayant la distance minimale. *S(r)* $\leftarrow(r{+}{+}, Ward's MinimalDistance_r, Clusters_r)$.

Étape 3 : mettre à jour la matrice de distance en remplaçant les deux clusters VP_i et VP_g par le nouveau cluster VP_{ig}et en calculant sa distance par rapport aux autres clusters.

Étape 4 : répétez les étapes 2 et 3 jusqu'à ce que tous les objets appartiennent au même cluster.

Étape 5 : récupérer la structure de stockage des données et l'utiliser dans le processus de détection automatique du meilleur partitionnement de l'ensemble de données, qui s'effectue comme suit:

- ✓ Calculer les sauts entre chaque paires successives de partitions P_i et P_{i+1} qui est égal à la différence entre les valeurs minimales de pertes d'inertie inter-classes obtenues dans les niveaux successifs d'agrégation.

  $Hop_{P_i,P_{i+1}} = |\text{ Ward's Minimal Distance }_{P_{i+1}} - \text{Ward's Minimal Distance }_{P_i}|$ (2)
- ✓ Chercher le plus grand saut parmi les sauts calculés entre chaque deux partitions successives: $Max_{Saut_{P_i,P_{i+1}}}$
- ✓ Etant donné que l'on cherche à minimiser la perte d'inertie inter-classes, la partition finale à considérer est celle qui précède le plus grand saut. On récupère la partition $P_i$et on la considère comme la dispersion finale.

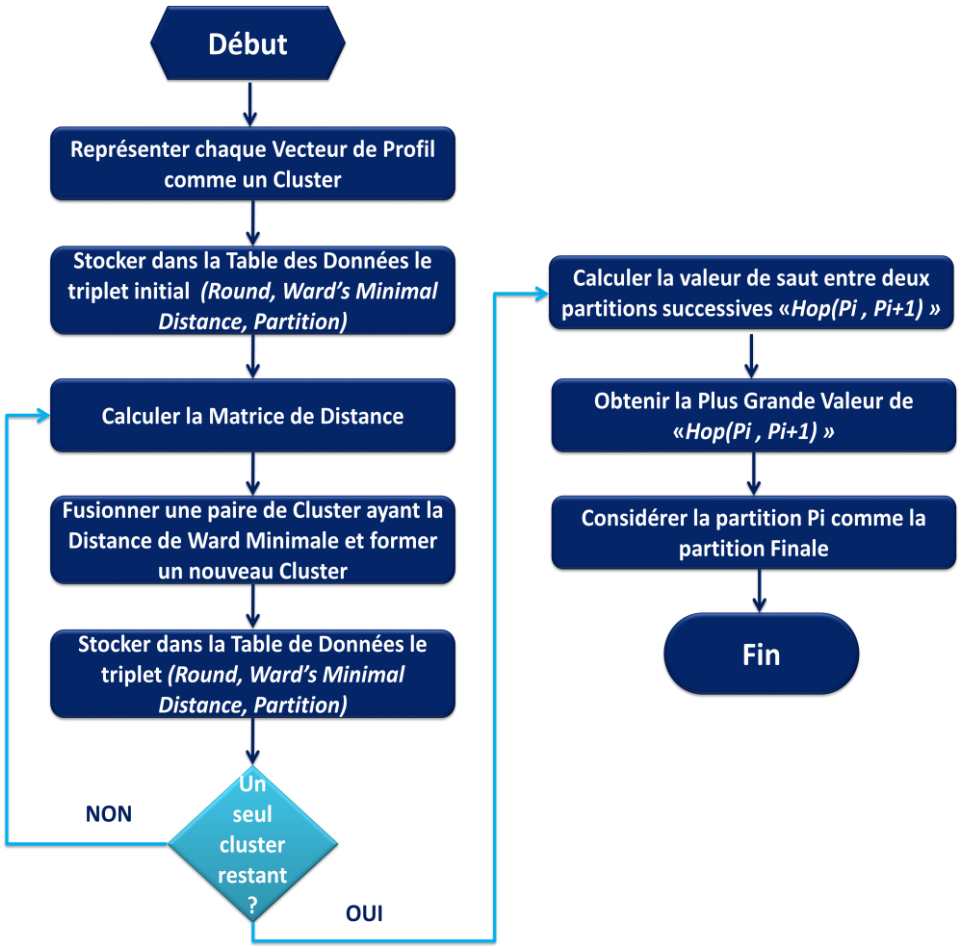


Figure 7. 3: Algorithme de Classification Ascendante Hiérarchique avec le critère de Ward.

Sur la base de cette partition, les informations et les données nécessaires à l'initialisation de l'algorithme *TFCM* peuvent être extraites. Étant donné la matrice de vecteurs des profils des clients

VPM présentée précédemment, P_{HAC} est la partition résultante de l'algorithme *HAC* avec $P_{HAC} = \{C_1, C_2, \dots, C_k\}$. L'algorithme C-Moyennes Floues à Seuil se déroule comme suit (Figure 7.4):

Étape 1 (étape d'initialisation) : en premier lieu, on définit le nombre de clusters *K* en attribuant une valeur égale au nombre de clusters de la partition résultante de l'algorithme *HAC*, $K=|P_{HAC}|$. Ensuite, nous initialisons la matrice d'appartenance $U^{r=0}$ en se basant sur les clusters générés par *HAC* où le vecteur de profil de chaque client VPi se verra attribuer un degré d'appartenance de 1 au cluster auquel il appartient et un 0 pour les autres clusters:

$$u_m(P_i) = u_{im} = \begin{Bmatrix} 1; & si\ P_i \in C_m \\ 0; & sinon \end{Bmatrix} \text{ ; où: } 1 \le i \le N, 1 \le m \le K. \quad (3)$$

Le résultat sera une matrice binaire $K \times N$:

$$U^{r=0} = \begin{pmatrix} u_{ij} & \cdots & u_{iK} \\ \vdots & \ddots & \vdots \\ u_{Nj} & \cdots & u_{NK} \end{pmatrix}$$

Enfin, on attribue arbitrairement une valeur réelle au paramètre de degré de flou "*m*" sachant qu'il doit être supérieur à 1 ($m>1$) et une valeur comprise entre 0 et 1 au critère de terminaison ε ($\varepsilon \in [0, 1]$).

Étape 2 : A chaque itération *r*, le centroïde v_j de chaque cluster C_j est calculé avec U^r en appliquant l'équation suivante:

$$v_j = \frac{\sum_{i=1}^{N} (u_{ij})^m * VP_i}{\sum_{i=1}^{N} (u_{ij})^m} \quad (4)$$

Étape 3 : Mise à jour de la matrice d'appartenance U^r, U^{r+1} en utilisant l'équation suivante :

$$u_{ij} = \left[\sum_{l=1}^{K} \left(\frac{\|VP_i - v_j\|}{\|VP_i - v_l\|} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (5)$$

Étape 4 : Si $\|U^{r+1} - U^r\| \le \varepsilon$, passer à l'étape 5, sinon revenir à l'étape 2.

Étape 5 (Formation de clusters doux ou semi-flou) : dans cette étape, l'algorithme doit être capable de former la partition semi-floue finale où les membres de chaque cluster doivent avoir un degré d'appartenance suffisant. Pour ce faire, il faut procéder aux étapes suivantes:

- ✓ ***Étape de seuillage (****en anglais* ***Threshold Step):*** évaluer le degré d'appartenance u_{ij} de chaque vecteur de profil utilisateur PV_i à un cluster *j* par rapport au seuil τ défini comme le rapport du nombre de clusters:

$$\tau = \frac{1}{K} \text{ , où k est le nombre de clusters.} \quad (6)$$

$$\begin{cases} Si\ u_{ij} \ge \tau; \text{ ajouter le vecteur de profil utilisateur i au cluster j, } tel\ que\ 1 \le i \le n\ et\ 1 \le j \le c\ . \\ ; Sinon. \end{cases} \quad (7)$$

- ✓ ***Étape de Normalisation (****en anglais* ***Normalization Step):*** cette étape permet de calculer les nouveaux degrés d'appartenance $u_{ik}{}^*$ en utilisant l'équation suivante :

$$u_{ij}{}^* = \frac{u_{ij}}{\sum_{j=1}^{k} u_{ij}} \text{; où: } 1 \le i \le n \text{ et } 1 \le j \le k \quad (8)$$

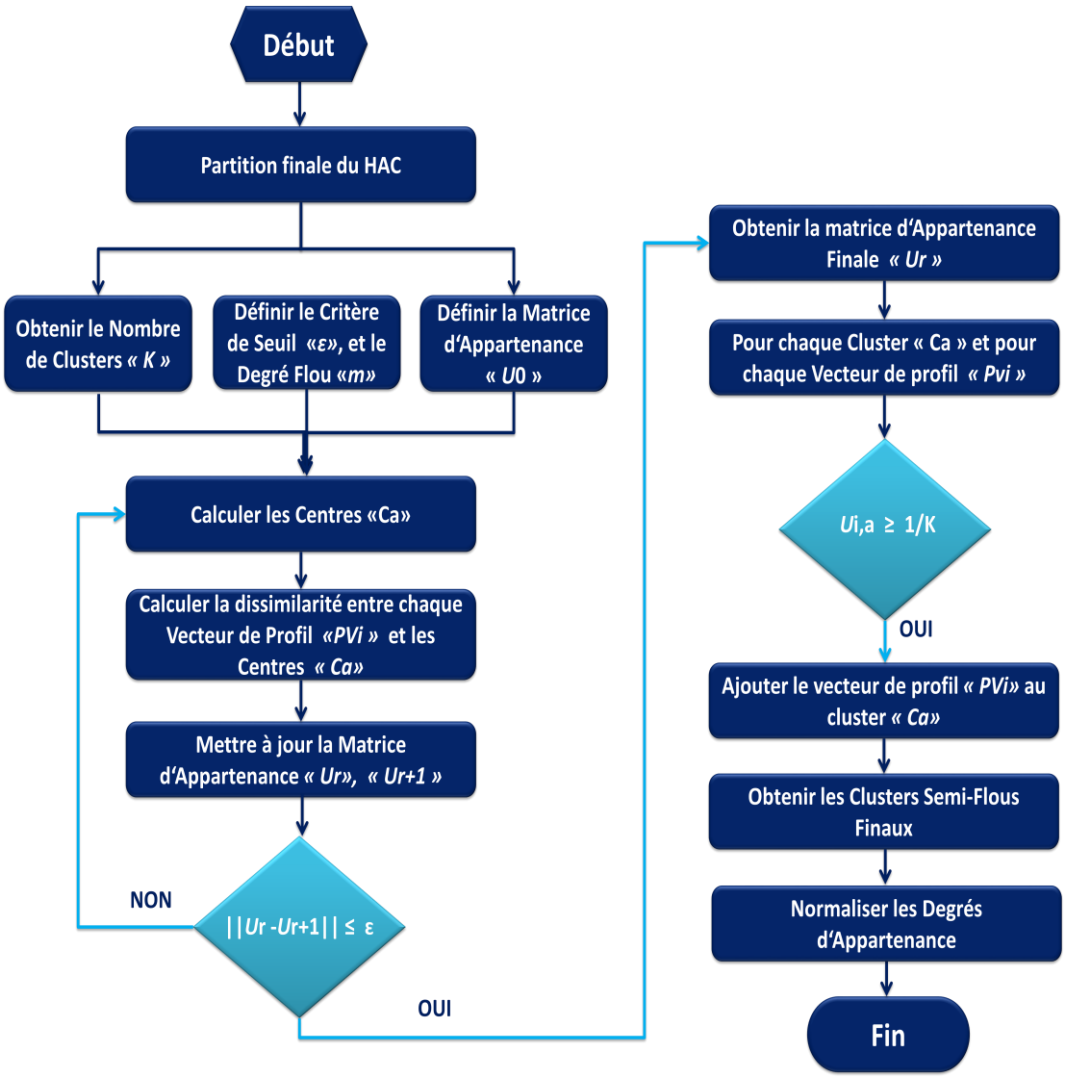


Figure 7. 4: Algorithme C-Moyennes Floues à Seuil Modifié.

Dans la suite de l'algorithme *HAC*, l'algorithme *TFCM* permet tout d'abord d'ajouter un facteur de degré d'appartenance u_{ij} à chaque vecteur de profil client VP_i. Il permet également d'associer chaque vecteur de profil aux clusters auxquels il a un degré d'appartenance suffisant. En utilisant une fonction d'appartenance u_{ij}, le *TFCM* mesure le degré d'appartenance du vecteur de profil client VP_i à chaque cluster $C_m \subseteq C$ et sa valeur varie entre zéro et un $u_{ij} \in [0,1]$. En outre, la somme des degrés d'appartenance pour chaque vecteur de profil du client VP_i doit être égale à un $\sum_{j=1}^{k} u_{ij} = 1$. Le résultat généré par *TFCM* est une partition semi-floue *SFPA* de l'ensemble des vecteurs de profil C= $\{C_m \subset VPM \mid m \in K \leq N\}$, où chaque vecteur peut appartenir à plus d'un cluster parmi les k clusters résultants mais pas nécessairement à tous. L'union de tous les clusters permet de reproduire la matrice de profil des vecteurs, ce qui signifie: $\forall\, m, s \in R \; avec \; 1 \leq {}^{m,s}_{m \neq s} \leq K$; $C_m \neq \emptyset; C_s \neq \emptyset$; $\coprod_{m=1}^{K} Cm = PVM$, tandis que leurs intersections ne donnent pas nécessairement l'ensemble vide. Le pseudo-code du mécanisme de Clustering proposé est présenté dans l'algorithme 1. Ce code illustre l'algorithme de Clustering hybride proposé, incluant *HAC* et *TFCM*.

Algorithm 1 Hybrid Hierarchical Ascending Classification with Modified Thersholded Fuzzy C-Means Algorithm

Input: A set of profil vectors $VP = \{VP_1, \dots, VP_N\}$

Output: $SFPA_{TFCM} = \{C_1, C_2, \dots, C_k\}$, U_{TFCM}

Parameters: C: set of all clusters. **r**: the current iteration, r=0. **K**: number of clusters. $\mathbf{P_r}$: partition at the iteration r.
dist$(\mathbf{c_i}, \mathbf{c_j})$: distance between cluster c_i and c_j obtained based on Euclidean distance and Ward's criterion.
S: data structure to store the triplets (r, Ward's Minimal Distance $_r$, P_r) in each iteration.
D: distance matrix that contain the set of all dist(c_i, c_j).
τ : the threshold parameter. **ε**: the threshold representing the convergence error. **m**: the fuzziness degree.

START

Hierarchical Ascending Classification "HAC"

For i=1 to N
  $c_i \leftarrow \{VP_i\}$
  $C \leftarrow c_i$
endFor
$S(0) = (0, 0, C_0)$
Obtain D
Repeat
  Find *the* smallest element $dist(c_i, c_j)$ in D, Ward's Minimal Distance$_{P_r}$ = MIN$dist(c_i, c_j)$
  Create new cluster c_l by merging cluster c_i and c_j (where, c_i , $c_j \in$ C)
  Remove c_i and c_j from C
  Add c_l to C
  r++
  $P_r \leftarrow$C
  $S_r \leftarrow$ (r, MIN$dist(c_i, c_j)$, P_r)
  Update D by Calculating the distance of c_l from the other clusters
Until C.size = = 1
For r=0 to S.size do
  $Hop_{P_r, P_{r+1}} \leftarrow$ | Ward's Minimal Distance $_{P_{r+1}}$ − Ward's Minimal Distance $_{P_r}$|
endfor
Get the largest hop-loss $Max_{hop_{P_r, P_{r+1}}}$
Retrieve the partition P_r and consider it as the final HAC dispersion, $P_{HAC} \leftarrow P_r \leftarrow \{C_1, C_2, \dots, C_k\}$

Modified Thersholded Fuzzy C-Means "TFCM"

Set K←$|P_{HAC}|$, r←0, $\tau \leftarrow \frac{1}{K}$
Based on P_{HAC} initialize the membership matrix $U^{r=0}$:
For i=1 to N
  For j=1 to K
    if $VP_i \in C_j$ $U_{ij} \leftarrow 1$
    else $U_{ij} \leftarrow 0$
  endFor
endFor
Repeat
  For j=1 to K
    Calculate the centroid v_j of cluster C_j with U^r by applying equation (4)
  endFor
  For i=1 to N
    For j=1 to k
      Update the membership matrix U^r, U^{r+1} using equation (5)
    endFor
  endFor
Until $\|U^{r+1} - U^r\| \leq \varepsilon$
For j=1 to K
  For i=1 to N
    Threshold values of membership degrees:
      if $U_{ij} \geq \tau$ $C_j \leftarrow VP_i$
      else $U_{ij} = 0$
  endFor
  $\boldsymbol{FPA_{TFCM}} \leftarrow C_j$
endFor
For i=1 to N
  Normalize the thresholded membership degree values using formula (7)
endFor
$\boldsymbol{U_{TFCM} = U}$

END

2.2.2.2. Formation des Nœuds Virtuels et de la Couche Virtuelle

L'ensemble des clusters obtenus lors de la partition finale est noté C= $\{C_1, C_2, \dots, C_K\}$, où chaque cluster C_m est constitué de X vecteurs de profils similaires. Après la formation de ces clusters, un nœud virtuel VN_{C_m}sera formé au niveau de chaque cluster C_m et considéré comme son représentant. En d'autres termes, le nœud viruel est une entité abstraite constituée d'un vecteur représentant un cluster spécifique. Chaque nœud virtuel est généré à partir d'un processus d'agrégation qui s'effectue en faisant la moyenne des valeurs des vecteurs de profil X, $(X \leq P)$ formant le cluster. Le résultat de cette agrégation sera un ensemble de K vecteurs où chacun d'eux représente un nœud virtuel, et formé selon l'équation suivante :

$$VN_C = \left\{ VN_{C_m}(Val_{VN_{C_{m1}}}, Val_{VN_{C_{m2}}}, \dots Val_{VN_{C_{mq}}}); 1 \leq m \leq K;\ Val_{VN_{C_{mj}}} = \frac{\sum_{l=1}^{X}(PV_l)_j}{X_{C_m}} ; 1 \leq j \leq Q \right\} \quad (9)$$

Par conséquent, la partition semi-floue du client et les nœuds virtuels résultants forment ensemble une couche virtuelle qui sera ajoutée à l'architecture de notre système proposé « Cloud-SaaS basé sur un réseau P2P Hydride pour le Traitement des Services à Distance ».

Comme mentionné ci-dessus, la couche virtuelle permet la formation de zones condensées de chaque service, ce qui augmente la probabilité de sa disponibilité au moment de sa demande dans une région spécifique. De plus, comme la localisation géographique des pairs est l'une des caractéristiques prises en compte lors du processus de Clustering, les pairs situés dans des zones géographiques et des fuseaux horaires différents et similaires ont tendance à être regroupés dans le même cluster. Cela aura un impact positif sur la disponibilité du service et les performances du système. D'une part, la proximité géographique entre pairs voisins augmente la probabilité de trouver des nœuds proches qui répondent aux demandes de service. D'autre part, la diversité géographique des pairs voisins est avantageuse en termes de disponibilité, car elle permet aux services d'être hébergés et exécutés par des pairs situés dans des lieux géographiques différents et d'être présents dans plusieurs zones physiques.

