Université Dr MOULAY TAHER SAIDA

Mémoire de Fin d'Etude

Pour l'Obtention
du Diplôme de Master en Informatique

Présenté par :
AYADI Ahmed Yazid

Domaine : Mathématiques Et Informatique

Spécialité : Réseaux Informatique et
Systèmes Répartis (RISR)

Une stratégie d'ordonnancement et de réplication des données dans les Fogs Computing

---

Encadré par :
Mr Limam S.

DÉDICACE

Je dédie ce travail à :

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études, A toute ma famille pour leur soutien tout au long de mon parcours universitaire, Que ce travail soit l'accomplissement de vos vœux tant allégués, et le fruit de votre soutien infaillible, J'attire un grand remercie à Monsieur LIMAM Said pour sa sacrifice avec moi durant la réalisation de cette thèse.

Remerciements

Je remercie, en premier lieu, ALLAH, le tout puissant, de m'avoir permis et accorder la volonté, la patience et le courage pour réaliser ce travail.

Je tiens à exprimer à remercier à l'encadreur Dr LIMAM Said pour me avoir donner l'opportunité de réaliser ce sujet sous sa direction, la confiance faite ainsi que ses conseils fructueux, et son temps consacré tout au long du travail.

Résumé

Dans les années à venir, l'Internet des objets (IoT) sera l'une des applications de génération de données les plus populaires. En fait, plus de 150 milliards d'objets seront connectés d'ici 2025. Les données IoT peuvent être traitées et utilisées par divers appareils répartis sur le réseau. Le modèle centralisé traditionnel de traitement des données dans le cloud peut difficilement évoluer car il ne peut pas répondre aux applications IoT les plus critiques en termes de temps de réponse. De plus, il génère beaucoup de trafic réseau lorsque le nombre d'objets augmente. Le Fog computing est le début de la solution à un tel défi. Dans ce présent travail, nous proposons une stratégie de gestion des données dans un environnement de fog computing basé sur la redondance des données. notre stratégie de replication de données permet de crée un nombre réduit de réplication en plus elle place ces copies d'une manière efficace afin de minimiser les transfères des données.Les résultats expérimentaux prouvent l'efficacité de notre proposition.

Mots clés : Fog computing, ordonnancement des tâches, workflows , réplication .

Abstract

In the years to come, the Internet of Things (IoT) will be one of the most popular data generation applications. In fact, more than 150 billion objects will be connected by 2025. IoT data can be processed and used by various devices spread across the network. The traditional centralized model of data processing in the cloud can hardly scale because it cannot meet the most critical IoT applications in terms of response time. In addition, it generates a lot of network traffic as the number of objects increases. Fog computing is the beginning of the solution to such a challenge. In this present work, we propose a data management strategy in a fog computing environment based on data redundancy. our data replication strategy allows to create a reduced number of replications in addition to placing these copies in an efficient manner in order to minimize data transfers. The experimental results prove the effectiveness of our proposal.

Keywords : Fog computing, scheduling tasks, workflows, replication.

ملخص

أحد أكثر تطبيقات توليد البيانات. في الواقع، سيتم توصيل (IoT) في السنوات القادمة، سيكون إنترنت الأشياء أكثر من 150 مليار كائن بحلول عام 2025. يمكن معالجة بيانات إنترنت الأشياء واستخدامها بواسطة أجهزة مختلفة موزعة عبر الشبكة. بالكاد يمكن للنموذج المركزي التقليدي لمعالجة البيانات في السحابة أن يتطور لأنه لا يستطيع تلبية معظم تطبيقات إنترنت الأشياء المهمة من حيث وقت الاستجابة. بالإضافة إلى ذلك، فإنه يولد الكثير من حركة مرور الشبكة عندما يزداد عدد الكائنات. حوسبة الضباب هي بداية الحل لمثل هذا التحدي. في هذا العمل الحالي، نقترح استراتيجية إدارة البيانات في بيئة الحوسبة الضبابية على أساس تكرار البيانات. نهجنا يجعل من الممكن تقليل حركة البيانات. النتائج التجريبية تثبت فعالية اقتراحنا

كلمات البحث: الحوسبة الضبابية، ترحيل ونسخ البيانات، البنيات الشجرية.

Introduction générale

L'Internet des objets (Internet of Things, IoT) est l'infrastructure utilisée pour connecter n'importe quel objet comme Smartphones, appareils phot et véhicules, pour permettre la capture à distance d'événements (par exemple : température, humidité, présence,...) via des capteurs, mais également le contrôle de l'environnement via des actionneurs.
Ces dernières années, les objets connectés ont connu une véritable démocratisation avec de plus en plus d'accessoires connectés entrant dans nos vies quotidienne, notre domicile, notre environnement de travail, mais aussi les villes. En effet, Cisco prédit que plus de 150 milliards d'objets seront connectés en 2025. Ce grand nombre d'objets générera une énorme quantité de données.
Actuellement, les données IoT sont traitées et stockées dans le cloud. Ce dernier répond à la majorité des besoins des applications IoT en termes d'accès ubiquitaire, de disponibilité et d'évolutivité des performances de traitement et de capacité de stockage.L'envoi de toutes les données vers le Cloud générera des goulets d'étranglement. Ceci produit des latences élevées et imprédictibles et, par conséquent, une qualité de service (QoS) dégradée.
De plus, il existe de nombreuses applications de l'Internet des objets, comme l'Internet des véhicules, la télémédecine et l'industrie 4.0, nécessitent un support de mobilité, un déploiement géo-distribué, un temps de réponse réel et une connaissance de la localisation que le Cloud computing ne peut pas assurer efficacement. Ces problèmes ont permis l'émergence d'un nouveau paradigme, le Fog computing. Le Fog offre une infrastructure informatique dense et géographiquement répartie à grande échelle. Ce paradigme étend le Cloud du cœur jusqu'aux périphérie du réseau (From cOre to edGe, en Anglais). Le Fog utilise les ressources disponibles dans des équipements du réseau localisés entre les centres de données du Cloud et les utilisateurs finaux éventuellement les objets connectés. Ces équipements, appelés aussi Nœuds du Fog, sont hétérogènes en termes de performances de traitement, de capacité de stockage et de latence d'accès pour les objets et les utilisateurs.
Les infrastructures de Fog peuvent également être hiérarchiques, avec une architecture de Cloud Computing à la racine de cette hiérarchie. Dans ces infrastructures, Les Nœuds du Fog, s'occupent des traitements peu coûteux mais qui nécessitent un faible temps de réponse, et délèguent au Cloud ceux qui sont gourmands en

ressources mais pour lesquels les temps de réponses sont moins importants.
Les caractéristiques du Fog computing conduisent à réexaminer de nombreux problèmes déjà connus dans d'autres contextes plus traditionnels. Comme la localisation de données est cruciale pour les traitements, il est important d'étudier comment les données doivent être organisées dans une telle infrastructure afin de réduire les latences dans le système et augmenter la disponibilité. C'est pourquoi nous cherchons à réaliser dans ce travail de fin d'étude, une solution de gestion des données dans lequel les utilisateurs soient capables d'accéder aux données depuis n'importe quel emplacement, avec le temps d'accès le plus faible possible, une réduction dans la consommation énergétique, sans chargé l'utilisation du réseau.
Dans notre travail l'objectif visé est de proposer une stratégie , basée sur la réplication des données dans le fog computing.
Notre stratégie comporte trois phases :

1. Phase de construction des vecteurs
2. Phase de clustering basée sur l'algorithme k-Means
3. Phase de réplication de données les plus fréquement demandé.

ORGANISATION DU MÉMOIRE

Le présent mémoire est composé de quatre chapitres principaux qui peuvent être résumés comme suit :

Chapitre 1 :

définit la notion, les domaines d'application et les défis d'Internet des objets. La définition du modèle de calcul Cloud Computing ainsi ces caractéristiques sont également donnés dans ce chapitre.

Chapitre 2 :

ce chapitre présente le domaine d'intérêt de ce projet qui est le Fog Computing, les avantages apportés et les différentes défits scientifiques à relever. Un état de l'art est établi par la suite sur les travaux existants qui utilisent le paradigme du Fog Computing et la réplication .

Chapitre 3 :

Le troisième chapitre sera réservé à la description détaillée de notre stratégie proposée.

Chapitre 4 :

Ce dernier chapitre présentera les étapes de l'implémentation de l'approche proposée. Nous y détaillerons la réalisation de certaines fonctionnalités ainsi que l'étude d'évaluation de cette stratégie. Les résultats d'expérimentation seront interprétés.
Enfin, Une synthèse et un ensemble de perspectives viendront pour clôturer notre travail.