Lorsqu'un client du système effectue une recherche de service, la recherche est initialement effectuée dans la couche virtuelle et non pas en utilisant tous les pairs du système. Par conséquent, le client demandeur ne contacte que les fournisseurs qui font partie de son cluster virtuel. Toutefois, si cette recherche échoue, elle sera lancée dans la couche " Serveurs Cloud Partiel". Le client demandeur envoie une demande de recherche de service à son propre Serveur Cloud Partiel afin de localiser le service et de chercher un fournisseur. Cela permet d'augmenter la probabilité de disponibilité des services requis, de réduire l'espace de recherche et d'améliorer le temps de réponse.

2.2.3. Mesure de la Disponibilité

Le modèle de mesure de la disponibilité proposé fait référence au calcul et l'évaluation efficaces de la disponibilité dans un réseau dynamique. L'idée de base de notre estimation de la disponibilité des services est principalement basée sur l'utilisation de la couche virtuelle du système. En d'autres termes, l'évaluation de la disponibilité des services repose sur le calcul de la disponibilité de certaines entités, notamment la disponibilité des pairs du réseau ainsi que la disponibilité des nœuds virtuels. Pour obtenir les meilleures performances dans un système de prestation de services, différents types de disponibilité sont importants à différents niveaux du système. Ces disponibilités conservent le noyau de base de leurs définitions traditionnelles, de sorte qu'elles peuvent sembler similaires à celles typiquement utilisées dans les réseaux P2P mais sont affinées pour inclure des considérations spéciales dans leurs calculs en raison de leur utilisation particulière:

Disponibilité des fournisseurs de services : dans la mesure où les pairs fournisseurs constituent l'unité qui effectue la tâche d'hébergement et de fourniture de services aux demandeurs de services,

l'augmentation de leur probabilité de disponibilité est importante pour le système de prestation de services.

La *Probabilité de Disponibilité Temporelle* (en anglais *Temporal Availability Probability, abrégé TAP)* d'un fournisseur de services $TAP_{p(i)}$ dans un système de services, en utilisant des formules de disponibilité basées sur le temps, est simplement définie comme :

$$TAP_{p(i)} = \frac{Up\ Time}{Total\ Observation\ Time} = \frac{\sum_{z=1}^{S} t_p(i)}{T} \quad (10)$$

Où $t_p(i)$ est la durée de la session du pair *P*, *S* est le nombre de sessions dans lesquelles *P* est connecté, *T* est le temps de fonctionnement global du système.
La *Probabilité de Disponibilité basée sur l'Activité* (en anglais *Activity-based Availability Probability*, abrégé *AAP*) d'un fournisseur de services $AAP_{p(i)}$ pour un intervalle de temps dans un système de services reflète le taux de sa capacité à fournir correctement le service. Elle est déterminée sur la base d'un rapport entre son succès à fournir les services demandés et le nombre total de demandes de service reçues de tous les pairs demandeurs. Elle se calcule comme suit :

$$AAP_{p(i)} = \frac{Total\ Success}{Total\ Request\ Number} = \frac{\sum_{r=1}^{R} S(r)}{R} = 1 - \frac{\sum_{r=1}^{R} F(r)}{R} \quad (11)$$

Où, *R* est le nombre de demandes que le pair fournisseur $p(i)$ a reçu pendant un intervalle de temps spécifique, *S(r)* est le succès du fournisseur à fournir correctement le service demandé *(r)*, *F(r)* est l'échec du fournisseur à fournir correctement le service demandé *(r)*.

Disponibilité des nœuds virtuels: Comme mentionné précédemment, l'objectif principal de la couche virtuelle est de maintenir la disponibilité des services dans le système de prestation de services CC-P2P. Cette disponibilité repose principalement sur les nœuds virtuels. La probabilité de disponibilité des nœuds virtuels dans un intervalle de temps donné est obtenue sur la base de la métrique de disponibilité *k-of-n*. Un nœud virtuel VN_mest dit disponible si un sous-ensemble de *k* pairs de l'ensemble de *n* pairs formant le cluster *m* a une probabilité de disponibilité suffisante. Habituellement, cette mesure se base sur une fonction linéaire qui prend deux valeurs possibles, soit 1 lorsque tous les *k* pairs sont connectés durant l'intervalle de temps de mesure, sinon 0. En conséquence, le résultat est une valeur simple soit l'objet en mesure est disponible ou non. Cependant, notre mesure de disponibilité k-of-n nous permet de calculer le degré de probabilité de disponibilité d'un objet et d'un ensemble en adoptant le même concept. En d'autres termes, la probabilité de disponibilité d'un nœud virtuel dépend du degré de probabilité de disponibilité des k pairs parmi les n pairs qui constituent son cluster virtuel. La méthode commence par trier l'ensemble des pairs constituant le cluster m par ordre décroissant en fonction de leurs probabilités de disponibilité, où le pair ayant la valeur de probabilité de disponibilité la plus élevée (/la plus faible) sera situé en haut de la liste formée. Ensuite, les k premiers (/les k derniers) pairs ayant les valeurs de disponibilité les plus grandes (/les plus petites) seront sélectionnés et pris en compte pour calculer la probabilité de disponibilité maximale (/minimale) du nœud virtuel représentant le cluster m. La probabilité de disponibilité d'un noeud virtuel est donnée par la moyenne arithmétique des probabilités de disponibilité des k pairs sélectionnés, c'est-à-dire leurs probabilités de disponibilité basées sur le temps et sur l'activité. Elle est définie comme suit :

$$AP_{VN_m} = \frac{\sum_{i=1}^{k} AP_{p_i}}{k} \quad (12)$$

$$TAP_{VN_m}^{k-of-n} = \frac{\sum_{i=1}^{k} TAP_{p(i)}}{k} \quad (13)$$

$$AAP_{VN_m}^{k-of-n} = \frac{\sum_{i=1}^{k} AAP_{p(i)}}{k} \quad (14)$$

Où $TAP_{VN_m}^{k-of-n}$ *et* $TAP_{VN_m}^{k-of-n}$ reflètent respectivement les probabilités de disponibilité basée sur le temps et sur l'activité d'un nœud virtuel *m* et ses valeurs sont comprises entre les valeurs minimale et maximale des probabilités de disponibilité basée sur le temps et sur l'activité des *k* pairs; $TAP_{VN_m}^{k-of-n} \epsilon \left[Min_{TAP_{p(i)}}, Max_{TAP_{p(i)}}\right]$, $AAP_{VN_m}^{k-of-n} \epsilon \left[Min_{AAP_{p(i)}}, Max_{AAP_{p(i)}}\right]$. En se basant sur la valeur obtenue de cette métrique, nous pouvons indiquer si le nœud virtuel est disponible ou non. Pour ce faire, on définit une fonction linéaire $O_{VN(m)}(t)$ qui prend une valeur de 1 si la probabilité de disponibilité du nœud virtuel est supérieure ou égale à un seuil préfixé, sinon elle prend une valeur de 0. Cette fonction est formulée comme suit:

$$O_{VN_m}(T) = \begin{cases} Si\ TAP_{VN_m}^{k-of-n} \geq \beta\ ou\ TAP_{VN_m}^{k-of-n} \geq \beta;\ Alors\ 1. \\ Sinon;\ 0. \end{cases} \quad (15)$$

Disponibilité du Système de Fourniture de Services: la disponibilité d'un système pendant une période de mesure reflète sa capacité à garantir son principal objectif, qui est le maintien d'un taux adéquat de fourniture des services demandés sous toutes les contraintes. Cela montre clairement que la disponibilité des services et des pairs du réseau est importante pour nous lorsque nous voulons connaître l'état général du système à un moment donné. Nous l'identifions dans un réseau de fourniture de services comme la *disponibilité du système de fourniture de services* (en anglais *service provision system availability*). Dans une approche basée sur le temps, la disponibilité définit la probabilité que le système de fourniture de services soit dans un état actif durant un intervalle de temps donné qui est influencé par la disponibilité temporelle des pairs constituant le système. La *probabilité de disponibilité basée sur le temps du système de fourniture de services (Time-based Availability Probability of the Service Provision System*) peut être exprimée comme suit:

$$TAP_{SPS} = \frac{1}{N}\sum_{i=1}^{N} TAP_{p(i)} \quad (16)$$

Dans le cas de l'approche basée sur l'activité, la disponibilité détermine la probabilité que le système de fourniture de service soit fonctionnel durant une période de temps précise. Cela se traduit par la capacité moyenne des pairs à être fonctionnels en termes de prestation effective des services demandés pendant la même période. Le calcul de la *probabilité de disponibilité du système de fourniture de service basée sur l'activité* (*Activity-based Availability Probability of the Service Provision System*) est similaire au précédent.

$$AAP_{SPS} = \frac{1}{N}\sum_{i=1}^{N} AAP_{p(i)} \quad (17)$$

Dans le cas de la disponibilité basée sur la présence, la probabilité de disponibilité du système de fourniture de services serait égale à la probabilité de disponibilité moyenne des nœuds virtuels représentant les différents clusters, qui est également égale à la probabilité de disponibilité moyenne

des différents clusters dans le réseau à *n* pairs, où *k* d'entre eux sont suffisants pour une fonctionnalité complète :

$$PAP_{SPS}^{k-of-n} = \frac{1}{N}\sum_{i=1}^{N} PAP_{VN_{(i)}}^{k-of-n} \quad (18)$$

Disponibilité du service (Service Availability): Dans le système de prestation de services CC-P2P, la disponibilité du service signifie que le service doit être utilisable par les clients lorsqu'ils en ont besoin et pendant le temps nécessaire pour fournir le service efficacement, et qu'il doit être satisfaisant pour les pairs demandeurs. Cela nécessite la disponibilité d'au moins un pair fournisseur qui héberge le service requis au moment de la demande. La mesure de la probabilité de disponibilité du service pendant un intervalle de temps donné est basée sur la couche virtuelle du système où un ensemble de pairs est dispersé sur un certain nombre de clusters représentés par des nœuds virtuels. Elle est exprimée comme le rapport entre la somme des degrés de probabilité de disponibilité de chaque nœud virtuel pondérée par la valeur de disponibilité de chacun des pairs possédant le service parmi les pairs qui le composent et la somme des degrés de probabilité de disponibilité de chaque nœud virtuel pondérée par la valeur maximale de probabilité de disponibilité que peut atteindre chacun de ses pairs possédant le service qui prend logiquement la valeur de 1. La *probabilité de disponibilité du service* (*Service Availability Probability)* est obtenue comme suit :

$$SAP_s = \frac{\sum_{m=1}^{K}\sum_{i=1}^{n} AP_{Provider_{S_{im}}} * AP_{VN_m}}{\sum_{m=1}^{K} Max\_AP_{Provider_{S_{im}}} * AP_{VN_m}} \quad (19)$$

$$TSAP_s = \frac{\sum_{m=1}^{K}\sum_{i=1}^{n} TAP_{Provider_{S_{im}}} * TAP_{VN_m}}{\sum_{m=1}^{K} Max\_TAP_{Provider_{S_{im}}} * TAP_{VN_m}} \quad (20)$$

$$ASPA_s = \frac{\sum_{m=1}^{K}\sum_{i=1}^{n} AAP_{Provider_{S_{im}}} * AAP_{VN_m}}{\sum_{m=1}^{K} Max\_AAP_{Provider_{S_{im}}} * AAP_{VN_m}} \quad (21)$$

Où $TAP_{Provider_{S_{im}}} et\ AAP_{Provider_{S_{im}}}$ sont les probabilités de disponibilité basées sur le temps et basées sur l'activité du pair fournisseur *i* du service *S* qui appartient au cluster *m*.

$TAP_{VN_m}\ et\ AAP_{VN_m}$ sont la probabilité de disponibilité basée sur le temps et basée sur l'activité du nœud virtuel *m*.

$Max\_TAP_{Provider_{S_{im}}} et\ Max\_AAP_{Provider_{S_{im}}}$ sont les valeurs maximales de la probabilité de disponibilité basée sur le temps et sur l'activité qu'un pair peut atteindre et qui est logiquement égale à 1.

Selon cette formule, la probabilité de disponibilité d'un service est augmentée parallèlement au fait que le service est hébergé par de nombreux pairs et que ces pairs ont une probabilité de disponibilité élevée et appartiennent à plusieurs clusters dont les nœuds virtuels ont une probabilité de disponibilité élevée.

La disponibilité du service dans notre approche est conditionnée par la disponibilité d'autres entités dans un cycle fermé comme le montre la figure 7.5. En d'autres termes, un service est disponible dès que le système de provision de services sur lequel il s'exécute est disponible. En outre, un système est disponible s'il y a au moins un pair disponible appartenant à un ou plusieurs clusters représentés par des Nœuds Virtuels en plus des liaisons de communication. De plus, un Nœud Virtuel est disponible s'il y a au moins *k* pairs disponibles parmi ses *n* pairs. Un pair est considéré disponible s'il est connecté

au système et s'il est capable de remplir efficacement une tâche allouée, c'est-à-dire de fournir correctement les services demandés. Par conséquent, un service est disponible lorsqu'un nœud fournisseur est disponible et accessible au moment de la demande de service qu'il doit la traiter efficacement.

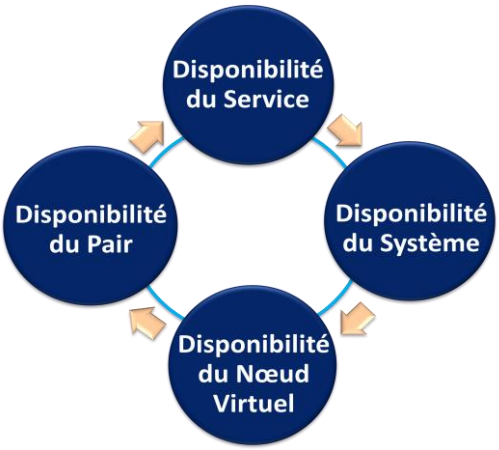


Figure 7. 5: Cycle de Disponibilité.

3. Evaluation et Analyse des Résultats

Cette section commence par une brève description de l'environnement expérimental et de la méthodologie que nous avons utilisée. Ensuite, nous analysons les résultats de simulation obtenus afin de mesurer l'impact de l'approche proposée sur la disponibilité du service et la performance du système.

3.1. Montage Expérimental

Nous avons implémenté un prototype du système Cloud-SaaS basé sur un réseau P2P hybride pour le traitement à distance qui s'appuie sur un mécanisme de Clustering semi-flou hybride des profils des clients afin d'améliorer la disponibilité des services qui les détiennent. Le prototype conserve le cœur de l'implémentation réalisée dans notre première contribution (prséntée dans le chapitre 6) tout en prenant en compte d'autres considérations. Afin d'étudier la performance et l'impact de l'approche proposée sur la disponibilité des services, nos expériences sont mises en œuvre selon trois scénarios différents, à savoir :

Scénario 1 : dans lequel nous avons mis en œuvre « un système Cloud-SaaS basé sur P2P hybride pour le traitement des services à distance » sans utiliser de mécanisme de Clustering, permettant l'évaluation des performances de l'approche proposée et la formation de données de cache pour être utiliser pendant l'initialisation du réseau dans un scénario différent.

Scénario 2 : dans lequel nous avons mis en œuvre « un système Cloud-SaaS basé sur P2P hybride pour le traitement des services à distance » qui utilise différents algorithmes de Clustering (*FCM*, *HAC*) appliqués séparément sur le réseau superposition.

Scénario 3 : dans lequel nous avons mis en œuvre « un système Cloud-SaaS basé sur P2P hybride pour le traitement des services à distance » qui utilise l'algorithme de Clustering hybride proposé (*HAC-TFCM*) comme mécanisme de mise en cluster.

L'approche proposée a été implémentée et comparée au système de base proposé dans notre première contribution afin d'étudier le facteur de disponibilité du service dans les deux architectures. De plus, la technique de Clustering proposée a été évaluée par rapport à différents algorithmes afin de mesurer sa qualité et son impact sur la disponibilité du service et les performances du système. Le simulateur PeerSim (Jesi 2005) (Srati et al. 2017) est utilisé comme cadre expérimental et les résultats sont étudiés en utilisant les métriques décrites dans la section suivante.

Tableau 7. 1: Paramètres de Configuration.

Taille Réseau	10000, 20000, 30000, 40000, 50000 nœuds
Nombre de PCSs	30
Le nombre maximal de voisins pour chaque PCS	5
TTL (Time To Live)	7
Technique de Communication entre les PCSs	Inondation (en anglais, Flooding)
Nombre de pairs à ajouter ou supprimer du réseau	Varie entre 10% et 20% de la taille du réseau. Pour la suppression, les pairs sont choisis au hasard ou en fonction de leur réputation.

Un grand nombre de simulations dynamiques utilisant le modèle PeerSim basé sur les cycles sont réalisées pour évaluer notre prototype. La simulation commence par la lecture du fichier de configuration constitué d'un simple fichier texte ASCII, essentiellement composé de paires clé-valeur et contient tous les paramètres de simulation concernant tous les objets impliqués dans l'expérience, y compris la définition des variables du réseau, la déclaration des structures de nœuds, les protocoles, les Dynamics et les observateurs utilisés (Jesi 2005) (Srati et al. 2017). Pour plus de clarté, nos expériences ont utilisé des configurations de test variables en combinant les paramètres présentés dans le tableau 7.1. Il convient de noter que tous les résultats de simulation sont effectués sur un PC DELL G5 15 5587 avec un processeur Intel Core i7 de 2,2 GHz ; Turbo 4,1 GHz / 9, 32 Go de mémoire et un disque dur de 1 To + 256 SSD. De plus, nous avons implémenté un ensemble de classes qui décrivent le comportement des principaux acteurs dans les différentes couches du système :

L'administrateur : représenté par une classe *Control* qui simule son comportement dans la couche Cloud, accessible par tous les pairs et elle utilise une ***Liste globale des services « Global Service List »*** comme structure de données. Cette dernière contient toutes les informations relatives aux services du système (notamment : leurs noms, catégories, entrées et sorties) et est utilisée par les pairs comme référence pour sélectionner les services à héberger ou à demander. En plus des tâches attribuées à l'administrateur dans le système de base proposé dans la première contribution, y compris l'alimentation du système en services ainsi que leur gestion, ce nœud sera responsable de la planification du processus de Clustering dans le système mettant en œuvre l'approche proposée. Ceci est fait en spécifiant les cycles ou les moments dans lesquels la tâche de Clustering des nœuds est effectuée.

Les Serveurs Cloud Partiels : ils sont représentés par une classe qui possède les modules d'indexation, de recherche et les outils nécessaires pour la communication avec leurs voisins de la couche « Serveurs Cloud Partiel ». Cette classe implémente l'interface *Control* qui possède *execute()* comme une méthode principale permettant d'informer chaque client de l`adresse de son *PCS*. Elle utilise la ***Liste Partielle des Services Hébergés*** « ***Partial list of hosted services*** » comme

structure de données permettant à chaque serveur Cloud partiel de garder les informations relatives aux services hébergés localement par ses pairs.