Table des matières

Introduction générale **5**

1 IOT , Cloud Computing et Fog Computing **9**
1.1 Introduction : 9
1.2 Internet of Things (internet des objets) : 10
1.3 Domaine d'application de l'internet des objets : 11
1.3.1 L'internet des objets dans le domaine de la santé : 11
1.3.2 L'internet des objets dans le domaine des sportifs : 11
1.3.3 L'internet des objets dans le domaine domotique : 11
1.3.4 L'internet des objets dans le domaine de L'automobile : . . 12
1.3.5 L'internet des objets dans le domaine de la sécurité : 12
1.3.6 L'internet des objets dans le domainede l'industrie : 12
1.3.7 L'internet des objets dans le domaine de l'agriculture : . . . 12
1.4 Définition de Cloud Computing : 13
1.5 Terminologie générale du Cloud : 13
1.5.1 CAAS : 13
1.5.2 SAAS : 14
1.5.3 PAAS : 14
1.5.4 IAAS : 15
1.6 Types de Cloud Computing : 15
1.7 Modèle d'application de Cloud Computing : 15
1.8 Les Défis de Cloud Computing : 16
1.9 Domaines d'application du Cloud computing : 17
1.10 fog computing : 18
1.11 Architecture du Fog computing 18
1.12 Les avantages du Fog computing 20
1.13 Les défis du Fog computing 21
1.13.1 Hétérogénéité 21
1.13.2 Sécurité 21
1.13.3 Gestion et provision de ressources 21
1.13.4 Gestion des données 21
1.13.5 Gestion de l'énergie 22
1.13.6 Modèle de programmation 22

1.13.7 Qualité de service (QoS) : 22
1.13.7.1 la connectivité : 22
1.13.7.2 la fiabilité : 22
1.13.7.3 la capacité : 23
1.13.7.4 la latence : 23
1.13.8 Complexité : 23
1.14 Conclusion 24

2 ETAT DE L'ART SUR LA RÉPLICATION DE DONNÉES 25
2.1 Introduction 25
2.2 LA RÉPLICATION DES DONNEES : 26
2.3 Types de réplication : 26
2.3.1 Réplication synchrone : 26
2.3.2 Réplication asynchrone : 26
2.4 AVANTAGES DE LA RÉPLICATION : 26
2.5 Travaux connexes : 27
2.5.1 Fog Computing and Its Role in the Internet of Things [31] : 27
2.5.2 Data Placement and Task Scheduling Optimization for Data Intensive Scientific Workflow in Multiple Data Centers Environment[32] : 28
2.6 CONCLUSION 28

3 Desciption et Modélisation de l'approche proposée 29
3.1 Introduction 29
3.2 Architecture hiérarchique proposée 30
3.3 APPROCHE PROPOSÉE : 31
3.3.1 PHASE DE CONSTRUCTION ET NORMALISATION DES VECTEURS : 31
3.3.1.1 ETAP 1 : construction des vecteurs 31
3.3.1.2 ETAP 2 : la normalisation des vecteurs 32
3.3.2 PHASE DE CLUSTERING AVEC K-MEANS 33
3.3.3 PHASE DE RÉPLICATION DYNAMIQUE : 35
3.3.3.1 CRÉATION DES RÈPLIQUES 35
3.3.3.2 NOMBRE DE RÉPLIQUE 36
3.3.3.3 PLACEMENT DES RÉPLIQUES 36
3.4 CONCLUSION 37

4 IMPLÉMENTATION ET RÉSULTATS 38
4.1 INTRODUCTION 38
4.2 Langage et environnement de développement 39
4.2.1 Langage de programmation Java : 39
4.2.2 Environnement de développement 40
4.2.3 Simulateur iFogSim 41
4.3 IMPLEMENTATION : 44

4.3.1 INTERFACE PRINCIPALE : 44
4.3.2 Configuration de simulation : 45
4.3.3 Lancement de simulation et visualisation des résultats . . . 45
4.3.3.1 SIMULATION 1 : 45
4.3.3.2 SIMULATION 2 : 46
4.3.4 Utilisation totale du réseau 47
4.4 Conclusion 48

Chapitre 1

IOT , Cloud Computing et Fog Computing

1.1 Introduction :

Internet des objets est une réalisation technique de l'informatique omniprésente où la technologie est naturellement intégrée aux choses de tous les jours. Très prometteur, ce concept ouvre la voie vers une multitude de scénarios basés sur l'interconnexion entre le monde physique et le monde virtuel : domotique, e-santé, ville intelligente, logistique, sécurité, etc. Cependant, comme d'autres concepts prometteurs, Il est confronté à un certain nombre de problèmes techniques et non techniques qui doivent être étudiés pour permettre à l'Internet des objets d'atteindre son plein potentiel..

1.2 Internet of Things (internet des objets) :

L'Internet des objets est un concept concrétisant la vision de l'informatique ubiquitaire telle qu'imaginée en 1991 par Mark Weiser[29], où la technologie s'efface peu à peu dans l'environnement des utilisateurs, intégrée naturellement à l'intérieur des objets du quotidien. La technologie n'est plus alors représentée par un objet unique, l'ordinateur personnel, mais se présente au contraire sous la forme d'appareils spécialisés et simples d'emploi, capables de communiquer au travers de plusieurs types de réseaux sans : liseuses numériques, télévisions et montres connectées, ordinateurs de bord, téléphones intelligents, etc...
Le terme Internet des objets a été utilisé pour la première fois en 1999 par Kevin Ashton (est un pionnier britannique de la technologie qui a cofondé l'Auto-ID Center du MIT « Massachusetts Institute of Technology ») pour décrire des objets équipés de puces d'identification par radio fréquence (ou puce RFID) [14].
Jusqu'à présent, l'IoT n'est pas clairement défini, nous présentons dans ce que suit quelques définitions de différentes organisations mondiales. Le terme « Internet des objets » est utilisé comme mot-clé générique pour couvrir divers aspects liés à la l'extension d'Internet et du Web dans le domaine physique, grâce au déploiement généralisé de dispositifs répartis dans l'espace avec des capacités intégrées d'identification, de détection et / ou d'actionnement. L 'Internet of Things envisage un avenir dans lequel les entités numériques et physiques peuvent être liées, au moyen de technologies de l'information et de la communication, pour permettre une toute nouvelle classe d'applications et de services. [19]
Un « Internet des objets » signifie « un réseau mondial d'objets interconnectés adressables de manière unique, basé sur des protocoles de communication standard ", ou, plus largement :" Des objets ayant des identités et des personnalités opérant dans des espaces intelligents utilisant des interfaces intelligentes pour se connecte r et communiquer à l'intérieur contextes sociaux, environnementaux et utilisateurs " [5]
Une infrastructure réseau globale, reliant des objets physiques et virtuels grâce à l'exploitation de la capture de données et capacités de communication. Cette infrastructure comprend Internet et un réseau existants et en évolution développements. Il offrira des capacités spécifiques d'identification d'objet, de capteur et de connexion comme base pour le développement de services et d'applications coopératifs indépendants. Ceux-ci seront caractérisés par un degré de saisie autonome des données, transfert d'événements, connectivité réseau et interopérabilité [6].
L'idée de base de ce concept est la présence omniprésente autour de nous d'une variété de choses ou d'objets – tels sous forme d'étiquettes d'identification par radiofréquence (RFID), de capteurs, d'actionneurs, de téléphones mobiles, etc. des schémas d'adressage uniques, sont capables d'interagir les uns avec les autres et de coopérer avec leurs voisins pour atteindre objectifs communs [7].

1.3 Domaine d'application de l'internet des objets :

Le marché des objets connectés est promis à une grande croissance dans les années à venir car il a une valeur immense dans les différents domaines d'objets connectés pour les professionnels. Cependant, seules quelques applications sont actuellement déployées [2]. L'utilisation de l'IDO permettra le développement de plusieurs applications intelligentes qui toucheront essentiellement ceux qu'on citera dans ce qui suit, nous citrons brièvement des exemples d'applications de l'IDO [3].

1.3.1 L'internet des objets dans le domaine de la santé :

Les objets connectés peuvent servir à réduire quelque éléments de dépenses pour les remplacer par d'autrs il permet aussi de favoriser l'hospitalisation à domicile, qui assurera le contrôle et le suivi des signes cliniques des patients par la mise en place des réseaux personnels de surveillance, ces réseaux seront constitués de bio-captures posés sur le corps des patients ou dans leurs lieux d'hospitalisation. Cela facilitera la télésurveillance des patients qui permettras de réduire les erreurs médicales, optimiser la consommation de médicaments ou encore leur prise régulière, et même encourager la prévention de certaines maladies, l'internet des objets permettre aussi de suivre sa tension, son rythme cardiaque, la qualité de sa respiration ou encore sa masse graisseuse, et d'autres objets connectés médicaux , brosse a dent connectée ou encore, le scanner qui calcule le nombre de calories dans votre assiette[4].

1.3.2 L'internet des objets dans le domaine des sportifs :

De nombreux objets connectés comme des montres ou des bracelets connectés vous permettrons pendant la journée de calculer le nombre de pas effectuée, la distance par courue, votre temps d'activités, les calories brulées, ainsi pendant la nuit en calculant vos heurs de sommeil. Pour les passionnés de High-tech, c'est un grand marché qui s'ouvre à eux ! De la montre connectée au téléviseur connecté en passant par les appareils photos, les montre, les drones, les lunettes (Google glass)[6].

1.3.3 L'internet des objets dans le domaine domotique :

Les objets connectés sont une réelle révolution, ils permettent de la rendre connectée, d'où le nom très utilisé de smart home des nombreux dispositifs de sécurité détecteur de fumée, surveillance, serrure connecté dans le domaine de l'électroménager réfrigérateur connecté, l'électroménager connecté. De la décoration de la maison des designs, lampe connecté, cadre lumineux, plante connecté. L'internet des objets change les modalités d'accès au réseau et produit de nouvelles interactions homme-machine. Il existe aussi des villes intelligente (Smart Caties) est utiliser pour désigner l'écosystème cyber physique émergeant par le

déploiement d'une infrastructure de communication avancée et de nouveaux services sur des scénarios à l'échelle de la ville. Grace à des services avancés, il est en effet possible d'optimiser l'utilisation des infrastructures physiques de la ville (par exemple, les réseaux routiers, le réseau électrique, etc.) et la qualité de vie des citoyens [7].

1.3.4 L'internet des objets dans le domaine de L'automobile :

Le marché des transports à déjà anticipé l'arrivée des objets connectés. Parmi les enjeux les plus fréquents que ce domaine fait naitre on retrouve la réduction des accidents et des embouteillages, le partage de voitures, le développement des offres de VTC et de TAX ou encore la gestion des flots automobile [7].

1.3.5 L'internet des objets dans le domaine de la sécurité :

Pour le cabinet en stratégie, ces entreprises vont rapidement se positionner comme des alliés des personnes qui résident dans leur domicile. En fournissant des données relatives à la consommation d'énergie aux foyers, ces groupes vont apparaitre comme des arguments contre le facteur EDF pour les fournisseurs d'énergie la précision sera difficile à tenir car ils seront probablement contraints d'accompagner leurs clients dans une baisse de leurs facteurs énergétique [8].

1.3.6 L'internet des objets dans le domainede l'industrie :

Le déploiement de L'IoT dans l'industrie sera certainement un grand support pour le développement de l'économie et le secteur des services, puisque L'IDO permettra d'assurer un suivi total des produits, de la chaine de production, jusqu'à la chaine logistique et de distribution en supervisant les conditions d'approvisionnements. Cette traçabilité de bout en bout facilitera la lutte contre la contrefaçon, la fraude et les crimes économiques transfrontaliers. Certains éditeurs tels que SAP et CISCO montrant d'ores et déjà comment certaines zones industrielles comme le port d'Hambourg ont pu être équipés en pucs et autres objets connectés. L'internet couvre un énorme nombre d'industries et utilise des cas qui s'étendent d'un seul dispositif contraint aux déploiements croisés de technologies intégrées de systèmes Cloud connectés en temps réel.

1.3.7 L'internet des objets dans le domaine de l'agriculture :

L'IoT présentera des outils de choix pour la supervision de l'environnement des cultures, ce qui permettra une meilleure aide à la décision en agriculture. L'IoT servira non seulement à optimiser l'eau d'irrigation, l'usage des intrants et la planification des travaux agricoles, mai aussi, cette technologie peut être utilisée pour lutter contre la pollution (l'air, le sol et les eaux) et améliorer la qualité de l'environnement en général. L'usage des objets connectés se démocratise dans l'agriculture. De nombreuses amélioration ou découlent la gestion des engins

agricoles, la maitrise de l'irrigation ou la gestion optimisée des intrants, que la surveillance de la croissance des plantes ou encore la prévention des risques météo. De quoi renouveler en profondeur les pratiques des cette activité ancestrale, grâce à l'analyse des données récoltées et au pilotage de plus en plus fin des exploitations [9].

1.4 Définition de Cloud Computing :

Le Cloud Computing est un concept qui consiste à accéder à des données et des services sur un serveur distant. Traditionnellement, une entreprise utilisait sa propre infrastructure pour héberger ses services. Elle achetait donc ses propres serveurs, et assurait le développement et la maintenance des systèmes nécessaires à son fonctionnement. Par opposition, le Cloud Computing se repose sur une architecture distante, Le fournisseur donc assure la continuité du service et de la maintenance. Les services de Cloud Computing sont accessibles via un navigateur web [4]. Le terme Cloud Computing étant anglais, on retrouve comme synonymes les termes suivant : informatique virtuelle, informatique dans les Cloud et informatiques en Cloud ou encore informatique dématérialisée. L'emplacement des données dans le Cloud n'est pas connu des clients, ceux-ci ont simplement l'accès a la partie applicative, sans se soucier du reste [4].

1.5 Terminologie générale du Cloud :

Le monde informatique contient des milliers d'abréviations et d'acronymes tous plus obscurs les uns que les autres, et le Cloud Computing n'échappe pas à la règle [5]. Le monde du Cloud Computing est littéralement noyé sous les abréviations et les acronymes dont certains ont même plusieurs sens. En l'occurrence, ceux qui nous intéressent actuellement sont les acronymes en AAS (As A Service). On va décrire les plus importants sur la figure suivante. Fig. I.1.

1.5.1 CAAS :

Plusieurs significations différentes de CAAS sont utilisées comme : Capability as a Service, Communication as a Service, Cloud as a Service, Computing as a Service, Content as a Service, Community as a Service, Car as a Service [5]. Toutefois, la plupart des gens s'accordent pour dire que le CAAS correspond bien à (Communication as a Service). Le CAAS consiste à fournir des moyens de communication en tant que service, via Internet. Par exemple, (Telephony as a service) est un sous-groupe du CAAS, au même titre que (Email as a service) [5]. En réalité, ce n'est pas du tout nouveau et cela consiste uniquement à mettre des noms différents à des services existants (VOIP, web mail, etc.).

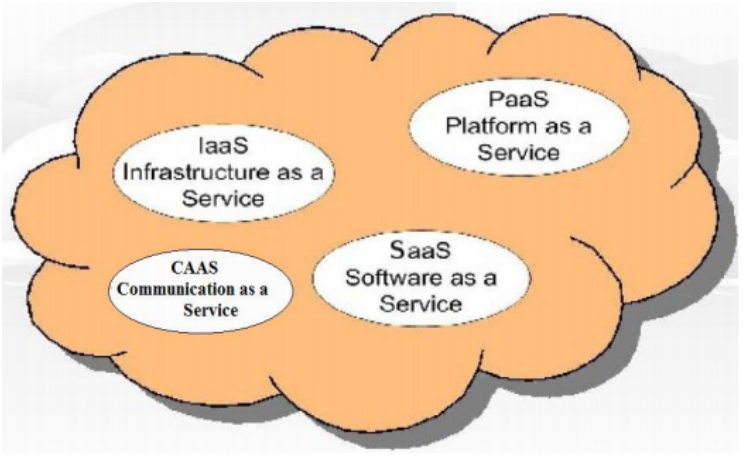

FIGURE 1.1 – Terminologies de Cloud Computing

1.5.2 SAAS :

L'acronyme « SAAS » est le plus connu dans le monde du Cloud Computing. Sa signification est « Software as a Service », autrement dit, application en tant que service, c'est un modèle de déploiement d'application dans lequel un fournisseur loue une application clé en main à ses clients en tant que service à la demande au lieu de leur facturer des licences. De cette façon, l'utilisateur final n'a plus besoin d'installer tous les logiciels existants sur sa machine de travail. Cela réduit également la maintenance en supprimant le besoin de mettre à jour les applications. Ce type de modèle transforme les budgets logiciels en dépenses variables et non plus fixes et il n'est plus nécessaire d'acquérir une version du logiciel pour chaque personne au sein de l'entreprise [5].