Les Clients : ils sont représentés par une classe contenant les modules nécessaires pour effectuer diverses tâches, notamment: la communication avec leurs *PCSs* et leurs voisins virtuels, l'hébergement, la recherche, la demande des services et l'envoi des résultats. Cette classe implémente l'interface *CDProtocol* qui hérite de la classe *Protocol* et dispose de la méthode *nextCycle ()* au sein de laquelle sont implémentées les différentes tâches qu'un pair doit effectuer durant son fonctionnement, notamment 1) l'hébergement de services, 2) la recherche de services, 3) la suppression de services et 4) le pair ne fait rien. Ces scénarios sont fournis par le système avec une probabilité égale à 0,25 pour chacun d'eux. Cette classe utilise les structures de données suivantes :

- ***Liste locale des services hébergés*** qui contient l'ensemble des services hébergés propres à chaque pair.
- ***Liste partielle des voisins virtuels*** qui contient l'ensemble des nœuds voisins dans la couche virtuelle. Cette liste sera utilisée en premier lieu comme base de recherche d'un fournisseur pour un service requis et dans le cas où la recherche échoue, le pair se bascule vers la recherche via l'intermédiaire de son Serveur Cloud Partiel.

Sur la base du fichier de configuration, le moteur de simulation piloté par le cycle est chargé afin de configurer le réseau en initialisant les nœuds et les protocoles du réseau comme des instances de classes implémentant une ou plusieurs interfaces. Ils sont créés par clonage en utilisant la méthode « *clone ()* » de la classe « *Node* » permettant de créer une seule instance à l'aide du constructeur de l'objet, qui sert de prototype à partir lequel tous les nœuds du réseau sont clonés. La phase d'initialisation est réalisée par des objets de contrôle dont l'exécution n'est programmée qu'au début de chaque expérience. Après l'initialisation, des composants (protocoles et commandes) sont appelés par le moteur piloté par cycle une fois par cycle. Cependant, d'autres composants sont configurés pour qu'ils ne s'exécutent que dans certains cycles jusqu'à la fin de la simulation (Jesi 2005) (Srati et al. 2017). À la fin, les données statistiques collectées par les composants observateurs pendant la simulation sont formatées et envoyées vers une sortie standard pour des tâches spécifiques, à savoir l'analyse et le sauvegarde. Nos composants observateurs héritent de la classe "*Observer*" de PeerSim et implémentent une méthode *Analyze()* modifiée. Cette dernière a été redéfinie pour surveiller et collecter les statistiques voulues sur tous les entités du système.

3.2. Critères d'Evaluation

Afin d'évaluer les performances de notre système de fourniture des services « Cloud-SaaS basé sur un réseau P2P hybrid pour le traitement des services à distance » ainsi l'impact de l'approche proposée sur la disponibilité des services, nous avons utilisé différents critères d'évaluation, communément utilisés pour évaluer la validité du mécanisme hybride du Clustering semi-flou sur lequel notre approche proposée est basée, ainsi d'autres critères pour mesurer le degré de la disponibilité et la qualité des services. Ces critères se présentent comme suit:

3.2.1. Critères d'évaluation de Clustering

Étant donné que la structure réelle des données expérimentales d'entrée est inconnue et que les étiquettes des classes de données et les informations externes ne sont pas disponibles, la validation de l'approche de Clustering semi-flou nécessite l'utilisation de mesures internes qui reposent uniquement sur les informations contenues dans le jeu des données analysées. Les mesures internes présentées dans le « chapitre 5 sous-section 4.2. » sont utilisées pour estimer la qualité des résultats du

mécanisme de Clustering hybride semi-flou par rapport aux techniques traditionnelles de Clustering dur et pour déterminer son impact sur la disponibilité du service et sur les performances du système.

3.2.2. Critères d'Evaluation de la Disponibilité des Services et des Performances du Système

Durant chaque cycle, des mesures sont captées et analysées à la fin de la simulation pour évaluer les performances du système proposé en termes de disponibilité des services et répondre à la qualité du service requise par les utilisateurs. Nous avons focalisé notre attention sur les indicateurs de performance suivants:

- *Recherche Réussie (*en anglais *Success Search)*, la recherche d'un service demandé prend une valeur positive si le Serveur Cloud Partiel trouve au moins un participant qui héberge le service demandé.
- *Recherche Echouée (*en anglais *Failure Search)*, la recherche d'un service demandé échoue si les Serveurs Cloud Partiels ne trouvent aucun de fournisseur de service.
- *Résultat Réussite ((*en anglais *Success Result)*, exprime le succès de résultat retourné par les participants après exécution des services demandés.
- *Résultat Echoué (*en anglais *Failure Result)*, exprime la défaillance des nœuds participants à fournir correctement le résultat du service demandé.
- *Nombre de nœuds ressources (*en anglais *Number of Resource Nodes)*, indique le nombre de nœuds qui hébergent le service demandé.
- *Disponibilité moyenne maximale des services (*en anglais *Maximum Average Service Availability)*, désigne la probabilité moyenne maximale que les services soient disponibles au moment de la demande et qu'ils soient en mesure d'exécuter correctement les taches requises aux utilisateurs.
- *Disponibilité Moyenne Minimale des Services (*en anglais *Minimum Average Service Availability)* représente la probabilité moyenne minimale que les services soient disponibles au moment de la demande et qu'ils soient en mesure d'accomplir correctement les tâches requises pour les utilisateurs.

Ces deux derniers sont les facteurs les plus importants qui reflètent la capacité du système à assurer la caractéristique de l'approvisionnement des ressources à la demande du Cloud Computing dans un système de fourniture de services SaaS basé sur un réseau P2P. Ils sont calculés pendant l'intervalle de simulation dans le deuxième et le troixième scénario en utilisant la formule (19), sachant que pour le premier scénario, la formule est exprimée comme étant le rapport entre la somme du degré de disponibilité de chacun des pairs possédant le service dans le système et la somme de la valeur maximale de disponibilité que chacun de ces pairs peut atteindre, qui prend logiquement la valeur de 1. La disponibilité d'un service dans le premier scénario est similaire à celle habituellement utilisée. Elle est mesurée comme suit :

$$PAv_{Services} = \frac{\sum_{i=0}^{n} PAv_{Provider_{Si}}}{\sum_{i=1}^{n} Max\_PAv_{Provider_{Si}}} \quad (22)$$

Où $PAv_{Provider_{S_i}}$ est la disponibilité du pair fournisseur i de service S et $Max\_PAv_{Provider_{S_i}}$ est la valeur de disponibilité maximale qu'un pair peut atteindre et qui est égale à 1.

3.3. Résultats et Discussion

Nous avons effectué 500 simulations pour chaque scénario afin d'évaluer l'impact de l'approche proposée sur l'amélioration de la disponibilité des services et des entités ainsi que sur les performances du système. Les résultats expérimentaux sont exprimés comme la variance moyenne de toutes les expériences identiques dans le cadre des scénarios décrits ci-dessus, et variant suivant cinq autres scénarios de simulation en fonction de la taille du réseau P2P (taille : 10000, 20000, 30000, 40000, 50000).

Afin de mener des expérimentations comparatives, nous avons d'abord examiné la méthode du Clustering hybride semi-flou «*HAC-TFCM*» proposée dans des scénarios de Clustering de données multidimensionnelles de profils des clients. Pour ce faire, nous l'avons comparée à des méthodes largement utilisées, à savoir la méthode *Fuzzy C-Means «FCM»* et la méthode *Hierarchical Ascending Classification «HAC»*. Les données expérimentales sont de nature dynamique et sont collectées durant le fonctionnement du réseau au sein de chaque expérience. De plus, ces données sont considérées comme des entrées initiales pour tous les algorithmes et les formules utilisées dans ce travail. Dans le cas de la méthode *FCM* « *Fuzzy C-Means* », nous avons effectué une série d'expériences de validation du Clustering qui reflètent les propriétés de l'environnement en faisant varier le nombre de clusters entre 2 et 6. La valeur du paramètre de Fuzzification *m* a été fixée à 2, $\varepsilon = 0{,}01$ avec un nombre maximal d'itérations=100 dans toutes les expériences de simulation.

Dans cette section, le mécanisme de Clustering hybride proposée a été évaluée avec six indices de validité interne introduits précédemment. Le diagramme de la figure 7.6 montre la tendance de la variation moyenne des divers indices de validité des clusters sur l'ensemble des données expérimentales qui sont capturées pendant les scénarios de simulation.Pour une meilleure visualisation, la valeur minimale de tous les indicateurs est 0 et la valeur maximale est 1 par traitement de normalisation. Il convient de noter que l'effet de Clustering est meilleur lorsque tous les indicateurs sont aussi petits que possible, à l'exception de l'indice *PC* (*Partition Coefficient*).

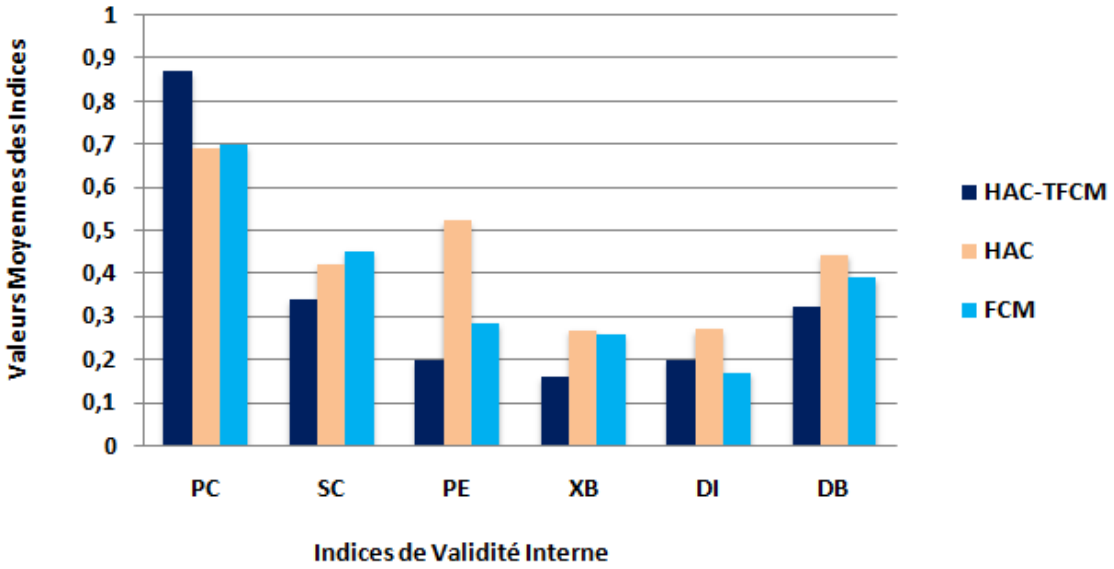


Figure 7. 6: Valeurs Moyennes des Indices de Validité Interne du Mécanisme de Clustering Hybride Proposé, de Fuzzy C-Means et de la Classification Ascendante Hiérarchique.

Tableau 7. 2: Validation du Fuzzy C-Means, Hierarchical Ascending Classification, mécanisme de Clustering Hybride Semi-Flou Hierarchical Ascending Classification-Thersholded Fuzzy C-Means, appliquée à l'ensemble de données des profils des clients.

m=2, ϵ=0.01		**FCM**					**HAC**	**HAC-TFCM**
Taille du Réseau	**Indice**	**Cluster**						
		2	3	4	5	6		
10000 noeuds	SC	0.4498	0.3918	0. 411	0. 407	0.418	0.466	0.391
	PC	0.6047	0.6867	0.6446	0.6241	**0.**8234	0.7387	0.856
	PE	0.192	0.321	0.294	0.337	0.369	0.426	**0.196**
	XB	0.156	0.181	0.231	0.310	0.242	0. 36	0.115
	DI	0.146	0.1497	0.1588	0.1807	0.175	0.1696	0.193
	DB	0.402	0.421	0.386	0.364	0.351	0.477	0.314
20000 noeuds	SC	0.475	0.431	0.421	0.415	0.345	0.391	0.312
	PC	0.642	0.597	0.601	0.776	0.868	0.632	0.871
	PE	0.215	0.226	0.237	0.249	0.304	0.515	0.207
	XB	0.172	0.167	0.184	0.164	0.106	0.197	**0.091**
	DI	0.143	0.171	0.112	**0.103**	0.154	0.285	0.185
	DB	0.412	0.386	0. 389	0. 377	0. 364	0.426	**0.291**
30000 noeuds	SC	0.510	0.485	0.471	0.382	0.341	0.394	0.314
	PC	0.590	0.633	0.670	0.669	0.792	0.617	0.837
	PE	0.215	0.229	0.336	0.326	0.364	0. 453	0.224
	XB	0.295	0.326	0.314	0.354	0.307	0.294	0.184
	DI	0.217	0.196	0.222	0. 192	0. 204	0.325	0.212
	DB	0.399	0.394	0.386	0.412	0.371	0.501	0.361
40000 noeuds	SC	0.547	0.612	0.584	0.513	0.490	0.435	0.425
	PC	0.623	0.601	0.671	0.741	0.707	0.694	**0.893**
	PE	0.206	0.216	0.258	0.312	0.359	0.610	**0.176**
	XB	0.312	0.298	0.241	0.261	0.208	0.265	0.205
	DI	0. 157	0. 132	0.159	0.211	0.193	0.265	0.194
	DB	0.421	0.429	0.405	0.381	0.364	0.396	0.350
50000 noeuds	SC	0.561	0.430	0.429	0.436	0.394	0.424	**0.243**
	PC	0.683	0.751	0.813	0.816	0.859	0.787	0.871
	PE	0.264	0.324	0.339	0.321	0.354	0.494	0.189
	XB	0.348	0.375	0.3097	0.298	0.266	0.237	0.196
	DI	0.243	0.177	0.172	0.152	0.154	0.312	0.223
	DB	0.412	0.433	0.397	0.384	0.371	0.425	0.307

Les résultats montrent que les valeurs de l'indice *PC (Partition Coefficient)* varient dans une plage supérieure à 0,5 et sont proches de +1 pour tous les algorithmes. Cependant, le mécanisme de clustering proposé peut atteindre des valeurs *PC (Partition Coefficient)* supérieures à celles constatées pour les algorithmes *FCM (Fuzzy C-Means)* et *HAC (Hierarchical Ascending Classification)*, avec une moyenne de 0,87 indiquant un bon clustering et une structure sous-jacente solide. Par opposition à l'indice *PC (Partition Coefficient)*, les plus petites valeurs de *PE (Partition Entropy)* et *SC (Partition Index)* ont été trouvées par les algorithmes *FCM (Fuzzy C-Means)* et *HAC-TFCM (Hybrid Hierarchica lAscending Classification with Thersholded Fuzzy C-Means)* ($SC_{HAC-TFCM}$=0.335, SC_{FCM}= 0.454, $PE_{HAC-TFCM}$=0.198, PE_{FCM}= 0.286) indiquant la présence de clusters bien séparés, tandis que les valeurs *PE (Partition Entropy)* pour *HAC (HierarchicalAscending Classification)* s'approchent de la limite supérieure de la plage, indiquant l'incapacité de l'algorithme à extraire toute structure de clustering de l'ensemble de données. De plus, les valeurs minimales des mesures de validité *XB (Xie et Beni's)* et *DB (Davies-Bouldin)* par *FCM (Fuzzy C-Means)* indiquent une meilleure qualité de clustering lorsque le nombre de classes est égal à 6. Parallèlement, l'algorithme *HAC-TFCM (Hybrid Hierarchical Ascending Classification with Thersholded Fuzzy C-Means)* peut atteindre des valeurs minimales de *XB (Xie and Beni's)* et *DB (Davies-Bouldin)* encore mieux que *FCM (Fuzzy C-Means)* et *HAC (Hierarchical Ascending Classification)*. Par conséquent, l'algorithme hybride proposé

est capable de détecter automatiquement le nombre idéal de clusters pour un meilleur partitionnement final du jeu de données. En outre, il convient de noter que l'indice de *Dunn* montre que le classificateur *FCM (Fuzzy C-Means)* avec 5 clusters fournit les meilleurs résultats, contrairement aux autres indices qui désignent *HAC-TFCM (Hybrid Hierarchical Ascending Classification with Thersholded Fuzzy C-Means)* comme étant plus précis. Les résultats expérimentaux vérifient la supériorité de l'algorithme hybride *HAC-TFCM (Hybrid Hierarchical Ascending Classification with Thersholded Fuzzy C-Means)* dans le regroupement efficace et fiable de l'ensemble des données du profil du client, en créant des clusters bien séparés et plus significatifs avec une grande compacité. En revanche, l'algorithme *HAC (Hierarchical Ascending Classification)* n'a pas permis de créer des clusters bien séparés et plus denses.

Les principaux objectifs des expériences de regroupement de profils des clients étaient d'étudier l'aptitude du clustering doux à découvrir de bonnes relations entre les clients et de concevoir un système de service Cloud-SaaS basé sur P2P hautement disponible afin d'améliorer l'expérience des clients. La figure 7.7 montre l'impact de l'approche proposée sur la variation des pourcentages de *Recherche Réussie (Success Search)* et de *Recherche Echouée* (*Failure Search)* par cycle pour les différents scénarios de simulation.

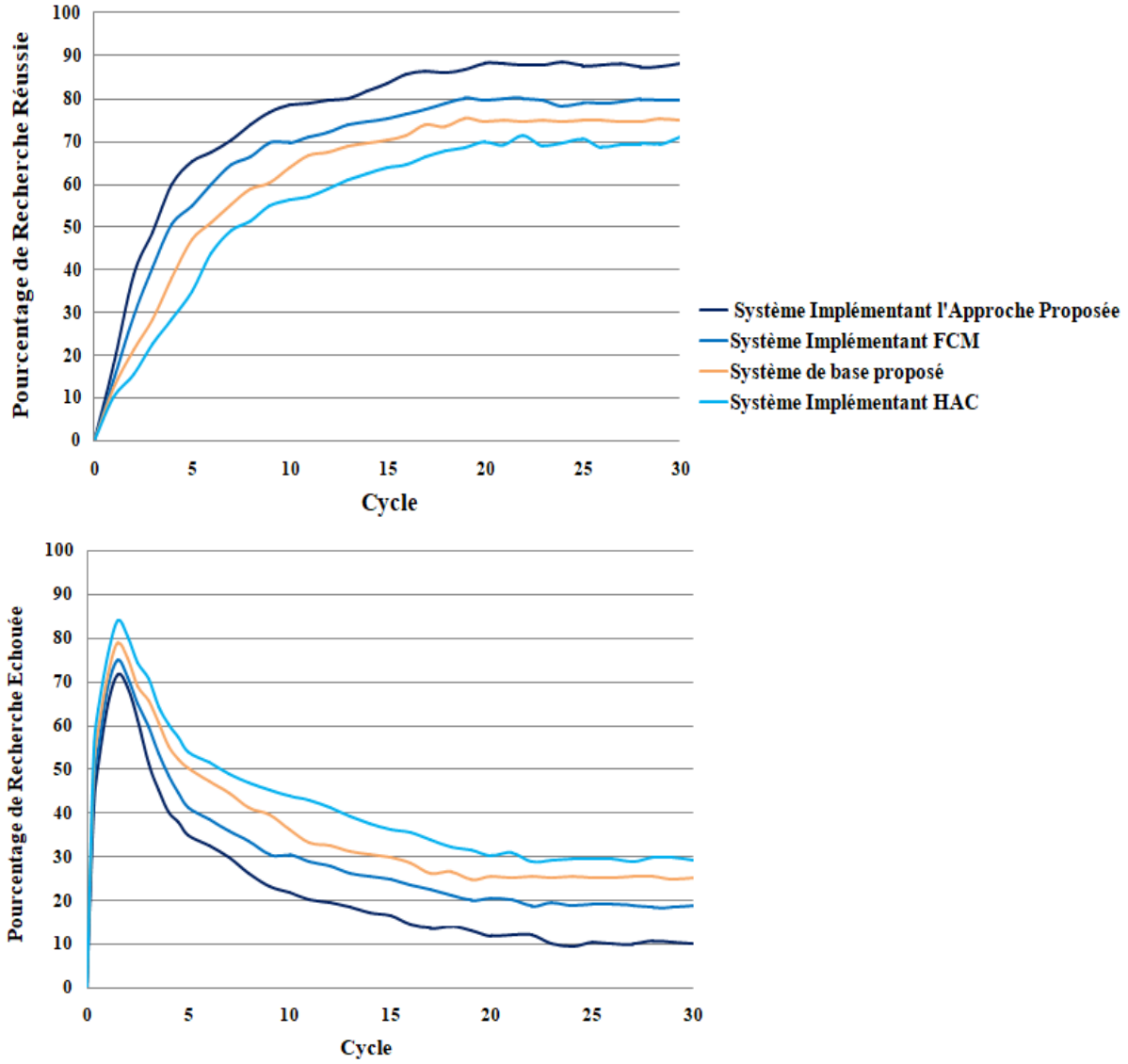


Figure 7. 7: Pourcentage de Recherche Réussie Vs Pourcentage de Recherche Echouée.