1.5.3 PAAS :

Le PAAS qui signifie « Platform as a Service » est une architecture composée de tous les éléments nécessaires pour soutenir la construction, la livraison, le déploiement et le cycle de vie complet des applications et des services exclusivement disponibles à partir d'internet. Elle est également connue sous le nom de « CloudWare » [5]. Le PAAS offre des facilités à gérer le déroulement des opérations lors de la conception, du développement, du test, du déploiement et de l'hébergement d'applications web à travers des outils et des services tels que [5] :
1. Le travail collaboratif (« team collaboration »).
2. L'intégration des services web et bases de données.
Ces services sont fournis au travers une solution complète destinée aux développeurs et disponible immédiatement via l'internet.

1.5.4 IAAS :

4. IAAS : L'IAAS (Infrastructure as a Service) est un modèle qui permet de fournir des infrastructures informatiques en tant que service. Ce terme était originellement connu sous le nom de (Hardware as a Service).Ces infrastructures virtuelles composent un des domaines du « As a Service » en empruntant la même philosophie de fonctionnement et de tarification que la plupart des services du Cloud Computing [5]. Plutôt que d'acheter des serveurs, des logiciels, et l'espace dans un centre de traitement de données et/ou de l'équipement réseau, les clients n'ont plus qu'à louer les ressources auprès des prestataires de service. Le service est alors typiquement tarifé en fonction de l'utilisation et de la quantité des ressources consommées. De ce fait, le coût reflète typiquement le niveau d'activité de chaque client. C'est une évolution de l'hébergement Internet qui se différencie des anciens modes de fonctionnement, on distingue [5] :
1. Hébergement mutualité : une machine pour plusieurs clients, gérée par un prestataire de service et dont les clients payent le même prix peu importe leur utilisation.
2. Hébergement dédié : une machine pour un client, gérée le plus souvent par le client lui même et pour laquelle le client paye le même prix chaque mois peu importe son utilisation.
3. Infrastructure as a Service : un nombre indéfini de machines pour un nombre indéfini des clients, dont les ressources sont combinées et partagées pour tous les clients. Chaque client paye en fonction de son utilisation de l'architecture.

1.6 Types de Cloud Computing :

Le concept de Cloud computing est encore en développement. Cependant, on peut citer trois types de Cloud computing [6] :
1. le Cloud privé (ou interne) : c'est un réseau informatique propriétaire ou un centre de données qui fournit des services hébergés pour un nombre limité d'utilisateurs.
2. le Cloud public (ou externe) : C'est un prestataire de services qui propose des services de stockage et d'applications Web pour le grand public. Ces services peuvent être gratuits ou payants.
3. le Cloud hybride (interne et externe) : C'est un environnement composé de multiples prestataires internes et externes.

1.7 Modèle d'application de Cloud Computing :

Le Cloud Computing est un nouveau modèle pour fournir aux entreprises les services informatiques. Ce modèle est basé sur une architecture standard qui contient trois phases : la phase stratégique, la phase de planification et la phase de déploiement, chaque phase contient plusieurs étapes. La figure ci-dessous montre la structure générale de ce modèle (Fig. I.2) [3].

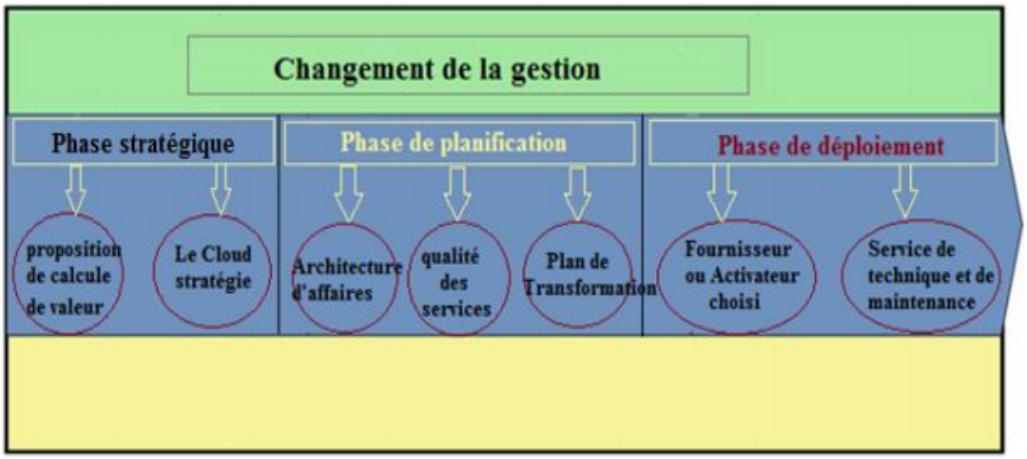

FIGURE 1.2 – Modèle d'application de Cloud Computing

1.8 Les Défis de Cloud Computing :

Le Cloud présente plusieurs défis [2] :

Sécurité : L'utilisation des réseaux publics, dans le cas du Cloud public, entraîne des risques liés à la sécurité du Cloud. En effet, la connexion entre les postes et les serveurs applicatifs passe par le réseau Internet, et expose à des risques supplémentaires de cyber attaques, et de violation de confidentialité.

Disponibilité : Le client d'un service de Cloud computing devient très dépendant de la qualité du réseau pour accéder à ce service. Aucun fournisseur de service Cloud ne peut garantir une disponibilité de 100

Piratage : Tout comme les logiciels installés localement, les services de Cloud computing sont utilisables pour lancer des attaques (craquage de mots de passe, déni de service..). En 2009, par exemple, un cheval de Troie a utilisé illégalement un service du Cloud public d'amazone pour infecter des ordinateurs.

Cadre légal : Des questions juridiques peuvent se poser, notamment par l'absence de localisation précise des données du Cloud computing.

Réversibilité : En cas de rupture de contrat ou de changement de fournisseur, le client doit s'assurer de la récupération et de la destruction de ses données sur l'infrastructure du fournisseur après sa migration.

1.9 Domaines d'application du Cloud computing :

Actuellement et sans que les utilisateurs se rendent compte, de nombreuses applications sont passées dans le Cloud :

Domaine militaire : La technologie employée dans un environnement militaire doit être opérationnelle en toute circonstance. Qu'il s'agisse de garantir l'exécution correcte des opérations quotidiennes, le fonctionnement constant des communications internationales ou la gestion réussie de situationde crise, la solution militaire Camera manager (vidéo protection dans le Cloud) de Panasonic permet de garder un oeil constant sur les organisations militaires. Il ne se limite pas à l'enregistrement et l'affichage en temps réel, mais il embarque également de nombreuses fonctions courantes de gestion vidéo-programmation, alarme, utilisation d'une API, gestion des utilisateurs, et est capable de supporter tout ce qu'on fait, quelles que soient les difficultés à surmonter .

Domaine médical :Le Cloud a la capacité d'améliorer la collaboration dans lindustrie des soins de santé. En permettant aux professionnels de stocker et accéder aux données à distance. Les professionnels de santé aux quatre coins du monde peuvent accéder aux données des patients et appliquer les soins nécessaires au plus vite. De plus, la téléconférence, les mises à jour des dernières innovations en soins de santé et les conditions du patient en temps réel, permettent d'économiser du temps pour sauver des vies [19]. Followmed est une solution Cloud proposant de multiples fonctionnalités ayant pour objectif de simplifier le quotidien des professionnels de santé, gérez leur activité médicale, innovez leur communication sur le réseau social dédié aux professionnels de la santé, utilisez ce logiciel médical en Cloud pour stocker et accéder à l'ensemble de leurs données où qu'ils soient .

Domaine de l'éducation :Le Cloud est un excellent levier technologique pour fournir des services collaboratifs extrêmement riches permettant de créer des sessions extrêmement interactives et de réaliser des opérations d'apprentissage à distance qui vont être complémentaires à l'enseignement traditionnel. Par exemple des usages de type blending-learning (ou formation mixte) avec de la vidéo en apprentissage seul, du cours en présentiel et du cours virtuel connecté qui va permettre aux étudiants d'affranchir des limites de l'espace et offrir un panel de possibilité plus important . Virtual Computing Lab (VCL) est une infrastructure Cloud qui fédère des ressources informatiques de plusieurs sites (serveurs, systèmes de stockage, instances de logiciels). Elle permet aux élèves des différents établissementsprimaires et secondaires de l'Etat ainsi qu'aux étudiants des différents campus de l'université d'accéder, où qu'ils se trouvent (en ce compris chez eux), à un pool de ressources techniques et pédagogiques évoluées et à jour .

Domaine industriel Pour pérenniser la croissance de l'industrie manufacturière, les fabricants doivent sans cesse améliorer la précision et la rapidité de

leurs procédés, et optimiser chaque interaction avec les fournisseurs, distributeurs et prestataires de services. On assiste ainsi à la démocratisation des architectures technologiques "just in time" (ou dynamiques), basées sur des Clouds publics et privés, qui aident les fabricants à tenir leurs objectifs et à maintenir leur compétitivité . Salesforce est une plateforme Cloud complète permettant de gérer les interactions entre les fournisseurs et leurs clients, et les aider à prospérer et réussir .

1.10 fog computing :

Le Fog computing ou « l'informatique en brouillard » est un paradigme proposé par Cisco en 2012 essentiellement pour faire face aux problématiques des latences élevées et du trafic réseau important causés par l'utilisation du Cloud [13, 14]. il est considéré comme un paradigme de calcul distribué qui agit comme une couche intermédiaire entre les centres de données du Cloud et les dispositifs/capteurs de l'IoT. Il offre des fonctions de calcul, de mise en réseau et de stockage afin que les services basés sur le Cloud puissent être étendus au plus près des dispositifs/capteurs de l'IoT [15]. Le Fog ne remplace pas le Cloud, mais ces deux paradigmes de calcul sont complémentaires [16]. En effet, ils coopèrent pour fournir une plate-forme extensible qui supporte à la fois des applications nécessitant des latences courtes et prédictibles ainsi que des applications nécessitant des traitements complexes et du stockage de données permanent [17, 25]. L'idée est de servir les requêtes nécessitant des latences courtes (ex. contrôle du trafic routier, parking intelligent et diffusion en direct) par des équipements localisés dans le Fog, tandis que les requêtes nécessitant des traitements complexes et des données d'historisation (ex. photos, flux vidéos, données des réseaux sociaux, etc) sont servies par le Cloud [17]. Il est important de noter que les infrastructures de Fog sont hétérogènes, c'est-à-dire, qu'ils peuvent être hébergées sur différents types des équipements physiques sont de nature hétérogène au regard de leur performance [18]. Comme discuté auparavant, le Cloud fournit une capacité de stockage et de calcul quasi illimitée au prix d'une latence élevée tandis que le noeud de Fog le plus proche de l'utilisateur fournit avec une très faible latence, une faible capacité de calcul et de stockage.

1.11 Architecture du Fog computing

Le Fog présente une infrastructure virtuelle et géo-distribuée intégrant un grand nombre d'équipements hétérogènes et qui sont interconnectés [19]. Comme illustré dans la Figure 2.1, l'infrastructure du Fog comprend les périphéries du réseau, le niveau d'agrégation et le cœur de réseau. Ci-après, nous décrivons chaque niveau de l'infrastructure illustrée dans la Figure 2.1. L'architecture hiérarchique est composée des trois couches suivantes :

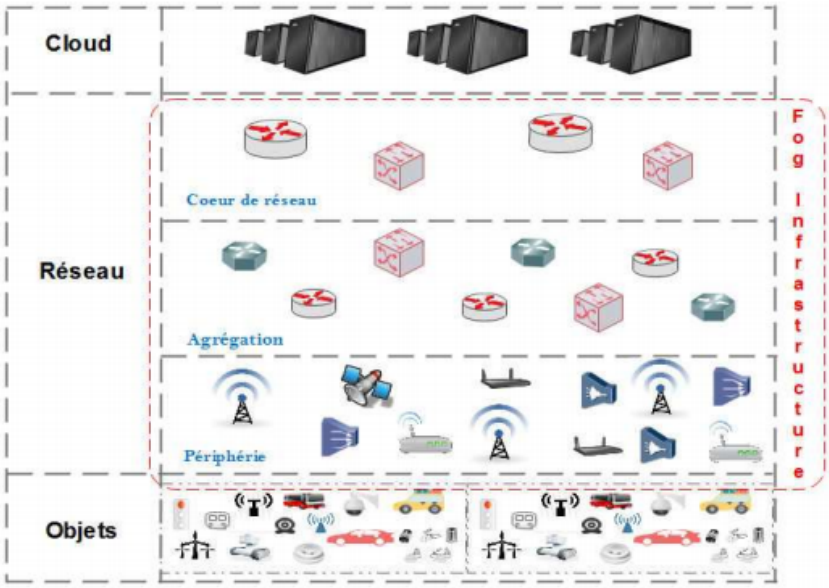

FIGURE 1.3 – Infrastructure du Fog computing

Objets : C'est la couche la plus proche des utilisateurs finaux et de leur environnement physique. Il se compose de divers dispositifs de l'IoT, par exemple des capteurs, des smartphones, des véhicules intelligents, des robots, des lecteurs, etc. Ces dispositifs sont généralement distribués géographiquement. Ils sont chargés de surveiller l'environnement et de transmettre ces données à la couche supérieure pour le traitement et le stockage [20].

Fog : Cette couche est située en bordure du réseau, elle est composée d'un grand nombre de nœuds de Fog, qui comprennent généralement des routeurs, des passerelles, des commutateurs, des points d'accès, des stations de base, des serveurs de Fog spécifiques, etc. De plus, Les nœuds de Fog sont largement distribués dans différentes zones géographiques afin de couvrir une vaste zone à grande échelle de réseaux d'IoT [21]. La couche Fog est utilisée pour héberger des applications nécessitant un temps de réponse court, ainsi que pour faire l'agrégation, le filtrage et le pré traitement des données avant de les envoyer au Cloud. En effet, les terminaux peuvent se connecter aux nœuds de Fog par le biais des technologies de communication comme le Wi-Fi, Bluetooth ou le réseau cellulaire. Par ailleurs, d'autres fonctionnalités peuvent être déployées dans ce niveau comme celles du CDN pour mettre en caches des données, et du SDN et du NFV pour administrer le réseau [17] [21].

Cloud : c'est le niveau le plus haut dans l'infrastructure. La couche de cloud computing se compose de plusieurs serveurs de stockage et de calcul d'haute per-

formance dans des centres de données distants permettant d'héberger et de fournir divers services d'application destinés, tels que l'analyse et le stockage permanent d'une énorme quantité de données[19]. Voici un tableau (cf. fig. 2.1) récapitulatif de notre analyse de l'existant...

1.12 Les avantages du Fog computing

Plusieurs travaux de recherche ont listé les bénéfices apportés par le Fog computing par rapport au paradigme du Cloud computing [23, 22, 23]. Ci-après nous en citons quelques uns.

Réduction des latences : en raison de sa proximité avec les utilisateurs finaux, le Fog a la capacité de supporter des applications qui nécessitent des latences courtes et stables (ex. la réalité augmentée, les jeux vidéo en ligne).

Réduction du trafic réseau : le Fog computing permet d'exécuter des fonctions de traitement, de filtrage et d'agrégation de données tout au long du chemin du réseau. En effet, la pertinence de l'envoi de données est examinée à chaque étape de la transmission permettant une réduction importante du trafic réseau.

Géolocalisation et support de la mobilité : la distribution géographique des nœuds de Fog aide à localiser les objets. De plus, la latence existante entre des nœuds de Fog voisins est courte. Cela aide à migrer des taches (ex. à base de conteneurs) d'un nœud à un autre de manière transparente, afin de supporter la mobilité des objets.

Passage à l'échelle (i) la géodistribution des nœuds de Fog et leur proximité avec les utilisateurs finaux permettent de gérer un grand nombre d'objets connectés (la répartition de la charge de travail).
(ii) Les nœuds de Fog sont de nature hétérogène avec des performances et des coûts différents. Ce deuxième point permets de déployer, selon le besoin, de nouveaux nœuds de manière simple et moins coûteuse (en comparaison avec le Cloud). • Interopérabilité et fédération : les infrastructures de Fog comprennent un grand nombre de nœuds géo-distribués. Plusieurs de ces nœuds peuvent fédérer/coopérer dans un cluster pour réaliser des taches complexes telle que l'analyse de données massives.

Décharge de traitement (offloading) en raison de sa proximité, le Fog peut aider les objets ayant des ressources limitées à décharger une partie de leurs traitements aux nœuds de calcul localisés en périphérie du réseau. Cela permet aux

objets, par exemple, de préserver leur énergie, de réduire le temps de traitement ou d'augmenter la capacité de stockage de données.