D'après les statistiques, nous observons d'abord que pendant les premiers cycles de simulation, le *Pourcentage de la Recherche Echouée (*en anglais *Failure Search Percentage)* est supérieur au *Pourcentage de la Recherche Réussie (*en anglais *Success Search Percentage)*. A partir du 4ème cycle de simulation, le *Pourcentage de la Recherche Réussie* commence à progresser en surpassant le *Pourcentage de la Recherche Echouée* dans tous les scénarios du système et durant le reste de la stimulation. En outre, les résultats montrent que *le Pourcentage de la Recherche Réussie* obtenu par le système mettant en œuvre l'approche proposée augmente constamment au fil du temps et commence à se stabiliser à partir du 16ème cycle pour atteindre un maximum de 88,19% au 20ème cycle. Selon ces statistiques, nous pouvons observer que *la Recherche Réussie* a progressé de façon continue dans le système adoptant le *FCM (Fuzzy C-Means),* atteignant un pourcentage maximal de 80.1% dans le 24ème cycle, de même que dans le système adoptant le *HAC (Hierarchical Ascending Classification)* atteignant un maximum de 71.18% dans le 24ème cycle. Par la suite, elle a stagné jusqu'à la fin de la simulation. Par ailleurs, une variation similaire et simultanée peut être observée pour le *Pourcentage de la Recherche Réussie* dans notre système de base proposé dans la première contribution, atteignant un maximum de 75,26% au 23ème cycle. Cela se traduit par la présence de pairs du réseau qui hébergent progressivement les services et deviennent des participants au système. En revanche, le *Pourcentage de la Recherche Echouée* dans les deux systèmes augmente au début de la simulation car à ce stade, le système est encore dans ses premières phases d'exécution et il y a moins de participants. A partir du 3ème cycle, nous pouvons observer une tendance à la baisse du *Pourcentage de la Recherche Echouée* au fil du temps, qui diminue continuellement et tend vers des valeurs inférieures à 30% à la fin de la simulation où le système implémentant l'approche proposée atteint des valeurs qui tendent vers 10%. Cependant, nous pouvons clairement voir que le *Pourcentage de la Recherche Réussie* a augmenté de manière significative au fil du temps dans le système mettant en œuvre l'approche proposée, atteignant un pic de 60% dans le 4ème cycle, ce qui est une conséquence évidente puisque l'approche proposée adopte un mécanisme de Clustering permettant de regrouper virtuellement les utilisateurs avec des pairs fournisseurs de services ayant des profils similaires dans des régions spécifiques, ce qui facilitera leur détection dans des intervalles de temps raisonnables. En effet, le processus de recherche n'utilise initialement que les fournisseurs appartenant au même cluster virtuel du pair demandeur reflétant les meilleures similarités, ce qui permet de réduire l'espace de recherche et d'améliorer le temps de réponse. Par conséquent, l'approche proposée offre une amélioration du *Pourcentage de la Recherche Réussie* de 10% en moyenne pendant les scénarios de simulation par rapport à notre système de base proposé dans la première contribution. De plus, les résultats de *Recherche Réussie* obtenus par le système adoptant l'approche proposée surpassent ceux obtenus par le système adoptant *FCM (Fuzzy C-Means)* et *HAC (Hierarchical Ascending Classification)*, offrant une amélioration de 8% et de 17% respectivement. Il convient de noter que les résultats du système proposé dans notre première contribution sont meilleurs que ceux obtenus par le système mettant en œuvre l'algorithme *HAC (Hierarchical Ascending Classification)*. Par conséquent, on peut formuler l'hypothèse que la qualité du regroupement affecte les performances du système, de sorte que lorsque le partitionnement final des données des profils des clients est optimal, le processus de recherche devient plus efficace.

Dans les systèmes orientés services, la préoccupation majeure est d'assurer une fourniture réussie des services qui reflètent leurs capacités à assurer la fonction de provisionnement des ressources à la demande. La figure 7.8 montre l'évolution du pourcentage des deux mesures: *Résultat Réussi* et *Résultat Echoué*, en fonction des demandes effectuées dans les différents scénarios de simulation. Ces statistiques sont représentées en supposant le même taux d'échec pour les fournisseurs de services

(25%) ainsi que la probabilité d'absence d'échec chez les fournisseurs pairs dans les différents systèmes.

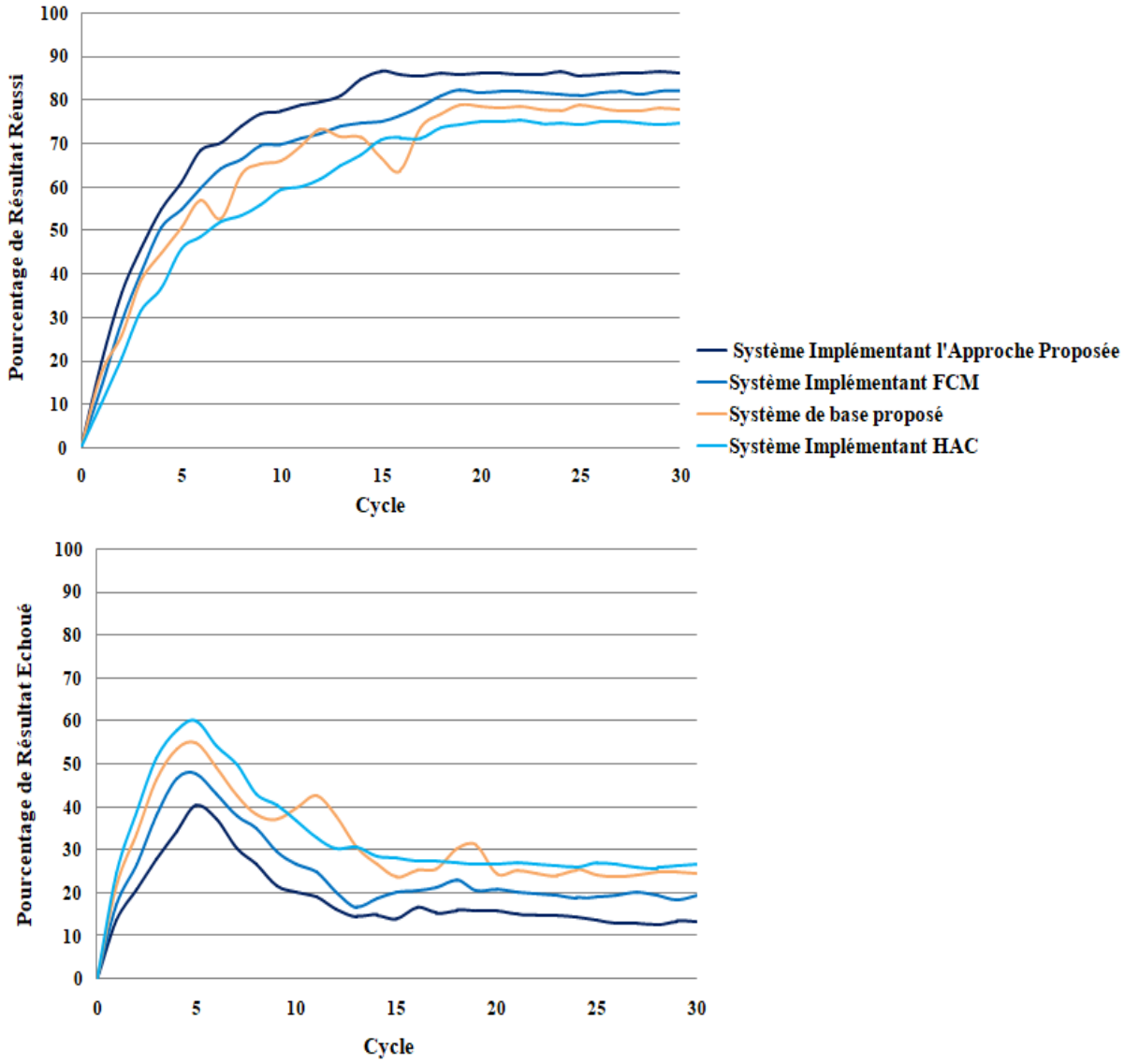


Figure 7. 8: Pourcentage de Résultat Réussi Vs Pourcentage de Résultat Echoué.

Selon ces statistiques, on peut observer que le *Pourcentage de Résultat Réussi* dans le système de base proposé dans notre première contribution a augmenté de manière continue pour atteindre un maximum de 78.91% dans le 19ème cycle, et qu'il stagne ensuitejusqu'à la fin de la simulation. Ce changement est associé avec une variation du *Résultat Echoué*, qui augmente au cours du temps jusqu'au 8ème cycle puis diminue progressivement et se stabilise jusqu'à la fin de la simulation, atteignant un minimum de 25,47%. En parallèle, on observe une augmentation significative du *Pourcentage de Résultat Réussi* durant les six premiers cycles par le système mettant en œuvre l'approche proposée, atteignant un taux de 68,5%. Cette variation est associée à une augmentation rapide du *Pourcentage de Résultat Echoué*, atteignant environ 40,32%. À partir du 6ème cycle, le *Pourcentage de Résultat Réussi* continue à augmenter rapidement pour atteindre une valeur maximale de 86,65% au 15ème cycle et commence, par la suite, à se stabiliser jusqu'à la fin de la simulation. Quant au *Pourcentage de Résultat Echoué,* il commence à diminuer avec le temps et est, ensuite, associé à une stagnation qui tend vers une valeur de 13%. Par ailleurs, nous pouvons constater que le *Pourcentage de Résultat Réussi* et de *Résultat*

Echoué dans les systèmes implémentant le *FCM (Fuzzy C-Means)* et le *HAC (Hierarchical Ascending Classification)* suit la même tendance de variation, sachant que le *Pourcentage de Résultat Réussi* atteint des valeurs inférieures en moyenne de 6% et 12.3% successivement à celles obtenues par le système adoptant notre approche proposée. Tandis qu'ils atteignent un maximum de 82,2% et 75,41% dans le 19ème et 22ème cycle successivement avant de stagner jusqu'à la fin de la simulation. En revanche, le *Pourcentage de Résultat Echoué* dans ces systèmes a atteint des valeurs moyennes de 5% et 7%, supérieures à celles obtenues par le système adoptant notre approche proposée où ce critère augmente dans les premiers cycles de simulation jusqu'à 53% et 48% avant de commencer à décliner et se stabiliser vers des pourcentages de 20% et 25%. Par conséquent, l'amélioration des taux de *Résultat Réussi* et *Résultat Echoué* indique une fiabilité et une performance supérieures de l'approche proposée sur le système de fourniture des services SaaS par rapport aux algorithmes de Clustering *HAC (Hierarchical Ascending Classification)* et *FCM (Fuzzy C-Means)*. En outre, il ressort de ces résultats que le taux de fourniture efficace de services dans le système intégrant l'approche proposée est significativement amélioré en moyenne de 86,91% en comparaison avec le pourcentage de 78,91% obtenu par le système proposé dans notre première contribution. Dans la mesure où le succès de 1 % est plus important, l'augmentation obtenue par le système mettant en œuvre l'approche proposée est en moyenne de 9,69 % par rapport à celui qui n'utilise pas le mécanisme de mise en cluster proposée. Ce dernier génère une couche virtuelle dans laquelle des pairs utilisateurs et fournisseurs similaires ayant des caractéristiques fonctionnelles différentes et complémentaires peuvent être regroupés. En maintenant la diversité, le système peut conserver une opérabilité continue par des pairs qui peuvent appartenir à des régions géographiques et des fuseaux horaires différents et similaires. En d'autres termes, cela permet aux services d'être hébergés et exécutés par des pairs ayant des propriétés variables dans différents lieux géographiques et d'être présents dans plusieurs zones physiques. Cela aura un impact positif sur la disponibilité des services requis et les performances du système.

Ici, nous étudierons le comportement de la probabilité de disponibilité des services SaaS en tenant compte de l'impact de la conception du système Cloud-SaaS basé sur un réseau P2P hybride pour le traitement des services à distance.

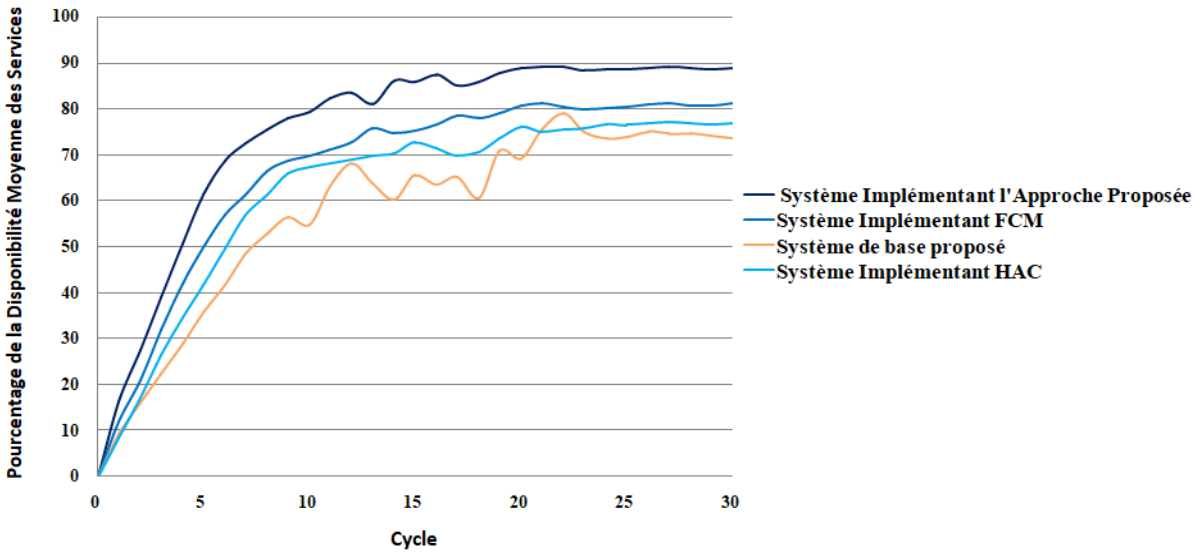


Figure 7. 9: Pourcentage de la Disponibilité Moyenne des Services.

Tableau 7. 3: Valeurs moyennes, minimales et maximales des courbes de la figure 7.9.

	Taille du Réseau (noeuds)	Disponibilité Moyenne (%)	Disponibilité Maximale (%	Disponibilité Minimale (%)
Système Implémentant l'Approche Proposée	10000	87,18	87,97	63.51
	20000	87,67	89,25	65.46
	30000	89,05	90,34	70.14
	40000	88,24	88,85	68.83
	50000	88,56	89,54	69.25
Système Implémentant FCM	10000	78,05	83,21	59.28
	20000	78,34	80,92	56.75
	30000	78,23	82,18	59.07
	40000	80,15	80,36	56.04
	50000	79,36	79,68	54.61
Système Implémentant HAC	10000	73,23	75,31	29.16
	20000	73,05	75,20	27.88
	30000	75,15	75,42	31.28
	40000	74,36	77,03	35.94
	50000	74,34	77,64	36.47
Système de base proposé	10000	75,17	77,31	39.22
	20000	74,61	77,22	38.91
	30000	75,82	77,65	40.36
	40000	78,54	80,03	41.56
	50000	76,09	78,52	40.62

La figure 7.9 montre la variation du pourcentage de disponibilité des services dans les différents scénarios de simulation. La disponibilité des services dans le système implémentant l'approche proposée augmente rapidement au début de la simulation, atteignant un pourcentage de 61,38% au 5ème cycle. Cependant, la disponibilité décélère entre les 5ème et 11ème cycles, puis fluctue légèrement entre les 11ème et 19ème cycles, car le système réorganise périodiquement son infrastructure virtuelle. Cette variation ne diminue pas de 80% jusqu'à ce qu'elle se stabilise, atteignant une disponibilité maximale de 89,19% au 21ème cycle. En outre, les pourcentages de disponibilité du service dans les systèmes implémentant les algorithmes de Clustering, y compris *FCM (Fuzzy C-Means)* et *HAC (Hierarchical Ascending Classification)*, varient de manière similaire au système précédent, atteignant une disponibilité maximale de 81,27% et 76,12% respectivement. Cependant, le système mettant en œuvre l'approche proposée a réussi à atteindre des valeurs d'amélioration moyennes de 9% et 14% par rapport au système adoptant l'algorithme de Clustering *FCM (Fuzzy C-Means)* et *HAC (Hierarchical Ascending Classification)* successivement. En outre, la figure 7.9 montre également que la disponibilité du service dans le système de base proposé augmente lentement au début de la simulation pour atteindre un pic de 48,4% dans le 7ème cycle. En outre, nous constatons que la disponibilité du service se détériore considérablement à plusieurs moments entre les 12ème et 21ème cycles, puis elle commence à se stabiliser dans le temps pour atteindre un maximum de 78,14 % au 24ème cycle. À notre avis, le comportement volatile du pourcentage de disponibilité est dû au fait que la disponibilité du service dépend entièrement de la disponibilité individuelle des pairs exécutant le système, ce qui signifie que tous leurs propres services sont susceptibles d'être affectés par leurs propres défaillances.