La disponibilité : la géo-distribution des infrastructures de Fog aide à lancer une tache ou à stocker plusieurs copies d'une donnée sur des nœuds de Fog différents. Ceci permet d'augmenter la disponibilité du service et la résilience contre la perte de données.

1.13 Les défis du Fog computing

Bien que le Fog computing offre sans aucun doute plusieurs cas d'utilisation significatifs et beaucoup d'avantages, ce paradigme reste nouveau et et nécessite d'investiguer les défis suivants.

1.13.1 Hétérogénéité

en plus de l'hétérogénéité trouvée dans l'IoT au regard des différents types d'objets, de données, de technologies de communication et de services, les infrastructures de Fog comprennent des nœuds de nature et de performances différentes. La gestion de l'hétérogénéité dans un environnement de Fog et d'IoT représente un défi majeur aujourd'hui [12].

1.13.2 Sécurité

les nœuds du Fog peuvent être déployés à l'extérieur et laissés sans surveillance (i.e. dans les rues, au-dessus des bâtiments, etc.). Ils sont donc vulnérables à des attaques telles que le détournement de données et l'écoute indiscrète [24]. Afin de préserver la sécurité du système, le déploiement des solutions pour le contrôle d'accès, l'authentification et la détection d'intrusion sont nécessaires dans chaque niveau de l'infrastructure [25, 26].

1.13.3 Gestion et provision de ressources

: les nœuds de Fog sont en général des équipements réseaux ayant une puissance de calcul et une capacité de stockage limitées. Des solutions efficaces sont nécessaires pour gérer l'ordonnancement des taches dans ces infrastructures (ex. des solutions basées sur la priorité et la migration) [28]. De plus, afin de fournir un support de mobilité notamment dans le cas des objets connectés, les ressources doivent être pré-allouées, par exemple suivant des méthodes probabilistes basées sur l'historique des utilisateurs [17].

1.13.4 Gestion des données

le suivi et la gestion des données dans les différents niveaux de l'infrastructure du Fog (i.e. périphérie, agrégation et cœur) est un défi majeur. La découverte,

la réplication, le placement et la persistance des données nécessitent un examen attentif dans ce contexte [16].

1.13.5 Gestion de l'énergie

comme mentionné ci-dessus, les infrastructures de Fog comprennent un grand nombre de nœuds répartis géographiquement. La consommation énergétique doit être donc plus élevée en comparaison avec celle du Cloud [17, 13]. De grands efforts de recherches sont nécessaires pour développer des solutions efficaces pour la gestion d'énergie, par exemple, des processus de traitement de données et des protocoles de communications moins coûteux en termes de consommation énergétique sont à développer [13].

1.13.6 Modèle de programmation

dans le Cloud, les infrastructures sont transparentes pour les utilisateurs. Les traitements sont réalisés dans des serveurs virtualisés dans les centres de données. En revanche, dans le Fog, les traitements sont faits à différents niveaux de l'infrastructure (i.e. périphéries, agrégation et cœur de réseau) qui sont des plates-formes dynamiques et hétérogènes. Afin de faciliter le développement des applications sur les plates-formes de Fog computing, il est nécessaire de fournir un modèle unifié qui prend en compte l'aspect dynamique, hiérarchique et hétérogène des ressources du Fog [13, 15].

1.13.7 Qualité de service (QoS) :

dans [15], les auteurs ont étudié la QoS dans le Fog computing selon quatre aspects :

1.13.7.1 la connectivité :

le Fog étend les services du Cloud jusqu'aux périphéries du réseau. Dans un tel réseau hétérogène, le partitionnement, le regroupement et la collaboration fournissent une opportunité pour optimiser le coût, augmenter le débit, ou réduire la consommation énergétique [26, 99]. Le défi présenté ici consiste à concevoir des algorithmes de communication efficaces pour optimiser les métriques cités ci-dessus [1]

1.13.7.2 la fiabilité :

comme le Fog computing est réalisé par l'intégration d'un grand nombre d'équipements répartis géographiquement, la fiabilité est l'un des principaux défis à considérer lors de la conception d'un tel système [17]. La fiabilité peut être améliorée grâce à une vérification périodique pour reprendre après un échec, à un re-ordonnancement des tâches échouées ou à une réplication pour exploiter le

traitement en parallèle. Toutefois, les points de contrôle et d'ordonnancement ne peuvent pas s'adapter à l'environnement hautement dynamique du Fog à cause des latences. La réplication semble plus prometteuse, mais elle repose sur le fonctionnement synchronisé de plusieurs nœuds de Fog [15].

1.13.7.3 la capacité :

: le Fog computing a été proposé principalement pour lutter contre les latences élevées du Cloud. Le Fog est utilisé pour déployer des applications sensibles aux latences comme l'Internet des véhicules, la santé et l'industrie 4.0. Afin de réduire la latence, il est important d'étudier comment les données et leurs traitements sont placés dans l'infrastructure. Par exemple, un nœud de Fog peut avoir besoin de traiter des données distribuées dans plusieurs nœuds éloignés. Le calcul ne peut être démarré qu'après avoir récupéré toutes les données requises, ce qui ajoute de la latence au service [15].

1.13.7.4 la latence :

cet aspect a été étudié selon deux critères : (i) la bande passante réseau et (ii) la capacité de stockage. Afin d'obtenir une bande passante élevée et une utilisation efficace du stockage, il est important de réaliser des fonctions d'agrégations et de filtrage dans les périphéries du réseau. Cependant, ce niveau de l'infrastructure inclut des équipements limités en ressources de traitements et de stockage de données, ce qui crée un défi pour le choix de traitement à réaliser et de données à stocker dans ce niveau.

1.13.8 Complexité :

les algorithmes d'optimisation existants sont en général ciblés sur le temps de traitement et l'utilisation de ressources. Vu que le Fog fournit une plate-forme avec des milliers de nœuds pour servir des milliards d'objets connectés, il est nécessaire de concevoir des solutions de gestion décentralisées et coopératives. Par exemple, afin d'accélérer le temps de placement de données ou de traitements, il est nécessaire d'utiliser des algorithmes parallèles ou des méthodes approchées (ex. basées sur le concept de diviser pour régner), plutôt que d'utiliser des méthodes exactes et centralisées [12]

1.14 Conclusion

Dans cette partie, nous avons défini l'iot et le Cloud computing, nous avons aussi donné ses caractéristiques, ses challenges ainsi que ses domaines d'application. Le Cloud computing marque une réelle avancée vers l'infrastructure informatique dématérialisée. Il fournit des ressources informatiques, logicielles ou matérielles, accessible à distance en tant que service. L'adoption de ce modèle soulève un certain nombre de défis, notamment au sujet de la disponibilité des ressources. Dans le prochain chapitre nous introduirons le Fog computing ainsi et la réplication des données dans le environnement Fog computing.

Chapitre 2

ETAT DE L'ART SUR LA RÉPLICATION DE DONNÉES

2.1 Introduction

L'Internet des objets d'aujourd'hui est considéré comme l'une des technologies révolutionnaires de ce siècle. D'ici 2025, Cisco prédit que le nombre d'objets connectés dépassera 150 milliards. Ce grand nombre d'objets va générer une énorme quantité de données. Il est généralement traité et stocké dans le cloud. Cependant, le Cloud Computing présente un paradigme qualifié de centraliser et situé loin des utilisateurs, ce que résulte une latence très élevée et des goulots d'étranglements. Ces problèmes ont permis l'émergence d'un nouveau paradigme : le Fog Computing. Dans ce chapitre, nous allons dans un premier temps définir le Fog Computing. Ensuite, nous présenterons l'architecture de ce nouveau paradigme, puis nous décrirons les bénéfices apportés par le Fog computing par rapport au paradigme du Cloud computing et la réplication et leur type et les avantage de la réplication. La dernière section présentera quelques travaux connexes existants dans la littérature.

2.2 LA RÉPLICATION DES DONNEES :

La réplication crée une ou plusieurs copies d'une entité existante, pour améliorer la fiabilité, la tolérance aux pannes, ou la disponibilité. On parle de réplication de données si les mêmes données sont dupliquées sur plusieurs périphériques.