Comme le montre la dernière figure, la variation du pourcentage de disponibilité du service dans le système adoptant l'approche proposée suit une voie similaire à celle des autres systèmes, malgré le fait que le pourcentage de disponibilité du service de ce système reste plus élevé. De plus, en comparant les trois courbes associées aux scénarios dans lesquels un des algorithmes de Clustering est utilisé, nous remarquons que lorsque l'algorithme présente des performances plus faibles sur l'ensemble des données expérimentales et que la qualité du Clustering décline, le taux d'amélioration de la

disponibilité du service diminue et se rapproche progressivement du système proposé dans notre première contribution où le Clustering n'est pas utilisé.

Les résultats de ces simulations sont résumés dans le tableau 7.3 dont les statistiques montrent une tendance incrémentale entre la qualité du Clustering et la disponibilité des services, ce qui justifie à nouveau l'hypothèse qu'un modèle de Clustering optimal est nécessaire pour la conception d'un système de fourniture de services performant et fiable. En outre, les statistiques présentées montrent que la méthode de mise en cluster et de formation de couche virtuelle envisagée par l'approche proposée a un effet considérable sur l'amélioration du pourcentage de disponibilité des services SaaS dans un système Cloud basé sur P2P par rapport au système de base proposé dans notre première contribution. La distribution efficace des fournisseurs implique la formation de zones condensées de services requis hébergés ou exécutés par un ensemble de pairs. Par conséquent, l'emplacement condensé d'un service requis dans une région spécifique de la couche virtuelle augmente la probabilité de sa disponibilité au moment de sa demande.

L'approche proposée peut atteindre une disponibilité moyenne des services de 88% versus 76% avec une augmentation totale de 12% par rapport au système de base proposé dans notre première contribution. De plus, le système mettant en œuvre l'approche proposée peut atteindre une disponibilité de service moyenne maximale de 89,19 % et une disponibilité de service moyenne minimale de 67,43 % dans les calculs de disponibilité versus 78,94 % et 40,13 % dans le système de base proposé, ce qui signifie que même dans les cas de défaillance, le système reste relativement efficace. Par conséquent, au fur et à mesure qu'il y a plus de pairs fournisseurs disponibles appartenant à un ou plusieurs clusters représentés par des nœuds virtuels en plus des liens de communication, le système est plus disponible et répond plus rapidement. En outre, il semble que notre approche offre une combinaison parfaite de calculs utilisant des disponibilités basées sur le temps pour une meilleure analyse numérique, des disponibilités basées sur l'activité pour des informations plus précises sur le système pendant son fonctionnement, et des disponibilités basées sur la présence surveillant l'état général des entités du système.

L'approche proposée répond largement à la nature à grande échelle permettant la conception d'un système de fourniture de services Cloud-SaaS basé sur un réseau P2P capable de trouver le meilleur compromis entre la satisfaction du client en termes de qualité de service et le coût du système.

4. Discussion Comparative

Les travaux existants les plus pertinents pour améliorer la disponibilité dans le Cloud sont comparés à notre travail proposé afin de démontrer que ce dernier est meilleur. Il convient de noter que ces travaux se distinguent de notre approche proposée par leurs méthodes d'analyse des performances, chacune d'entre elles adopte un environnement expérimental distinct et se base sur des critères d'évaluation différents de ceux utilisés dans notre étude. Par conséquent, le champ de comparaison sera fait en termes de conception des techniques proposées ainsi que de leur impact sur la performance de l'environnement du Cloud, particulièrement sur la disponibilité des services et des ressources.

Afin d'assurer une haute disponibilité et fiabilité, les fournisseurs de services Cloud introduisent divers mécanismes tels que la redondance, la planification et la tolérance aux pannes dans leurs systèmes Cloud. Suite à ces mécanismes, les auteurs de (Hussein et Mousa, 2014) (Jhawar et Piuri 2013) (Lu et al. 2010) (Noraziah et all. 2013) (Park et al 2003) (Rahmati et Rahmani 2017) (Rama et Reddy, 2013) (Saadat et Rahmani 2012) (Sashi et Thanamani, 2010) (Sashi et Thanamani 2011) (Sun et al. 2012) (Bsoul et al 2012) (Thein et Park, 2009) (Vijaya et Ramachandhra 2014) (Wei et al. 2010) (Singh et al. 2012) ont présenté diverses stratégies efficaces. Ces approches proposées aient contribué

à améliorer la disponibilité et l'accessibilité des données, à prévenir les perturbations du réseau et à assurer la continuité du système. Cependant, elles souffrent toujours de problèmes liés à leur noyau de fonctionnement, qui repose principalement sur la nécessité d'une connaissance globale des réseaux et de leurs exploitations complexes, stockée et calculée pendant leur traitement. Outre le volume de données dynamiques analysées dans des environnements distribués à grande échelle, ces techniques produisent généralement d'énormes quantités de données circulant dans le réseau, ce qui entraîne une augmentation inutile des efforts et des coûts en termes de : stockage, processeur, frais généraux financiers, frais généraux de maintenance, gestion complexe et cohérente des données. De plus, les délais de traitement et d'exécution dans les grands réseaux, l'augmentation du temps d'inactivité des pairs et la limitation de la bande passante du réseau créent des goulots d'étranglement très sensibles à l'accès simultané aux données dans l'environnement distribué.

Contrairement à ces travaux, l'approche proposée répond aux exigences de disponibilité à faible coût en maintenant une couche virtuelle et dynamique ajoutée à l'architecture du système de service Cloud-SaaS basé sur un réseau P2P. Cette couche est construite à l'aide d'un mécanisme de mise en cluster hybride de vecteurs de profils de clients qui sont répartis, en fonction des similarités de leurs profils, dans plusieurs clusters. Elle permet la distribution de pairs fournisseurs et la création de zones condensées de service hébergées sur plusieurs nœuds, améliorant ainsi le degré de probabilité de disponibilité des services dans des régions spécifiques au moment de la demande. En outre, le processus de recherche de services est initialement effectué dans la couche virtuelle au lieu d'utiliser tous les fournisseurs du système et, dans le cas d'un échec, il sera lancé dans la couche réelle des « Serveurs Cloud Partiels ». L'approche proposée assure une haute disponibilité des services ainsi que des pairs qui les détiennent, une communication efficace et une efficacité du réseau. En d'autres termes, notre système est capable de garantir que les clients interagissent en permanence et correctement avec les applications et services SaaS, et que le désabonnement des pairs (normalement par déconnexion, anormalement par défaillance) n'affecte pas les performances du système. Par conséquent, l'approche proposée fournit une infrastructure optimale tolérante aux pannes qui répond largement à la nature dynamique des pairs ainsi qu'à leurs comportements variables dans la couche réseau réelle du système de prestation de services. Elle permet de réduire l'espace de recherche et le coût de communication, de maintenir un faible trafic réseau, d'améliorer le temps de réponse et le taux de réussite de l'exécution des services.

En adoptant diverses méthodes de regroupement, d'autres auteurs (Chavan et Kaveri 2014) (Durgadevi et Srinivasan 2016) (Malathy at al. 2013) (Meenakshi et al. 2019) (Oh et Kim 2019) (Saravanakumar et al. 2021) (Wu et al. 2014) ont proposé plusieurs techniques afin de maintenir une fiabilité et une disponibilité adéquates pour les besoins des utilisateurs Cloud. Cependant, les techniques proposées souffrent de plusieurs défis, notamment un trafic réseau élevé, une surcharge importante, un faible équilibrage de charge, une efficacité réduite dans les environnements à grande échelle et une manipulation des données et des méthodes de Clustering simples. Comme mécanisme de mise en cluster, l'approche proposée utilise un algorithme de Clustering hybride semi-flou « *HAC-TFCM* » pour effectuer un nouveau Clustering des pairs du réseau en plus de leur Clustering dans la couche réseau réelle. La technique de Clustering proposée surpasse les techniques présentées dans (Chavan et Kaveri 2014) et (Malathy at al. 2013), car elle permet à chaque client d'être automatiquement associé au nombre approprié de clusters jugés proches, sans connaissance à priori du numéro de cluster et du nombre de clusters auxquels il doit appartenir. En comparaison avec (Wu et al. 2014), l'approche que nous proposons a pu gérer des données complexes et dynamiques constituées de vecteurs de profil des clients du système qui déterminent leurs propres comportements dans le réseau. De plus, comparé à (Meenakshi et al. 2019) et (Oh et Kim 2019), notre système est évolutif et ses performances restent stables dans des environnements à grande échelle, ce qui permet de concevoir un système de prestation

de services SaaS basé sur un réseau P2P capable d'obtenir le meilleur compromis entre la satisfaction du client en termes de qualité de service et le coût du système. En outre, notre article se distingue des travaux présentés ci-dessous car il propose un modèle de mesure de la disponibilité des services qui prend en compte différentes entités à plusieurs niveaux du système et utilise différents types de disponibilité.

5. Conclusion

Dans le monde technologique contemporain, la technologie exige une approche efficace pour minimiser les coûts globaux et améliorer la fiabilité du système en offrant la meilleure qualité de services. Dans ce chapitre, nous avons proposé une approche pour améliorer la disponibilité permanente des services SaaS dans un environnement « Hybrid P2P Network-Based Remote Treatment SaaS Cloud Computing ». Cette approche vise à répondre aux exigences du provisionnement des services à la demande et à maintenir les performances du Cloud Computing en surmontant les problèmes de défaillances du système et de la dynamique des pairs. Elle suit principalement le rythme dynamique et rapide de ces pairs en analysant leurs comportements individuels et en les représentant sous la forme d'une matrice de vecteurs de profils de clients. Par ailleurs, un mécanisme de Clustering hybride semi-flou a été proposé en combinant dans un processus d'hybridation séquentiel l'algorithme de « Classification Ascendante Hiérarchique » et le « C-Moyennes Floues à Seuil », permettant à chaque vecteur de profil du client d'être automatiquement associé au nombre approprié de clusters considérés comme proches, sans aucune connaissance préalable du numéro de cluster et du nombre de clusters auxquels il devrait appartenir. Il vise à fournir une infrastructure virtuelle et optimale pour organiser les pairs du système en fonction de la similarité de leur profil en clusters distincts, où chaque cluster est représenté par un nœud virtuel formant ensemble une couche virtuelle. Cette couche assure à la fois la distribution des pairs utilisateurs et fournisseurs de services et la formation de zones denses de chaque service qui intéressent un ensemble de pairs qui les hébergent ou les utilisent, ce qui améliore considérablement la probabilité de disponibilité de ces services dans des régions spécifiques de la couche virtuelle au moment de la demande. En outre, un modèle de mesure de disponibilité des services a été proposé. Il est principalement basé sur l'utilisation de la couche virtuelle du système et prend en compte différentes entités à différents niveaux du système, notamment la probabilité de disponibilité des fournisseurs de services et la probabilité de disponibilité des nœuds virtuels. Pour obtenir les meilleures performances dans un système de provision de services, nous avons utilisé différents types de disponibilité existant dans la littérature pour mesurer la disponibilité de ces entités : « disponibilité basée sur le temps, disponibilité basée sur l'activité, disponibilité basée sur la présence de k-de-n », qui sont affinés pour inclure des considérations spéciales dans leurs calculs en raison de leur utilisation particulière.

Nous avons examiné la méthode de Clustering hybride semi-flou proposée sur la base de six indices de validité interne et les résultats expérimentaux vérifient la supériorité de l'algorithme de Clustering hybride semi-flou « *HAC-TFCM* » par rapport à *FCM* et *HAC* dans le regroupement efficace et fiable de l'ensemble de données de profil de client en créant des clusters mieux séparés et plus significatifs avec une grande compacité. En outre, les résultats de l'évaluation expérimentale avec PeerSim fournissent des preuves convaincantes que l'approche proposée peut améliorer efficacement la probabilité de disponibilité du service SaaS et la fiabilité du système Cloud-P2P pour le traitement de services à distance offrant une disponibilité moyenne du service de 89% avec une amélioration de 7%. Par conséquent, l'approche proposée fournit la preuve irréfutable qu'elle répond largement à la nature à grande échelle des systèmes distribués et qu'elle est capable d'obtenir le meilleur compromis pour maintenir la qualité du service en termes de disponibilité, de performance et de coût.

Conclusion

Au vu du progrès rapide des technologies de l'information et de l'Internet, le Cloud Computing s'est vite imposé comme étant le modèle de facto actuel permettant de fournir des ressources informatiques aux clients à la demande. La manière dont les industries informatiques mènent leurs activités, en déportant sur des serveurs distants les traitements, la gestion des données et leur stockage, effectués habituellement localement pour y accéder en tant que service, a été révolutionnée par ce paradigme technologique émergeant. Les analystes de l'industrie technologique prévoient la poursuite de la croissance du marché rentable du logiciel en tant que service. Ce dernier est au cœur des modèles de prestation de services Cloud. Ledit logiciel permet de fournir des applications logicielles accessibles par les clients en tant que service via l'Internet et hébergées de manière centralisée par les fournisseurs. Tout cela en éliminant les contraintes de maintenance, d'installation, de déploiement et de gestion complexe des logiciels et du matériel.

En dépit des avantages fonctionnels et économiques associés à la migration vers le Cloud, sa conception centralisée soulève de nombreux problèmes en termes de point de défaillance unique, de coût, de consommation énergétique et d'emplacement géographique. En effet, les enjeux d'infrastructure auxquels sont confrontés les concepteurs et les fournisseurs de services Cloud ont retenu une attention particulière. Récemment, la distribution du Cloud Computing en utilisant les technologies Pair-à-Pair est apparue comme une solution clé permettant aux fournisseurs de construire une infrastructure Cloud évolutive et performante à des coûts réduits par la coopération et le partage des ressources des pairs du réseau. Toutefois, une telle architecture résultante consiste en un environnement hétérogène et dynamique, dont la garantie d'une disponibilité satisfaisante des services représente un enjeu important qui peut entraver son adoption. En effet, la prestation correcte des services est influencée par les propriétés physiques et logiques des pairs dans le réseau, car toute raison de défaillance matérielle ou logicielle a pour conséquence de rendre les pairs du réseau inactifs et inaccessibles pour répondre aux besoins des clients. Les prestataires de services dans les « Cloud basés sur les réseaux Pair-à-Pair» sont censés assurer une disponibilité appropriée des services en vue de respecter la caractéristique fondamentale du «Cloud Computing», soit la fourniture de ressources à la demande.

L'objectif de cette thèse est l'étude de la faisabilité et la réalisation d'une solution Cloud-SaaS distribué en utilisant un réseau Pair-à-Pair comme infrastructure pour fournir des services à distance. Dans cette solution, nous nous focalisons sur le problème de la disponibilité des services qui a été une préoccupation croissante pour les fournisseurs et les utilisateurs qui ont opté pour des environnements distribués, notamment le Cloud basé sur les réseaux Pair-à-Pair. Pour tirer profit de cette architecture prometteuse et établir une collaboration technologique « CC-P2P » fructueuse, il est crucial d'adopter des mécanismes matures et puissants, en particulier ceux issus de l'exploration de données, afin d'analyser et de répondre efficacement au défi de la disponibilité des services et de maintenir des performances adéquates. Conformément à ces objectifs, les principales contributions de la thèse sont reprises ci-dessous:

Dans la première contribution, nous avons proposé un système Cloud-SaaS fondé sur un réseau Pair-à-Pair hybride pour le traitement des services à distance. Cette proposition vise principalement à résoudre le problème d'infrastructure du Cloud Computing en tirant parti de la tendance émergente du développement du Cloud basé sur les réseaux Pair-à-Pair, ce qui a permis la mise en place du modèle SaaS distribué pour la première fois. Le système proposé fait intervenir une

architecture Cloud récente qui utilise un réseau Pair-à-Pair hybride comme infrastructure pour la prestation distante de services sous forme de SaaS à des pairs qui n'en disposent pas. Il permet aux pairs de bénéficier de services SaaS qui sont initialement fournis par le Cloud mais hébergés et exécutés par d'autres pairs du réseau. Cette architecture consiste en un Cloud réparti dynamique, disposant d'une infrastructure minimale composée d'un *Administrateur*, de *Serveurs Cloud Partiels* et de *Clients*, dont les deux dernières entités représentent des super-pairs et des pairs normaux, constituant ainsi le réseau Pair-à-Pair hybride. L'administrateur a pour rôle d'alimenter le système en services. Les serveurs Cloud partiels sont connectés sur une couche réseau et sont responsables de l'indexation des services et de la coordination entre les clients. Ces derniers peuvent être des *clients participants* qui hébergent des services et les exécutent à distance pour le compte des *clients utilisateurs* qui cherchent à exploiter ces services. Ils utilisent le principe des réseaux P2P pour permettre une connexion directe entre eux en vue d'invoquer un service souhaité via la technologie Java RMI. En effet, les clients sont regroupés en clusters gérés par les serveurs Cloud partiels formant la couche réseau réelle du système. Le système proposé a été implémenté sous la forme d'un prototype Java. Son fonctionnement et le comportement de ses principales entités ont été adaptés par la suite au simulateur PeerSim à des fins d'évaluation. Dans ce contexte, les résultats obtenus sont intéressants et révèlent que le système proposé présente des performances notables, en offrant un fort taux de réussite atteignant les 80%. Ils font également la preuve que cette conception récente est en mesure de constituer une solution efficace aux problèmes d'infrastructure et un support technologique pour une prestation de services satisfaisante.

Dans la deuxième contribution, nous avons proposé une approche basée sur un mécanisme de classification non supervisée pour améliorer la disponibilité des services dans un système « Cloud-SaaS basé sur un réseau Pair-à-Pair hybride pour le traitement des services à distance ». L'approche proposée suit et analyse le comportement variable et dynamique des clients «participants et utilisateurs» dans la couche réseau réelle d'un tel système afin de collecter leurs informations et de les représenter sous la forme d'une *Matrice de Vecteurs de Profils de Clients*. Par ailleurs, le jeu de données de cette matrice a subi un prétraitement au moyen de la technique de normalisation Z-Score afin de les placer sur une échelle commune. En outre, nous avons proposé un mécanisme de mise en cluster hybride qui combine dans un processus séquentiel l'algorithme de *Classification Ascendante Hiérarchique* avec l'algorithme *C-Moyennes Floues à Seuil*. Il a pour objectif de classer et d'agréger les clients du système en fonction de leurs similarités de profil. Ce mécanisme exploite la matrice de vecteurs de profils de clients afin de réorganiser les pairs du réseau selon la similarité de leurs profils dans des clusters distincts, dont chaque cluster est représenté par un *Nœud Virtuel*, formant ensemble une *Couche Virtuelle*. Il en résulte que cette dernière crée des régions denses pour chaque service qui présente un intérêt pour un ensemble de pairs appartenant au même cluster, permettant ainsi de renforcer nettement la probabilité de disponibilité de ces services au niveau de zones spécifiques de la couche virtuelle au moment de la demande. De plus, en se basant sur la couche virtuelle, nous avons proposé un modèle permettant d'estimer la disponibilité des services du système ainsi que les pairs qui les détiennent. En considérant diverses entités à différents niveaux du système, le modèle proposé fait intervenir des métriques de disponibilité tirées de la littérature qui ont été redéfinies et reformulées, notamment la disponibilité basée sur le temps, la disponibilité basée sur l'activité et la disponibilité basée sur la présence k-of-n.

L'approche proposée a été implémentée et expérimentée via le Simulateur PeerSim. À la lumière des résultats obtenus, le mécanisme de clustering hybride proposé présente une efficacité plus importante sur le jeu de données des clients par rapport aux algorithmes *C-Means Flous* et de la *Classification Ascendante Hiérarchique*, permettant de générer des clusters mieux séparés et dotés d'une forte compacité. En outre, les résultats expérimentaux ont révélé que l'approche proposée

peut réellement contribuer à améliorer la probabilité de disponibilité des services SaaS et la fiabilité du système CC-P2P pour le traitement des services à distance, offrant une disponibilité moyenne du service de 89 %. Il ressort de nos évaluations que la qualité du clustering impacte fortement le taux de disponibilité des services. De plus, l'approche proposée a prouvé ses capacités pour gérer des données complexes et s'adapter à la dynamique et l'évolutivité du réseau en assurant la continuité du fonctionnement du système et la maintenance de ses performances. Elle permet de réduire l'espace de recherche et le coût de communication, de maintenir un faible trafic réseau, d'améliorer le temps de réponse et le taux de réussite de l'exécution des services. Par conséquent, l'approche proposée peut aboutir au meilleur compromis pour maintenir la qualité du service en termes de disponibilité, de performance et de coût.

Bibliographie

(Abdellaoui 2014) N.N.Abdellaoui, La classification hiérarchique ascendante, mémoire Licence, Université Abou Bakr Belkaid– Tlemcen, (2014).

(Aberer 2002) K. Aberer, “Scalable Data Access in P2P Systems Using Unbalanced Search Trees”, In 4th Workshop on Distributed Data and Structures (WDAS’2002), Carleton Scientific, 2002.

(Adekunle et Osimosu 2013) A. Adekunle, E.Osimosu. “Service Availability in Cloud Computing : Threats and Best Practices.” Bachelor Thesis (2013).

(Aggarwal et Reddy 2013) C.C. Aggarwal, C.K. Reddy. Data Clustering: Algorithms and Applications (1st. ed.). Chapman & Hall/CRC, (2013).

(Ahmed et al. 2017) H.A. S. Ahmed, M.H.Ali, L.M. Kadhum, M.F.B. Zolkipli, Y.A. Alsariera, A Review of Challenges and Security Risks of Cloud Computing, Journal of Telecommunication, Electronic and Computer Engineering, Vol. 9 No. 1-2, (2017).

(Ahuja et Mani 2012) S.P. Ahuja, S. Mani, Availability of services in the era of cloud computing, Network and Communication Technologies1 (2012), 2–6.

(Al King 2010) M. R.Al King, Localisation de sources de données et optimisation de requêtes réparties en environnement pair-à-pair, thèse de Doctorat, Université Toulouse III – Paul Sabatier, 2010.

(Alhaisoni et Liotta 2009) M.Alhaisoni, A.Liotta, Characterization of signaling and traffic in Joost, Peer-to-Peer Netw Appl (2009) 2:75–83.

DOI 10.1007/s12083-008-0015-5

(Alotibi et al. 2019) B.Alotibi, N.Alarifi, M.Abdulghani, L.Altoaimy, Overcoming Free-Riding Behavior in Peer-to-Peer Networks Using Points System Approach, Procedia Computer Science, Volume 151, 2019, pp 1060-1065.

(Alshayeji et al. 2018) M.H. Alshayeji, M.Al-Rousan, E.Yossef, H.Ellethy: A Study on Fault Tolerance Mechanisms in Cloud Computing, International Journal of Computer Electrical Engineering, Volume 10, Number 1, March 2018. doi: 10.17706/ijcee.2018.10.1.62-71

(Amad et al. 2012) M.Amad, A.Meddahi, D.Aissani, Peer to Peer Networks Management Survey, International Journal of Computer Science, Vol. 9, Issue 1, No 3, January 2012.

(Ambulkar et Borkar 2012) B.Ambulkar, V.Borkar : Data Mining in Cloud Computing, MPGI National Multi Conference 2012 (MPGINMC-2012) 7-8 April, 2012.

(Amelio et Tagarelli 2019) A.Amelio, A.Tagarelli : « Data Mining: Clustering ». Encyclopedia of Bioinformatics and Computational Biology (2019).

(Arunachalam et Vinayakumar 2021) A.Arunachalam, R.Vinayakumar: A study of the resource discovery approaches in Mobile Peer-to-Peer Networks. TechRxiv. Preprint. (2021).

https://doi.org/10.36227/techrxiv.13726114.v1

(Ataallah et al. 2015) S. M. A. Ataallah, S. M. Nassar and E. E. Hemayed, "Fault tolerance in cloud computing - survey," 11th International Computer Engineering Conference (ICENCO), 2015, pp. 241-245, doi: 10.1109/ICENCO.2015.7416355.

(Ayram et Karkkainen 2006) S.Ayram, T.Karkkainen : Introduction to partitioning-based clustering methods with a robust example, (2006).

(Babaoglu et al. 2002) O. Babaoglu, H. Meling, and A. Montresor. Anthill : A framework for the development of agent-based peer-to-peer systems. In The 22th International Conference on Distributed Computing Systems (ICDCS ’02). IEEE Computer Society, 2002.

(Babaoglu et Marzolla 2014) O.Babaoglu, M.Marzolla, Peer-to-Peer Cloud Computing, 2014.

(Bacache-Beauvallet et Cagé 2016) M.Bacache-Beauvallet, J.Cagé, "Peer-to-Peer": the True Economic Issues, Revue d'économie Industrielle, (2016), p. 11-39.

(Bala et Chana, 2012) A.Bala, I.Chana: Fault Tolerance- Challenges, Techniques and Implementation in Cloud Computing. International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, 2012.

(Bandela et al. 2015) S.Bandela, R.Gadde, S.Pabboju, Survey on Cloud Computing Technologies and Security Threats, International Conference on Emerging Trends in Electronics & Telecommunications (ICETET-15), 2015.

(Bashmal et al. 2017) L.Bashmal, A.Almulifi, H.Kurdi, Hybrid Resource Discovery Algorithms for Unstructured Peer-to-Peer Networks, Procedia Computer Science, Volume 109, 2017, pp 289-296.

(Bauer et Adams 2012) E.Bauer, R.Adams. Reliability and Availability of Cloud Computing, First Edition, Institute of Electrical and Electronics Engineers. John Wiley & Sons, Inc USA. (2012).

(Begredj et Madaoui 2013) F.Begredj, L.Madaoui, Problème de recherche des services dans les réseaux P2P, Mémoire de master, Université A/Mira de Béjaïa, 2013.

(Behera et Chakravarty 2015) A.Behera, S.Chakravarty, A Comparative Study on Hierarchical, K-Means and Fuzzy C-Means Clustering Algorithms and Application to Microarray Gene Expression Data, International Journal for Advance Research in Engineering and Technology, Volume 3, Issue I, (2015).

(BENAC et al. 2015) J.BENAC, O.LALLEMENT, TAJ : A.TRECA, Caisse des Dépôts : R.AMSELLEM, D.CELISSE, Commissariat Général à l'Égalité des Territoires (CGET) : A.FAURE, M.LAGET, Direction Générale des Entreprises (DGE) - Ministère de l'Économie, de l'Industrie et du Numérique : F.KAROLAK, C.MORA, Secrétariat Général à la Modernisation de l'Action Publique (SGMAP) : C.Faivre, Guide sur le Cloud Computing et les Datacenters à l'attention des collectivités locales, (2015).

(Bender 2007) M.Bender, Advanced Methods for Query Routing in Peer-to-Peer Information Retrieval, Thèse de Doctorat, Université des Saarlandes, (2007).

(Benoit 2006) A.Benoit, Algorithmique des réseaux et des télécoms - Notes de cours (ENS Lyon, M1), 2006.

(Bensaid et al. 1996) A.M. Bensaid, L.O.Hall, J.C. Bezdek, L.P. Clarke, M.L.Silbiger, J.A. Arrington, R.F.Murtagh, Validity-guided (Re) Clustering with applications to image segmentation, IEEE Transactions on Fuzzy Systems, (1996).

DOI: 10.1109/91.493905

(Benz et Bohnert 2013) K. Benz and T. Bohnert, Dependability Modeling Framework: A test procedure for High Availability in Cloud Operating Systems, in: Vehicular Technology Conference, Las Vegas, USA, September 2–5, 2013, pp. 1–8.

(Bezdek et al. 1984) J.C. Bezdek, R.Ehrlich, W.Full, FCM: The fuzzy c-means clustering algorithm, Computers & Geosciences, Volume 10, Issues 2–3, (1984), pp. 191-203.

(Brenner et al. 2014) S. Brenner, B. Garbers and R. Kapitza, Adaptive and Scalable High Availability for Infrastructure Clouds, in: 4th International Conference on Distributed Applications and Interoperable Systems, Berlin, Germany, June 3–4, 2014, pp. 16–30.

(Bruda et al. 2012) S.D.Bruda, F.Salehi, Y.Malik, B.Abdulrazak, A Peer-to-Peer Architecture for Remote Service Discovery, The 2nd International Symposium on Frontiers in Ambient and Mobile Systems (FAMS), Procedia Computer Science 10 (2012) 976 – 983.

(Bsoul et al 2012) M. Bsoul, A. Al-Khasawneh, Y. Kilani and I. Obeidat, A threshold-based dynamic data replication strategy, The Journal of Supercomputing 60 (2012), 301–310.

(Chaczko et al. 2011) Z.Chaczko, V.Mahadevan, S.Aslanzadeh, C.Mcdermid, Availability and Load Balancing in Cloud Computing, International Conference on Computer and Software Modeling, IPCSIT vol.14, Singapore, (2011).

(Chafika 2006) R.Chafika : Le clustering des données : une nouvelle approche évolutionnaire quantique, Université Mentouri de Constantine, mémoire de Magistère 2006.

(Chana et Kaur 2013) I.Chana, T.Kaur, Delivering IT as A Utility- A Systematic Review, International Journal in Foundations of Computer Science & Technology (IJFCST), Vol. 3, No.3, (2013).

(Chaudhary et Surolia, 2015) N.Chaudhary, J.Surolia, A survey on peer to peer system applications, International Journal of Innovative Computer Science & Engineering, Volume: 2, Issue: 1, 2015.

(Chavan et Kaveri 2014) V. Chavan, P.R. Kaveri, Clustered virtual machines for higher availability of resources with improved scalability in cloud computing, in: First International Conference on Networks and Soft Computing, Guntur, India, August 19–20, 2014, pp. 221–225.

(Chen et al. 2018) R. Chen, Q. Hua, Y. Chang, B. Wang, L. Zhang and X. Kong, A Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks, in IEEE Access 6 (2018), 64301–64320.

(Coulouris et al. 2011) G.Coulouris, J.Dollimore, T.Kindberg, G.Blair. Distributed Systems: Concepts and Design (5th. ed.). Addison-Wesley Publishing Company, USA. 2011.

(Critchley 2014) T.Critchley : High Availability IT Services (1st ed.). Auerbach Publications, (2014). https://doi.org/10.1201/b17958

(Crutcher et al. 2021) P.D.Crutcher, N.K.Singh, P.Tiegs, Essential Computer Science: A Programmer's Guide to Foundational Concepts, 1ère édition, Publisher Apress Berkeley, CA, (2021).

(Cui et al. 2017) H.CUI, K.ZHANG, Y.FANG, S.SOBOLEVSKY, C.RATTI, B.K.P.HORN, A Clustering Validity Index Based on Pairing Frequency, in *IEEE Access*, vol. 5, pp. 24884-24894, 2017, doi: 10.1109/ACCESS.2017.2743985.

(Davies et Bouldin 1979) D.L. Davies and D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1 (1979), 224–227.

(Dhote et Deshpande 2016) R.A. Dhote, S. P. Deshpande, Data Mining with Cloud Computing: - An Overview, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 5 Issue 1, January 2016.

(Dillon et Chang 2010) T.Dillon, C.Wu, E.Chang, Cloud Computing: Issues and Challenges, 24th IEEE International Conference on Advanced Information Networking and Applications, 2010.

(Djouhra 2016) D.Djouhra, Optimisation des Performances des Data Centres des Cloud sous Contrainte d'Energie Consommée, Université d'Oran 1, Ahmed Ben Bella, Thèse de Doctorat, 2016.

(Doudoux 2019) J.M.Doudoux. Développons en Java. 2019.

(Doyen 2005) G.Doyen, Supervision des réseaux et services pair à pair, thèse de dotorat, l'université Henri Poincaré – Nancy 1, 2005.

(Dudoit et al. 2002) S.Dudoit, J.Fridly, A prediction-based resampling method for estimating the number of clusters in a dataset. GenomeBiology, 3(7) : 0036. (2002), pp 1-21.

(Dunn et al. 2005) R.J. Dunn, J. Zahorjan, S.D. Gribble, H.M. Levy, Presence-based availability and P2P systems, in: 5th IEEE International Conference on Peer-to-Peer Computing, Konstanz, August 31–September 2, 2005, pp. 209–216.

(Durgadevi et Srinivasan 2016) P. Durgadevi and S. Srinivasan, Optimal resource discovery and dynamic resource allocation using modified hierarchal agglomerative clustering algorithm and bi-objective hybrid optimization algorithm, Asian Journal of Information Technology 15 (2016), 4464–4474.

(Durkee 2010) D. Durkee, Why Cloud Computing Will Never Be Free, Communications of the ACM, Volume 53 Issue 5, 2010, pp 62–69.
https://doi.org/10.1145/1735223.1735242

(Ebrahim et al. 2014) M.Ebrahim, S.Khan and S.Sherazand H.Mohani, Peer-to-Peer Network Simulators: an Analytical Review. arXiv preprint arXiv:1405.0400, 2014.

(Edwards 2002) J.Edwards, Peer-to-peer programming on Groove. Indianapolis, IN:Addison-Wesley, (2002).

(Endo et al 2016) P.T.Endo, M.Rodrigues, G.E. Gonçalves, J.Kelner, D.H. Sadok, C.Curescu, High availability in clouds: systematic review and research challenges. *J Cloud Comp* **5,** 16 (2016). https://doi.org/10.1186/s13677-016-0066-8

(Endrei et al 2004) M.Endrei, J.Ang, A.Arsanjani, S.Chua, P.Comte, P.Krogdahl, M.Luo,T.Newling, Patterns: Service-Oriented Architecture and Web Services, première édition, IBM redBook, (2004).

(Extensible Messaging and Presence Protocol, 2022) Extensible Messaging and Presence Protocol, 25 juillet 2022, Dans Wikipedia.
https://fr.wikipedia.org/wiki/Extensible_Messaging_and_Presence_Protocol

(Filali et al. 2011) I.Filali, F.Bongiovanni, F.Huet, F.Baude , A Survey of Structured P2P Systems for RDF Data Storage and Retrieval, Transactions on large-scale data-and knowledge-centered systems III, pp. 20–55. Springer-Verlag, Berlin, Heidelberg, 2011.

(Forgy 1965) E.W.Forgy: Cluster analysis of multivariate data: efficiency versus interpretability of classifications. In: Biometric Society Meeting, Riverside, California, 1965. Abstract in Biometrics, vol. 21, (1965), p. 768.

(Fraley et Raftery 1998) C. Fraley et A.E.Raftery, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", Technical Report No. 329. Department of Statistics University of Washington, 1998.

(Fremantle et al. 2002) P.Fremantle, S.Weerawarana, R.Khalaf, "Enterprise Services." Communications of the ACM, Vol. 45, No. 10, pp. 77- 82. (2002).

(Gangan 2015) S.Gangan : "A Review of Man-in-the-Middle Attacks." *ArXiv a*bs/1504.02115 (2015).

(Gao et al. 2012) G.Gao, R.Li, K.Wen, X.Gu, Proactive replication for rare objects in unstructured peer-to-peer networks, Journal of Network and Computer Applications, Volume 35, Issue 1, 2012, pp 85-96.

(Gartner 2022) (www.gartner.com)

(GR2841) Generic Requirements for Operations Systems Platform Reliability, Telcordia Technologies System Documentation, GR - 2841 - CORE, Issue 1, June 1994 .

(Grandini et al. 2020) M.Grandini, E.Bagli, G.Visani, METRICS FOR MULTI-CLASS CLASSIFICATION: AN OVERVIEW, A white paper, arXiv:2008.05756v1 [stat.ML] 13 Aug 2020.

(Grover 2014) N.Grover, A study of various Fuzzy Clustering Algorithms, International Journal of Engineering Research,Volume No.3, Issue No.3, (2014) pp : 177-181.

(Guha et al. 2006) S.Guha, N.Daswani, R.Jain, An experimental study of the Skype peer-to-peer VoIP system. In: Proceedings of the IPTPS'06. Santa Barbara, CA, Feb 2006.

(Gurugé 2004) A. Gurugé, Universal Description, Discovery, and Integration, Web Services, Digital Press, (2004), pp 145-188.

(Halkidi et al. 2002) M.Halkidi, Y.Batistakis, M.Vazirgiannis, Clustering Validity Checking Methods: Part II, ACM SIGMOD Record, Volume 31, Issue 3, September (2002), pp 19–27.
https://doi.org/10.1145/601858.601862

(Hamidouche et S.Hamidouche, T.Idjeraoui, Clustering : Approche par la théorie des jeux, thèse de Master,

Idjeraoui 2013)	Université Mira Abderrahmane de Béjaia, 2013.
(Hamze 2015)	M.Hamze, Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing, thèse de Doctorat, Université de Bourogne, 2015.
(Han et al. 2012)	J.Han, M.Kamber, J.Pei, Data Mining Concepts and Techniques, Third Edition, (3rd ed). Waltham, Mass.: Morgan Kaufmann Publishers, 2012.
(Han et Kamber 2001)	J.Han, M.Kamber, Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001.
(Hashemi et Bardsiri 2012)	S.M.Hashemi, A.K.Bardsiri, Cloud Computing Vs. Grid Computing, ARPN Journal of Systems and Software, VOL. 2, NO.5, (2012).
(Hassani et Seidl 2017)	M.Hassani, T.Seidl: Using internal evaluation measures to validate the quality of diverse stream clustering algorithms, Vietnam J Comput Sci (2017) 4:171–183.
(Hathaway et al. 1996)	J. Hathaway, J. Bezdek et J. Davenport., «On relational versions of c-means algorithm,» Pattern Recognition Letters, vol. 17, pp. 607-612, 1996.
(Hussain 2017)	S. Hussain, Survey on Current Trends and Techniques of Data Mining Research, London Journal of Research in Computer Science and Technology, Volume 17, Issue 1, 2017.
(Hussein et Mousa, 2014)	M. Hussein and M. Mousa, A light-weight data replication for cloud data centers environment, International Journal of Innovative Research in Computer and Communication Engineering 2 (2014), 2392–2400.
(Hurwitz et al. 2010)	J.Hurwitz, R.Bloor, M.Kaufman, F.Halper. Cloud computing for dummies. WILEY (2010).
(Höfer et Karagiannis 2011)	C.N. Höfer, G. Karagiannis, Cloud computing services: taxonomy and comparison, J Internet Serv Appl (2011) 2:81–94.
	DOI 10.1007/s13174-011-0027-x.
(Hosseini et Arani, 2015)	S.M.Hosseini, M.G.Arani: Fault-Tolerance Techniques in Cloud Storage: A Survey, International Journal of Database Theory and Application Vol.8, No.4 (2015), pp.183-190 http://dx.doi.org/10.14257/ijdta.2015.8.4.19
(Irani et al. 2016)	J.Irani, N.Pise, M.Phatak, Clustering Techniques and the Similarity Measures used in Clustering: A Survey, International Journal of Computer Applications (0975 – 8887, Volume 134 – No.7, January 2016.
(ISO/IEC 20000-1 2011)	ISO/IEC 20000-1:2011. Information Technology – Service Management – Part 1: Service Management System Requirements.
(ISO/IEC 17788 2014)	ISO/IEC 17788:2014 | ITU-T Recommendation Y.3500 (08/2014). Information technology – Cloud computing – Overview and vocabulary.
(ITILv3SD)	ITIL ® Service Design 2011 Edition, Cabinet Office, TSO, 2011, ISBN - 13: 978 - 0 - 11 - 331305 - 1.
(Jagli et Gupta 2013)	D. Jagli and A. Gupta, "Clustering Model for Evaluating SaaS on the Cloud", International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 12, December 2013.
(Jagli et al. 2014)	[63] D. Jagli, S. Mahajan, N. S. Chandra, “CBC Approach for Evaluating Potential SaaS on the Cloud”, International Technological Conference (I-TechCON), 2014.
(Jagli et al. 2017)	[64] D. Jagli, S. Mahajan, N. S. Chandra, "Comparative Clustering Approach Intended for Evaluating SaaS", in International Journal of Computer Applications, Volume 169 – No.8, July 2017.
(Jagli et al. 2018 (a))	D. Jagli, S. Mahajan and N. S. Chandra, "Implementation of Pam Cluster for Evaluating SaaS on the Cloud Computing Environment", Journal of Engineering, Vol. 08, Issue 4, PP 84-88, April 2018.