2.3 Types de réplication :

2.3.1 Réplication synchrone :

Elle garantit que lorsqu'une transaction met à jour une réplique primaire, toutes ses répliques secondaires sont mises à jour dans la même transaction. L'avantage essentiel de la mise à jour synchrone est de garder toutes les données au même niveau de mise à jour. Le système peut alors garantir la fourniture de la dernière version des données quel que soit la réplique accédée. Les inconvénients sont cependant multiples, ce sont d'une part, la nécessité de gérer des transactions multiples coûteuses en ressources et d'autre part, la complexité des algorithmes de gestion de panne d'un site, etc. C'est pour cela que l'on préfère souvent le mode de mise à jour asynchrone [29].

2.3.2 Réplication asynchrone :

C'est le mode de distribution dans lequel certaines sous-opérations locales effectuées suite à une mise à jour globale sont accomplies dans des transactions indépendantes en temps différé. Le temps de mise à jour des copies peut être plus au moins différé : les transactions de report peuvent être lancées dès que possible ou à des instants fixes, par exemple le soir ou en fin de semaine [30]. L'avantage est la possibilité de mettre à jour en temps choisi des données, tout en autorisant l'accès aux versions anciennes avant la mise à niveau. L'inconvénient est que l'accès à la dernière version n'est pas garanti. La réplication asynchrone est basée sur deux approches [29] :

– **Approche basée sur l'état :** La réplique source est mise à jour, ensuite le sous- système transmet l'état sur les répliques par la fusion de l'état livré à l'état local.

– **Approche basée sur les opérations :** Le sous-système envoie l'opération de mise à jour et ses paramètres à toutes les répliques.

2.4 AVANTAGES DE LA RÉPLICATION :

La technique de réplication procure un certain nombre d'avantages qui améliorent les performances. L'amélioration de performances grâce à la réplication est en terme de :

Disponibilité des données : : Permettre à un serveur secondaire de prendre rapidement la place du serveur principal lorsque celui-ci échoue, ou alors les répliques répondent aux requêtes qui sollicitent une même donnée. S'il y a une perte de toute connectivité réseau, avoir une réplique complète signifie qu'il y a toujours accès à toutes les données

Amélioration des performances d'accès aux données (meilleur temps de réponse, diminution du temps de transfert des données, etc.).

Autonomie accrue : Les utilisateurs peuvent manipuler des copies de données hors connexion, puis propager leurs modifications aux autres bases de données lorsqu'ils sont connectés.

2.5 Travaux connexes :

Dans ce travail de fin d'étude, nous nous intéressons à la problématique de la gestion de données dans une infrastructure de type Fog. L'objectif est de minimiser le temps de repense du système. Comme la problématique que nous cherchons à résoudre est récente, et par conséquent peu de travaux existantes dans la littérature, nous commençons cette section par présenter les principaux travaux existants qui utilisent le paradigme du Fog Computing et Réplication.
En effet, ces travaux sont proches de notre problématique du point de vue du contexte de travail (c'est-à-dire le Fog computing et réplication), et des objectifs (par exemple, la réduction de temps de réponse et la latence du système) ou même les environnements de simulation utilisés (par exemple : iFogSim)

2.5.1 Fog Computing and Its Role in the Internet of Things [31] :

L'article [31] de Bonomi a introduit le premier concept de Fog Computing en 2012. Dans ce travail, L'importance d'un nouveau paradigme informatique a été démontrée dans le brouillard. Selon les auteurs, les principales caractéristiques du Fog Computing à prendre en compte sont (i) une faible latence et une connaissance de l'emplacement, (ii) une haute distribution géographique, (iii) des réseaux de capteurs à grande échelle, (iv) le soutien à la mobilité et (v) l'hétérogénéité des dispositifs. La structure de ce modèle proposé vise à prendre en compte ces caractéristiques et, par conséquent, elle devrait se composer des couches suivantes, du bas vers le haut de l'infrastructure : les objets connectés, le réseau central où les nœuds Fog sont déployés, et le réseau de distribution du Cloud. Les auteurs mentionnent trois principaux cas d'application adaptés à l'architecture proposée. Tout d'abord, les voitures connectées, où les véhicules communiquent entre elles avec des feux de circulation intelligents et des unités en bordure de route situées le long des routes. Le deuxième cas mentionné est celui des smart grids. Ce dernier représente un système d'alimentation électrique intelligent avec de nombreux

fournisseurs et consommateurs répartis géographiquement. Troisièmement, les réseaux de capteurs et d'actionneurs sans fil sont constitués de nœuds de capteurs et d'actionneurs largement répartis géographiquement. Les trois scénarios exigent un traitement en temps réel, une faible consommation d'énergie et un système distribué à tolérance de pannes, et bénéficieraient ainsi d'une solution informatique sophistiquée pour le traitement de ce paradigme "fog computing".

2.5.2 Data Placement and Task Scheduling Optimization for Data Intensive Scientific Workflow in Multiple Data Centers Environment[32] :

Dans ce travail , l'auteur analyse d'abord " the challenges of scientific workflow applications" dans un environnement multiple data centers et proposer une première stratégie de placement de données basée sur la relation entre l'ensemble de données d'entrée et les tâches dans le workflow , La dépendance des données et la taille de l'ensemble de données sont prises en compte pour établir un modèle vectoriel multidimensionnel pour chaque ensemble de données. Ensuite, il utilise algorithme de clustering k-means pour distribuer les ensembles de données d'entrée dans différents centres de données avec les ensembles de données les plus connexes placés ensemble. Au moment de l'exécution des workflows scientifiques il proposer une stratégie de planification de réplication de tâches à plusieurs niveaux pour réduire le transfert de grands ensembles de données. Dans la stratégie, le coût de communication est échangé contre le coût de calcul en utilisant la réplication des tâches. il peut déduire du résultat de la simulation qu'en utilisant les deux stratégies, le transfère des données est considérablement réduite.

2.6 CONCLUSION

Dans ce chapitre, nous avons présentés le fog computing, l'architecture de ce paradigme et c'est avantages et défis ,et ensuite nous avons présnté la réplication de données et les types de la réplication et quelque travaux connexes. Dans le chapitre suivant on va presentes la description et la modélisation de noter approche proposee

Chapitre 3

Desciption et Modélisation de l'approche proposée

3.1 Introduction

Dans les chapitres précédents, nous avons présenté les différents concepts de notre domaine d'intérêt en relation avec le paradigme du Fog Computing. De plus, nous avons exploré quelques travaux qui ont été proposés dans la littérature et qui ont des points communs avec notre projet en terme de contexte, d'objectifs et d'environnement. Notre objectif principal est de proposer et d'implémenter une nouvelle stratégie de gestion des données dans un environnement de Fog Computing, et cela dans le but d'améliorer les performances telles que l'utilisation du reseau et le temps de réponse .
Le présent chapitre permet d'expliquer notre architecture et la stratégie proposée ainsi notre démarche en décrivant ses différentes phases et les algorithmes nécessaires pour son fonctionnement.

3.2 Architecture hiérarchique proposée

Dans notre strategie,nous avons utilisé une architecture hiérarchique composée de 3 niveau :

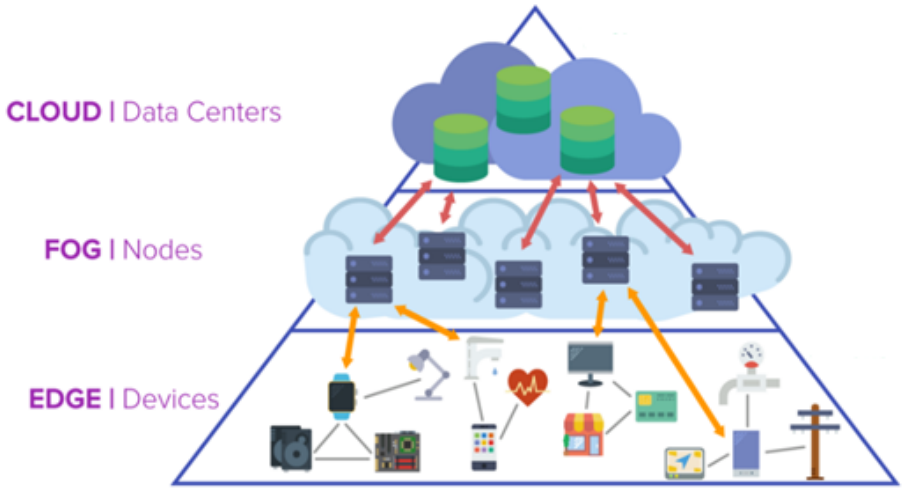

FIGURE 3.1 – Architecture hiérarchique

Le 1ère nivean Il se compose d'objets connectés IOT (Internet of Things), qui collecte des informations telles que la température, la luminosité, la vitesse, l'accélération, la position GPS et peut contrôler certains appareils simples tels que les stores des maisons, barrières, portes, interrupteurs ou robots d'intervention plus complexes.
Ces objets sont connectés via WiFi ou d'autres technologies de communication locales et permettent la communication avec le deuxième étage de la pyramide.