(Jagli et al. 2018 (b)) D.Jagli, S. Purohit , N.S. Chandra, "EM Clustering Model for Evaluating SaaS on Cloud Computing Environment", Journal of Computer Engineering, Volume 20, Issue 2, PP 67-72, 2018.

(Jagli et al. 2018 (c)) D.Jagli, S.Purohit, N.S.Chandra, "SAASQUAL: A Quality Model for Evaluating SaaS on the Cloud Computing Environment", In Big Data Analytics, pp. 429 -437. Springer, Singapore, 2018.

(Jaideep et Battula 2016) G.Jaideep, B.P.Battula, Survey on the Present State-of-the-Art of P2P Networks, Their Security Issues and Counter Measures, International Journal of Applied Engineering Research, Volume 11, Number 1 (2016), pp 616-620.

(Jain et Srivastava 2014) N. Jain and V. Srivastava, " Data Mining Techniques: A Survey Paper", International Journal of Research in Engineering and Technology, Volume: 02, Issue: 11, (2014).

(JALOTE 1994) P.JALOTE : Fault Tolerance in Distributed Systems, Prentice Hall these doctorate, 1994.

(Jathanna et Jagli 2017) R.Jathanna, D.Jagli, Cloud Computing and Security Issues, Rohan Jathanna. Int. Journal of Engineering Research and Application, Vol. 7, Issue 6, (Part -5) (2017), pp.31-38.

(Jesi 2005) G.P.Jesi, PeerSim HOWTO:Build a new protocol for the PeerSim 1.0 simulator, December 24, 2005.

(Jhawar et Piuri 2013) R. Jhawar and V. Piuri, Adaptive Resource Management for Balancing Availability and Performance in Cloud Computing, in: The 10th International Conference on Security and Cryptography, Reykjavik, Iceland, July 29–31, 2013, pp. 254–264.

(Jo et Han 2018) S.Jo, J.Han, Convergence P2P Cloud Computing, Peer-to-Peer Networking and Applications (2018) 11:1153–1155

https://doi.org/10.1007/s12083-018-0661-1

(Jouini et Rabai 2015) M.Jouini, L.B.A.Rabai, Design Challenges of Cloud Computing, IGI Global, 2015.

(Jourjon et El Baz 2005) G.Jourjon, D.El Baz, Some Solutions For Peer to Peer Global Computing, the 13th Euromicro Conference on Parallel, Distributed and Network-Based Processing (Euromicro-PDP'05), 2005.

(Kabilan et Jayaveeran 2015) R.Kabilan, N.Jayaveeran : SURVEY OF DATA MINING TECHNIQUES IN CLOUD COMPUTING, International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-8, November 2015.

(Kahanwal et Singh 2012) B.Kahanwal, T.P.Singh, The Distributed Computing Paradigms: P2P, Grid, Cluster, Cloud, and Jungle, International Journal of Latest Research in Science and Technology, Vol.1, Issue 2: Page No.183-187 , (2012).

(Kamalraj et al. 2012) R. Kamalraj, A.R. Kannan, S.Vaishnavi and V. Suganya, "A DataMining BasedApproach for Introducing Products in SaaS (Software as a Service)", International Journal of Engineering Innovation & Research, Volume 1, Issue 2, 2012.

(Kamel et Selim 1991) M.S. Kamel, S.Z. Selim, A thresholded fuzzy c-means algorithm for semi-fuzzy clustering, Pattern Recognition, Volume 24, Issue 9, 1991, pp 825-833

(Kamvar et al. 2003) S.D.Kamvar, M.T.Schlosser, H.Garcia-Molina, Incentives for Combatting Freeriding on P2P Networks, In: Kosch, H., Böszörményi, L., Hellwagner, H. (eds) Euro-Par 2003 Parallel Processing. Euro-Par 2003. Lecture Notes in Computer Science, vol 2790. Springer, Berlin, Heidelberg (2003). https://doi.org/10.1007/978-3-540-45209-6_171

(Kanagalakshmi et Gnanaselvam 2017) V. Kanagalakshmi V and R. Gnanaselvam , "Review of Clustering Technique using SaaS on the Cloud", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Volume 2, Issue 4, 2017.

(Karahoca et al. A.Karahoca, D.Karahoca, M.Şanver : Survey of Data Mining and Applications (Review

2012) from 1996 to Now), (2012).

(Karimi et al. 2008 (a)) O.B. Karimi, S. Yousefi, M. Fathy and M. Mazoochi, Availability in Peer to Peer Management Networks, Challenges for Next Generation Network Operations and Service Management Conference, in: 11th Asia-Pacific Network Operations and Management Symposium, Beijing, China, October 22–24, 2008, pp. 552–555.

(Karimi et al. 2008 (b)) O.B. Karimi, S. Yousefi, M. Fathy, M. Mazoochi, Availability measurement in Peer to Peer Network Management Systems, in: Third IEEE International Conference on Digital Information Management, (ICDIM), London, UK, November 13–16, 2008, pp. 745–750.

(Kaushal et al. 2015) C.Kaushal, A.Arya, S.Pathania : Integration of Data Mining in Cloud Computing, Advances in Computer Science and Information Technology (ACSIT), Volume 2, Number 7; April – June, 2015 pp 48 – 52.

(Kaur 2015) B. Kaur, Software as a service: A brief study, International Research Journal of Engineering and Technology 2 (2015), 789–792.

(Kaur et al. 2012) S.Kaur, K.Kaur, D.Singh. A framework for hosting web services in cloud computing environment with high availability, in: IEEE International Conference In Engineering Education: Innovative Practices and Future Trends (AICERA). (2012) pp. 1–6.

(Kaur et Singh, 2017) S.Kaur, G.Singh : Review on Fault Tolerance Techniques in Cloud Computing, International Journal of Engineering and Management Research. Volume-7, Issue-3, May-June 2017, 69-71.

(Kautkar 2014) R.A. Kautkar, "A Comprehensive Survey on Data Mining", International Journal of Research in Engineering and Technology, Volume: 03, Issue: 08, August-2014.

(Kavalionak et Montresor 2012) H.Kavalionak, A.Montresor, P2P and Cloud: A Marriage of Convenience for Replica Management, IWSOS, (2012).

(Kaya 2005) M. Kaya, An algorithm for image clustering and compression, Turkish Journal of Electrical Engineering and Computer Sciences 13 (2005), 79–91.

(Khanghahi et Ravanmehr 2013) N.Khanghahi, R.Ravanmehr, Cloud Computing Performance Evaluation: Issues and Challenges, International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol.3, No.5, October 2013.

(Khethavath 2014) P.Khethavath, Towards an Efficient Distributed Cloud Architecture, Oklahoma State University, thèse de Doctorat, (2014).

(Kisembe et Jeberson 2017) P.Kisembe, W.Jeberson, Future of Peer-to-Peer Technology with the Rise of Cloud Computing, International Journal of Peer to Peer Networks (IJP2P) Vol.8, No.2/3, (2017).

(Kokate et al. 2018) U.Kokate, Arvind V. Deshpande, Parikshit N. Mahalle and Pramod Patil. “Data Stream Clustering Techniques, Applications, and Models: Comparative Analysis and Discussion.” Big Data and Cognitive Computing. 2 (2018): 32.

(Koutrouli et Tsalgatidou 2006) E. Koutrouli and A. Tsalgatidou, Reputation-based Trust Systems for P2P Applications: Design Issues and Comparison Framework, in: Third International Conference onTrust and Privacy in Digital Business, Kraków, Poland, September 4–8, 2006, pp. 152–161.

(Kumar et Ramachandhra 2014) V.Kumar, G.A.Ramachandhra, Optimization of Large Data in Cloud computing using Replication Methods, International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 3034-3038.

(Kumar et SubbaRao 2019) D.G.Kumar, C.D.V.SubbaRao, Peer–To–Peer Computing: Architectures, Applications and Challenges, International Journal of Recent Technology and Engineering (IJRTE), Volume-8, Issue-1S4, June 2019.

(Li 2008) J.Li. “A Survey of Peer-to-Peer Network Security Issues.” (2008).

(Liu et al. 2016) K. Liu, J. Boehm and C. Alis, Change Detection of Mobile Lidar Data Using Cloud Computing, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Prague, Czech Republic, July 12–19, 2016, 309–313.

(López-Fuentes et al. 2012) F.A.López-Fuentes, I.Eugui-De-Alba, O.M.Ortíz-Ruiz, Evaluating P2P Networks against Eclipse Attacks, Procedia Technology, Volume 3, 2012, pp 61-68.

(Lu et al. 2010) W. Lu, Y. Zhou, J. Li and B. Yan, A Hierarchical Data Replication Method in ad hoc networks, in: 2 nd International Conference on Computer Engineering and Technology, Bali Island, Indonesia, Mars 26–29, 2010, pp. 23–27.

(Lu et al. 2013) Q.Lu, X.Xu, L.Zhu, L.Bass, Z.Li, S.Sakr, P.Bannerman : Incorporating Uncertainty into in-Cloud Application Deployment Decisions for Availability, in: IEEE International Conference on Cloud Computing. (2013) pp. 1–8.

(Maaref 2012) S.Maaref, Cloud Computing en Afrique Situation et perspectives, Avril 2012.

(MacQueen 1967) J.MacQueen : Some methods for classification and analysis of multivariate observations, (1967).

(Makhsous et al. 2016) S.H. Makhsous, A. Gulenko, O. Kao and F. Liu, High Available Deployment of Cloud-Based Virtualized Network Functions, in: International Conference on High Performance Computing and Simulation (HPCS), Innsbruck, Austria, July 18–22, 2016, pp. 468–475.

(Malathy at al. 2013) G. Malathy, R. Somasundaram, K. Duraiswamy, Performance improvement in cloud computing using resource clustering, Journal of Computer Science 9 (2013), 671–677.

(Marcus et Stern 2003) E.Marcus et H.Stern : Blueprints for High Availability (Second Edition). Wiley Publishing, (2003).

(Maurya et al. 2016) R.K.Maurya, S.Pandey, V.Kumar, A Survey of Peer- to-Peer Networks, International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 4, April 2016.

(Mayer et al. 2013) P.Mayer, A.Klarl, R.Hennicker, M.Puviani, F.Tiezzi, R.Pugliese, J.Keznikl, T.Bures, The Autonomic Cloud: A Vision of Voluntary, Peer-2-Peer Cloud Computing, IEEE 7th International Conference on Self-Adaptation and Self-Organizing Systems Workshops, (2013), pp. 89-94, doi: 10.1109/SASOW.2013.16.

(Meenakshi et al. 2019) A. Meenakshi, H. Sirmathi and J.A. Ruth, Cloud computing-based resource provisioning using k-means clustering and GWO prioritization, Soft Computing 23 (2019), 10781–10791.

(Mehta 2017) C. Mehta, "Basics of Data Mining: A Survey Paper", International Journal of Trend in Research and Development, Volume 4, 2017.

(Mehta et al. 2006) M.R.Mehta, S.Lee, J.R.Shah, Service-Oriented Architecture:Concepts and Implementation, (2006).

(Melo et al. 2018) R.Melo, V.P.F.M.Sobrinho, I.J.M.Filho, F.Feliciano, P.R.M.Maciel, Redundancy Mechanisms Applied in Video Streaming Service on Eucalyptus Cloud Computing Infrastructures. Journal of Information Systems Engineering & Management, 3(3), 25, (2018). https://doi.org/10.20897/jisem/2663

(Mengistu et Che 2019) T.M. Mengistu, D.Che, Survey and Taxonomy of Volunteer Computing, ACM Comput, (2019).

(Mesbahi et al. 2018) M.R. Mesbahi, A.M. Rahmani, M. Hosseinzadeh : Reliability and high availability in cloud computing environments: a reference roadmap. Hum. Cent. Comput. Inf. Sci. 8, 20 (2018). https://doi.org/10.1186/s13673-018-0143-8

(Meslem et Debbas 2017) Y.Meslem, S.I Debbas: Ordonnancement et réplication de données dans le Cloud Computing, Memoire de MASTER option: Réseau Informatique et Système Répartis, Université Dr. Tahar Moulay SAIDA, (2017).

(Milani et B.A.Milani, N.J.Navimipour : A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions, Journal of Network and

Navimipour, 2016) Computer Applications, Volume 64, 2016, 229-238,

https://doi.org/10.1016/j.jnca.2016.02.005.

(Miller 2001) J.Miller, Jabber – Conversational technologies. In A. Oram (Ed.), Peer-to-peer: Harnessing the benefits of a disruptive technology (pp. 77–88). Sebastopol, CA: O'Reilly. (2001)

(Milogicic et al. 2002) D.S.Milogicic, V.Kalogeraki, R.Lukose, K.Nagaraja, J.Pruyne, B.Richard, S.Rollins, Z.Xu, Peer-to-Peer Computing, (2002).

(Miyamoto 1998) S. Miyamoto, "An overview and new methods in fuzzy clustering," Second International Conference. Knowledge-Based Intelligent Electronic Systems. Proceedings KES'98 (Cat. No.98EX111), 1998, pp. 33-40 vol.1, doi: 10.1109/KES.1998.725825.

(Mukherjee et al 2015) S.Mukherjee, R.Shaw, N.Haldar, S.Changdar : A Survey of Data Mining Applications and Techniques, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 6 (5), (2015), 4663-4666.

(Nabi et al. 2016) M.Nabi, M.Toeroe, F.Khendek, Availability in the Cloud: State of the Art, Journal of Network and Computer Applications Volume 60, 2016, 54-67

(Naeem et al. 2016) M.M.Naeem, H.M.F.Memon, M.Siddique, A.Rauf, An Overview of Virtualization and Cloud Computing, Sci.Int.(Lahore), 28(4), 3799-3803 ,(2016).

(Naicken et al. 2006) S.Naicken, A.Basu, B.Livingston and S.Rodhetbhai, A Survey of Peer-to-Peer Network Simulators, In The Seventh Annual Postgraduate Symposium, vol. 2, 2006.

(Navimipour et Milani 2015) N.J.Navimipour, F.S.Milani, A comprehensive study of the resource discovery techniques in Peer-to-Peer networks, Peer-to-Peer Netw. Appl. (2015) 8:474–492.

DOI 10.1007/s12083-014-0271-5

(Neamtiu et Dumitras 2011) I.Neamtiu, T.Dumitras, Cloud software upgrades: Challenges and opportunities, in: International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems. IEEE, (2011) pp. 1–10. doi:10.1109/MESOCA.2011.6049037

(Netes 2018) V. Netes, End-to-End Availability of Cloud Services, in: The 22nd Conference of Open Innovations of Association, Jyvaskyla, Finland, May 15–18, 2018, pp. 198–203.

(Nguyen 2011) T.T.Nguyen, Un environnement pour le calcul intensif pair à pair, thèse de Doctorat, Université de Toulouse, (2011).

(Noraziah et all. 2013) A. Noraziah, A. Fairuzullah, C.F. Azila, A. Azuan, N. Aqtar, Z. Azlina and T. Herawan, Fault Tolerance On Binary Vote Assignment Cloud Quorum Replication Technique, in: International Conference on Computer Science and Information Technology, Yogyakarta, Indonesia, June 16–18, 2013, pp. 2–6.

(Oh et Kim 2019) Y. Oh, Y. Kim, A resource recommendation method based on dynamic cluster analysis of application characteristics, Cluster Computing 22 (2019), 175–184.

(OPARA 2019) C.D.OPARA, Cloud Computing in Amazon Web Services, Microsoft Windows Azure, Google App Engine and IBM Cloud Platforms: A Comparative Study, thèse Master, 2019.

(Ouaarab et al. 2014) A.Ouaarab, B.Ahiod, X.Yang, Discrete cuckoo search algorithm for the travelling salesman Problem, Neural Comput & Applic (2014) 24:1659–1669

DOI 10.1007/s00521-013-1402-2

(Oram 2001) A. Oram, Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2001.

(P2P VoIP 2018) P2P VoIP, (27 avril 2018), Dans Wikipedia

https://fr.wikipedia.org/wiki/P2P_VoIP

(Palacio-Niño et J.Palacio-Niño, F.Berzal, Evaluation Metrics for Unsupervised Learning Algorithms,

Berzal 2019) arXiv:1905.05667v2 [cs.LG] 23 May 2019.

(Panimalar et al. 2017) S.A.Panimalar, R.Priyadharshan.R, R.M.Kumar.R, G.Visweshwaran, Salesforce.com – A Cloud Provider, International Research Journal of Engineering and Technology (IRJET), Volume: 04 Issue: 09, (2017).

(Panzieri et al. 2011) F.Panzieri, O.Babaoglu, S.Ferretti, V.Ghini, M.Marzolla, Distributed Computing in the 21st Century: Some Aspects of Cloud Computing, Lecture Notes in Computer Science · January 2011.

(Parekh 2014) M.h.Parekh : Enhancement Clustering of Cloud Datasets using Improved Agglomerative Technique, International Journal of Advanced Networking Applications (IJANA), (2014).

(Park et al 2003) S. Park, J. Kim, Y. Ko and W. Yoon, Dynamic Data Grid Replication Strategy Based on Internet Hierarchy, in: International Conference on Grid and Cooperative Computing, Wuhan, China, October 21–24, 2003, pp. 383–846.

(Patel et Yadav 2016) J.Patel, O.P.Yadav, Comparison of Fuzzy C-Means and Hierarchical Agglomerative Clustering Algorithms for Data Mining, International Journal for Scientific Research & Development| Vol. 4, Issue 03, (2016).

(Pavlik et al. 2014) J. Pavlik, V. Sobeslav and J. Horalek, Statistics and Analysis of Service Availability in Cloud Computing, in: 18th International Database Engineering and Applications Symposium, Porto Portugal, July 7–9, 2014, pp. 310–313.

(Petre 2012) R.Ş.Petre, Data mining in Cloud Computing, *Database Systems Journal* vol. 3, no. 3, (2012).

(Pourebrahimi et al. 2005) B.Pourebrahimi, K.Bertels, S.Vassiliadis, A Survey of Peer-to-Peer Networks, (2005).

(Prasanth 2012) A.Prasanth, Cloud Computing Services: A Survey, International Journal of Computer Applications, Volume 46– No.3, 2012).

(Puthal et al. 2015) D.Puthal, B. P. S. Sahooy, S.Mishraz, S.Swain, Cloud Computing Features, Issues and Challenges: A Big Picture, International Conference on Computational Intelligence & Networks, (2015).

(Rahmati et Rahmani 2017) B. Rahmati and A.M. Rahmani, Data replication-based scheduling in cloud computing environment, Journal of Advances in Computer Engineering and Technology 3 (2017), 75–80.

(Rao et Selvamani 2015) R.V.Rao, K.Selvamani, Data Security Challenges and Its Solutions in Cloud Computing, International Conference on Intelligent Computing, Communication & Convergence, (2015).

(Rama et Reddy, 2013) A. Rama and M. Reddy, Data replication system in cloud based on data mining techniques, International Journal of Advanced Research in Computer and Communication Engineering 2 (2013), 4216–4221.

(Rauscher et al. 2006) K.F. Rauscher, E.K. Richard, P.R. James: Eight ingredients of communications infrastructure: A systematic and comprehensive framework for enhancing network reliability and security. *Bell Labs Technical Journal*, 11, (2006), 73-81.

(Rendón et al. 2011) E.Rendón, I.Abundez, A.Arizmendi, E.M.Quiroz, Internal versus External cluster validation Indexes, International Journal of Computers and Communications, Issue 1, Volume 5, (2011).

(Rodrigues et Druschel 2010) R.Rodrigues, P.Druschel, Peer-to-peer systems, Communications of the ACM, Volume 53, Issue 10, (2010), pp 72-82. https://doi.org/10.1145/1831407.1831427

(Rokach et Maimon 2005) L.Rokach, O.Maimon, Clustering Methods. In (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA (2005). https://doi.org/10.1007/0-387-25465-X_15

(Rose 2001) M. Rose. The Blocks Extensible Exchange Protocol Core. Internet Engineering Task Force, 2001. RFC 3080 (Proposed Standard).

(Rose et Klyne 2002) M. Rose, G. Klyne, D. Crocker. The Application Exchange (APEX) Access Service. Internet Engineering Task Force, July 2002. RFC 3341 (Proposed Standard).

(Saadat et Rahmani 2012) N. Saadat and A. M. Rahmani, PDDRA: A new pre-fetching based dynamic data replication algorithm in data grids, Future Generation Computer Systems 28 (2012), 666–681.

(Sajid et Raza 2013) M.Sajid, Z.Raza, Cloud Computing: Issues & Challenges, International Conference on Cloud, Big Data and Trust (2013), Nov 13-15.

(Sandanayake et Jayangani 2018) T.C. Sandanayake, P.G.C.Jayangani, Current Trends in Software as a Service (SaaS), International Journal for Innovation Education and Research, Vol:-6 No-02, (2018).

(Saravanakumar et al. 2021) C. Saravanakumar, M. Geetha, S.M. Kumar, S. Manikandan, C. Arun and K. Srivatsan, An Efficient Technique for Virtual Machine Clustering and Communications Using Task-Based Scheduling in Cloud Computing, Scientific Programming, 2021, 1–15.

(Sashi et Thanamani, 2010) K. Sashi and A.S. Thanamani, Dynamic replica management for data grid, International Journal of Engineering and Technology 2 (2010), 329–333.

(Sashi et Thanamani 2011) K. Sashi and A.S. Thanamani, Dynamic replication in a data grid using a modified BHR region based algorithm, Future Generation Computer Systems 27 (2011), 202–210.

(Schoder et al. 2005) D.Schoder, K.Fischbach, C.Schmitt, “Chapter I Core Concepts in Peer-to-Peer Networking, (2005).

(Schollmeier 2001) R. Schollmeier, "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications," *Proceedings First International Conference on Peer-to-Peer Computing*, 2001, pp. 101-102, doi: 10.1109/P2P.2001.990434.

*(*Schulte et Natis 1996) R. W. Schulte, Y. V. Natis, “"Service oriented'architectures, part 1, ” Gartner, SSA Research Note SPA-401-068, 1996.

(Selim et Ismail 1984) S.Z. Selim, M.A. Ismail, Soft clustering of multidimensional data: a semi-fuzzy approach, Pattern Recognition, Volume 17, Issue 5, (1984), pp 559-568.

(Shahapure et Jayarekha 2015) N.H. Shahapure and P. Jayarekha, Replication: A technique for scalability in cloud computing, International Journal of Computer Applications 122 (2015), 8875–8887.

(Shahriar et al. 2011) N. Shahriar, M. Sharmin, R. Ahmed, M. Rahman, R. Boutaba and B. Mathieu, Diurnal Availability for Peer-to-Peer Systems, in: The 9th Annual IEEE Consumer Communications and Networking Conference, Las Vegas, USA, January 14–17, 2011, pp. 619–623.

(Shahzad 2014) F.Shahzad, State-of-the-art Survey on Cloud Computing Security Challenges, Approaches and Solutions, The 6th International Symposium on Applications of Ad hoc and Sensor Networks (AASNET’14), (2014) 357 – 362.

(Sharma et al. 2014) A.Sharma, R.Sharma,V.K. harma,V.Shrivatava : Application of Data Mining – A Survey Paper, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), (2014), 2023-2025.

(Shirkhorshidi et al. 2015) A.S.Shirkhorshidi, S.Aghabozorgi, T.Y.Wah, A Comparison Study on Similarity and Dissimilarity Measures in Clustering Continuous Data, PLoS ONE 10(12): e0144059, (2015). doi:10.1371/journal.pone.0144059

(Shredeh 2016) K.Shredeh, Analysis of Attacks and Security Issues on the Peer-to-Peer Networks, International Journal of Computer Applications (0975 – 8887) Volume 138 – No.2, March 2016.

(Shree et Basha 2014) S.U. Shree and M.S.S. Basha, An exaustive survey of trust models in P2P network, International Journal on Web Service Computing 5 (2014), 1–12.

(Siebenhaar et al. M.Siebenhaar, O.Wenge, R.Hans, H.Tercan, R.Steinmetz, Verifying the Availability of

2013) Cloud Applications, the 3rd International Conference on Cloud Computing and Services Science, (2013), pp -494.

(Singh 2001) M. P. Singh. "Peering at Peer-to-Peer Computing". IEEE Internet Computing, Vol. 5, No. 1. January/February 2001.

(Singh et al. 2012) D. Singh, J. Singh and A. Chhabra, High Availability of Clouds: Failover Strategies for Cloud Computing using Integrated Checkpointing Algorithms, in: International Conference on Communication Systems and Network Technologies, Rajkot, India, May 11–13, 2012, pp. 698–703.

(Singh et al. 2018) B.Singh, G.Singh, A Study on Virtualization and Hypervisor in Cloud Computing, International Journal of Computer Science and Mobile Applications, Vol.6 Issue. 1, January- 2018, pg. 17-22.

(Singhal et al. 2007) A.Singhal, T.Winograd, K.Scarfone, Guide to Secure Web Services, Recommendations of the National Institute of Standards and Technology, Special Publication 800-95 Natl. Inst. Stand. Technol. Spec. Publ. 800-95, 128 pages (Aug. 2007).

(Sood et al. 2016) D.Sood, H.Kour, S.Kumar, Survey of Computing Technologies: Distributed, Utility, Cluster, Grid and Cloud Computing, Journal of Network Communications and Emerging Technologies (JNCET) www.jncet.org Volume 6, Issue 5, May (2016).

(Sun et al. 2012) D.W. Sun, G.R. Chang, S. Gao, L.Z. Jin and X.W. Wang, Modeling a dynamic data replication strategy to increase system availability in cloud computing environments, Journal of Computer Science and Technology 27 (2012), 256–272.

(Srati et al. 2017) S.Srati, D.C. Jinwala and S.Garg, A survey of simulators for P2P overlay networks with a case studyof the P2P tree overlay using an event-driven simulator, Engineering Science and Technology, an International Journal,Volume 20, Issue 2, April 2017, Pages 705-720.

(Steinmetz et Wehrle 2005) R.Steinmetz, K.Wehrle (Eds.), Peer-to-Peer Systems and Applications, 2005.

(Stencil 2002) Stencil, "The Laws of Evolution: A Pragmatic Analysis of the Emerging Web Services Market. " The Stencil Group: http://www.stencilgroup.com. (2002).

(Stutzbach et Rejaie 2005) D. Stutzbach, R. Rejaie. Characterizing Today's Gnutella Topology. Proceedings of the ACM SIGMETRICS, Poster Session, Alberta Canada, June 2005.

(Subbareddy et Begam 2021) R.Subbareddy, B.F.Begam, A Study on hypervisor virtualization in cloud computing, International Journal of Scientific & Engineering Research Volume 12, Issue 2, February-2021.

(Sungkar et Kogoya 2020) A.Sungkar, T.Kogoya, A REVIEW OF GRID COMPUTING, Computer Science & IT Research Journal, Volume 1, Issue 1, P.1-6, March, 2020.

(Surianarayanan et Chelliah 2019) C.Surianarayanan, P.R.Chelliah, Essentials of Cloud Computing A Holistic Perspective, 1ère édition, 2019.

(Syed et Baig 2013) H.I. Syed, N.A.Baig, Survey On Cloud Computing, International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 4, April 2013.

(Tabet et Ziani 2013) A.W.H. Tabet, S.Ziani, Clustering Hiérarchique de données à base de Ward, mémoire Licence, Université Abou Bakr Belkaid– Tlemcen, (2013).

(Talia et Trunfio 2010) D. Talia and P. Trunfio, "How Distributed Data Mining Tasks can Thrive as Knowledge Services", Communications of the ACM, vol. 53, n. 7, pp. 132 -137, (2010).

(Tanenbaum et Steen 2006) A. S.Tanenbaum, M.V.Steen. Distributed Systems: Principles and Paradigms (2nd Edition). Prentice-Hall, Inc., USA. (2006).

(Tchao et al. 2021) E.T.Tchao, D.A.Quansah, G.S.Klogo, F.Boafo-Effah, S.Kotei, C.Nartey, W.K.Ofosu, On cloud-based systems and distributed platforms for smart grid integration: Challenges and prospects for Ghana's Grid Network, Scientific African 12 (2021).

(Thampi 2009) S. M. Thampi, Introduction to Distributed Systems, ArXiv, abs/0911.4395. (2009)

(Thein et Park, 2009) T. Thein and J.S. Park, Availability analysis of application servers using software rejuvenation and virtualization, Journal of Computer Science and Technology 24 (2009), 339–346.

(TL9000) TL 9000 Quality Management System Measurements Handbook 4.5, Quality Excellence for Suppliers of Telecommunications Forum (QuEST Forum), 2010, http://tl9000.org.

(Tlili 2011) M.Tlili Infrastructure P2P pour la Réplication et la Réconciliation des Données, Thèse de Doctorat, Université de Nantes, 2011.

(Toeroe et Tam 2012) M.Toeroe, F. Tam : Service availability principles and practice. John Wiley and Sons Ltd publication, 2012.

(Toosi et al. 2014) A.N.Toosi, R.N.Calheiros, R.Buyya. Interconnected Cloud Computing Environments : Challenges, Taxonomy, and Survey. ACM computing Surveys 47, (2014) 1–47.

(Trifa et Khemakhem 2014) Z.Trifa, M.Khemakhem, Effects of Churn on Structured P2P Overlay Networks, International Conference on Automation, Control, Engineering and Computer Science (ACECS'14) Proceedings - Copyright IPCO-2014, pp.164-170.

(Tseng et al., 2005) G. C. Tseng, A.Thalamuthu, I.Mukhopadhyay, X.Zheng, Evaluation and comparison of gene clustering methods in microarray analysis. Bioinformatics, 22(19):2405-12, (2005).

(Undheim et al. 2011) A.Undheim, A.Chilwan, P.Heegaard. Differentiated Availability in Cloud Computing SLAs, in: IEEE/ACM 12th International Conference on Grid Computing (GRID '11). IEEE, (2011) pp. 129–136.

(Vani et Priya 2015) B.Vani, R.C.M.Priya, Availability in Cloud Computing, International Journal of Innovative Research in Information Security (IJIRIS), Issue 2, Volume 4 (2015).

(Vanneau et al. 2005) P.Vanneau, M.Samson, J.Donel, JINI – SOAP : Présentation, World Wide Web Consortium (W3C), (2005).

(Vargas 2000) E.Vargas : High Availability Fundamentals, Enterprise Engineering Sun BluePrints/Sun BluePrints™ OnLine. Novembre 2000.

(Varljen 2017) J.Varljen : Scalability and High Availability in Real-time Cloud Services, MASTER'S THESIS, University of Ljubljana, (2017).

(Vijay et Reddy 2013) G. R.Vijay, A.R.M.Reddy: Data Replication System in Cloud based on Data Mining Techniques, International Journal of Advanced Research in Computer and Communication Engineering, Vol. 2, Issue 11, (2013).

(Vijaya et Ramachandhra 2014) C.K. Vijaya and G.A. Ramachandhra, Optimization of large data in cloud computing using replication methods, International Journal of Computer Science and Information Technologies 5 (2014), 3034–3038.

(Visalakshi et Kuttiyannan 2009) N.K. Visalakshi and T. Kuttiyannan, Impact of normalization in distributed k-means clustering, International Journal of Soft Computing 4 (2009), 168–172.

(Wang et Li 2003) C.Wang, B.Li, Peer-to-Peer Overlay Networks: A Survey, (2003).

(Wang et Zhang 2007) W. Wang and Y. Zhang, On fuzzy cluster validity indices, Fuzzy Sets and Systems 158 (2007), 2095–2117.

(Washbourne 2015) L.Washbourne, A Survey of P2P Network Security, 2015.

(Wei et al. 2010) Q. Wei, B. Veeravalli, B. Gong, L. Zeng and D. Feng, CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster, in: 2010 IEEE International Conference on Cluster Computing, Heraklion, Greece, September 20–24, 2010, pp. 188–196.

(Wu et al. 2014) Q. Wu, M. Cui, M. Zhang, R. Zheng and Y. Lou, A cloud service resource classification strategy based on feature similarity, Journal of Networks 9 (2014), 2987–2993.

(Xing et Zhan 2012) Y.Xing, Y.Zhan. Virtualization and Cloud Computing. In: Zhang, Y. (eds) Future Wireless Networks and Information Systems. Lecture Notes in Electrical Engineering, vol 143. (2012). Springer, Berlin, Heidelberg.

https://doi.org/10.1007/978-3-642-27323-0_39

(Xiong 2018) J.Xiong, Cloud Computing for Scientific Research, 1ère édition, 2018.

(Xue et Xin 2016) C.T.S.Xue, F.T.W.Xin, BENEFITS AND CHALLENGES OF THE ADOPTION OF CLOUD COMPUTING IN BUSINESS, International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 6, No. 6, December 2016.

(Yanovsky et al. 2016) M.Yanovsky, O.Yanovskaya, V.Kharchenko, Analysis of Methods for Providing Availability and Accessibility of Cloud Services, 12th International Conference on ICT in Education, Research and Industrial Applications, (2016).

(Yeferny 2014) T.Yeferny, Proposition d'approches de routage de requêtes dans les systèmes pair-à-pair non structurés, Thèse de Doctorat, Institut National des Télécommunications, 2014.

(Yeo et al. 2007) C.S.Yeo, M.Dias .Assunção, J.Yu, A.Sulistio, Utility Computing on Global Grids. In Handbook of Computer Networks, H. Bidgoli (Ed.).(2007).

https://doi.org/10.1002/9781118256107.ch8

(Ying et al. 2016) R.Ying, L.Hong, L.Hua-wei, Z.Li-jun , W.Li-na, Data Mining Based on Cloud-Computing Technology, MATEC Web of Conferences 61, 07015 (2016).

(Yoshida et al. 2003) S. Yoshida, K. Kamei, T. Ohguro, and K. Kuwabara. Shine : a peer-to-peer based framework of network community support systems. Computer Communications, 26 :1199–1209, August 2003.

(Yusoh et Tang 2010) Z.I.M.Yusoh, M.Tang, "A Penalty- based Genetic Algorithm for the Composite SaaS Placement Problem in the Cloud", IEEE World Congress on Computational Intelligence, Barcelona, pp. 1-8, July 2010.

(Yusoh et Tang 2012 (a)) Z.I.M.Yusoh, M.Tang, "Clustering composite SaaS components in Cloud Computing using a Grouping Genetic Algorithm," IEEE Congress on Evolutionary Computation, Brisbane, QLD, pp. 1-8, 2012.

(Yusoh et Tang 2012 (b)) Z.I.M.Yusoh and M.Tang, "A penalty-based grouping genetic algorithm for multiple composite SaaS components clustering in Cloud,"2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, pp. 1396-1401, 2012.

(Zaki et Meira 2020) M.J. Zaki, W.Meira, Data Mining and Machine Learning: Fundamental Concepts and Algorithms, Second Edition, Cambridge University Press, March 2020.

(Zayed et al. 2015) A.H.Zayed, H.Mostafa, M.Abdelaziz, Cloud Computing et Sécurité : Approches et Solutions, International Journal of Computer Trends and Technology (IJCTT) – volume 30, (2015).

(Zeng 2018) J. ZENG : The development and application of Data Mining based on Cloud Computing", First International Conference on Advanced Algorithms and Control Engineering, IOP Publishing (2018).

(Zhang 2021) N.Zhang, An Overview of Advantages and Security Challenges of Cloud Computing, International Journal of Computer Science and Mobile Computing, Vol. 10, Issue. 1, January 2021.

(Zhang et al. 1996) T.Zhang, R.Ramakrishnan, R. and M.Livny, BIRCH: an efficient data Clustering method for very large databases. In Proceedings of the ACM SIG MOD Conference, 103-114, Montreal, Canada. 1996.

(Zhang et al. 2010) Q.Zhang, L.Cheng, R.Boutaba, Cloud computing: state-of-the-art and research challenges, J Internet Serv Appl (2010) 1: 7–18

DOI 10.1007/s13174-010-0007-6.

(Zhang et al. 2014) Q. Zhang, M.F. Zhani, M. Jabri and R. Boutaba, Venice: Reliable Virtual Data Center Embedding in Clouds, in: IEEE Conference on Computer Communications, Toronto, Canada, April 27–Mai 2, 2014, pp. 289–297.