Le 2eme niveau Il est de façon courante appelé GATEWAY ou passerelle : ils concentrent les informations localement reçues (maison, immeuble, usine ou bâtiment, parcelle agricole), les regroupent et les transmettent via Internet au troisième étage de la pyramide.

Le 3eme niveau Il est composé d'outils, de serveurs informatiques et de nombreux logiciels de traitement de l'information (Data Management (DTM), Business Intelligence (BI), Intelligence Artificielle (IA), Datamining, Data Science...) qui permettent de traiter un grand volume de données hétérogènes et de déterminer des prédictions et des actions à réaliser.

3.3 APPROCHE PROPOSÉE :

Notre stratégie est basée sur trois phases, la premier elle permet de construire les vecteurs des données et de les normaliser, la deuxième phase permet de clustériser les données à travers l'algorithme de k-maens, la derniére phase est une extension d'une approche existante [12] qui est basée sur la technique de réplication.

3.3.1 PHASE DE CONSTRUCTION ET NORMALISATION DES VECTEURS :

3.3.1.1 ETAP 1 : construction des vecteurs

Pour décrire les différentes phases de notre stratégie, nous allons utilliser les notation suivantes :
Ensemble de données initial $D = [d_1, d_2, d_3 \ldots]$ désigne tous les ensembles de - - Données d'entrée requis avant l'exécution de l'application de workflow.

Ensemble des tuples (tâches) $T = [t_1, t_2, t_3 \ldots]$ désigne toutes les tuples de l'application de workflow.

Dépendance des données et modèle de taille des données. Selon la relation entre l'ensemble de données et la tâche dans le DAG[directed acyclic graph], un vecteur multidimensionnel est établi pour chaque donnée , $d_i = (f_{it_1}, f_{it_2}, f_{it_3}..f_{it_n})$.

Où d_i désigne le i-ème ensemble de données d'entrée dans le DAG, N est le nombre total de tâches. f_{it_n} indique la taille d_i est utilisé par le tache t_n, si d_i est utiliser par t_n, la valeur de f_{it_n} c'est la taille de d_i , sinon, sa valeur est zéro. Il peut être exprimé par la formule suivante :

$$f_{it_n} = |d_i \in t_n(input)| * size(d_i)$$

$t_n(input)$, C'est l'ensemble des ensembles de données d'entrée requis par t_n.
Dans le modèle vectoriel multidimensionnel, la dépendance des données est reflétée par le nombre de valeurs non nulles dans le vecteur, Si plusieurs données sont traités par la même tâche, les attributs correspondants aux valeur de ses vecteurs multidimensionnels égale à la taille de l'ensemble de données. Par exemple :

d_0 =1GB d_1 =1GB d_2 =3GB d_3 =1GB d_4 =5GB d_5 =2GB d_6 =1GB d_7 =3GB

d_0 = (1,0,0,0,0,0)
d_1 = (1,1,0,0,0,0)
d_2 = (0,3,3,3,0,0)
d_3 = (1,1,0,0,0,0)
d_4 = (0,0,5,5,0,0)
d_5 = (0,0,0,2,2,0)
d_6 = (0,0,0,1,1,0)

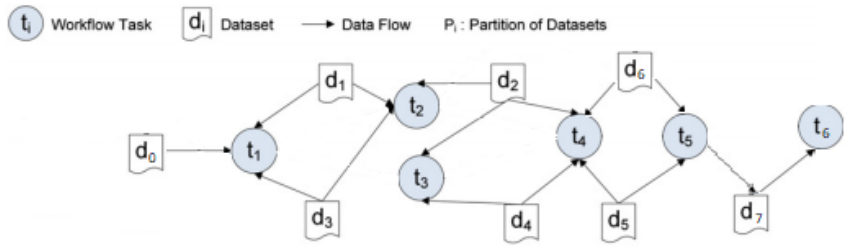

FIGURE 3.2 – exemple de placement de données.

d_7 = (0,0,0,0,3,3)
On peut trouver que $f_{1t_1} = f_{1t_2} = f_{1t_3} \neq 0$ et $f_{2t_1} = f_{2t_2} = s_{2t_3} \neq 0$ ce qui indique que d_1 et d_2 sont simultanément requis par t_1, t_2 et t_3 . Cela reflète la dépendance des données entre d_1 et d_2,et la valeur de 1 et 3 reflètent la taille des données.

3.3.1.2 ETAP 2 : la normalisation des vecteurs

Data Mining peut générer des résultats efficaces si la normalisation est appliquée à l'ensemble de données. C'est un processus utilisé pour normaliser tous les attributs de l'ensemble de données et leur donner un poids égal pour améliorer la précision du résultat. L'algorithme K-Means utilise la distance euclidienne qui est très sujette aux irrégularités dans la taille de diverses caractéristiques [11]. Il existe différentes méthodes de normalisation des données telles que Min-Max, Z-Score et Decimal Scaling. La meilleure méthode de normalisation dépend des données à normaliser. Ici, nous avons utilisé la technique de normalisation Min-Max dans notre algorithme car notre ensemble de données est limité et n'a pas beaucoup de variabilité entre le minimum et le maximum. La technique de normalisation Min-Max effectue une transformation linéaire sur les données. Dans cette méthode le vecteur multidimensionnel normalisé :

$$d'_i = (f'_{it_1}, f'_{it_2}, f'_{it_3}..f'_{it_n})$$

Avec :

$$f'_{it_n} = \frac{f_{it_n} - min(f_{it_n})}{max(f_{it_n}) - min(f_{it_n})}$$

Par exemple :
Exemple figure 1

$f'_{2t_1} = \frac{0-0}{3-0} = 0$

$f'_{2t_2} = \frac{3-0}{3-0} = 1$

$f'_{2t_3} = \frac{3-0}{3-0} = 1$

$f'_{2t_4} = \frac{3-0}{3-0} = 1$

$f'_{2t_5} = \frac{0-0}{3-0} = 0$

$f'_{2t_6} = \frac{0-0}{0-0} = 0$

ça donne

$d'_2 = (0, 1, 1, 1, 0, 0)$

les résultats de tout l'exemple

$d'_0 = (1, 0, 0, 0, 0, 0)$
$d'_2 = (1, 1, 0, 0, 0, 0)$
$d'_2 = (0, 1, 1, 1, 0, 0)$
$d'_4 = (0, 0, 1, 1, 0, 0)$
$d'_5 = (0, 0, 0, 1, 1, 0)$
$d'_6 = (0, 0, 0, 1, 1, 0)$
$d'_7 = (0, 0, 0, 0, 1, 1)$

3.3.2 PHASE DE CLUSTERING AVEC K-MEANS

L'algorithme k-means mis au point par McQueen en 1967, un des plus simples algorithmes d'apprentissage non supervisé, appelée algorithme des centres mobiles, il attribue chaque point dans un cluster dont le centre (centroïde) est le plus proche. Le centre est la moyenne de tous les points dans le cluster, ses coordonnées sont la moyenne arithmétique pour chaque dimension séparément de tous les Points dans le cluster c'est à dire chaque cluster est représentée par son centre de gravité.
Retournons à notre proposition étant donné que la relation entre les ensembles de

données est maintenant reflétée dans le vecteur, la question de trouver la dépendance entre ces ensembles de données est transformée en regroupement vectoriel. L'algorithme K-means est adopté pour regrouper des ensembles de données hautement dépendants. Ici, nous utilisons la similarité vectorielle pour mesurer le degré de dépendance des données, qui est représenté par la distance entre deux vecteurs d_i et d_j. Plus la distance des deux vecteurs est petite, plus ils sont similaires. La formule de distance euclidienne est utilisée dans notre modèle où

$$d(d'_i, d'_j) = \sqrt{\sum_{k=1}^{N} (f'_{it_k} - f'_{jt_k})^2}$$

Ensuite, la fonction de critère d'erreur au carré est utilisée pour déterminer si le regroupement des ensembles de données est stable.

$$E = \sum_{i=0}^{k} \sum_{p \in D_i} |p - m_i|^2$$

Où m_i représente le centre géométrique du cluster i, et D_i désigne une liste de sous-ensembles de données comprenant des ensembles de données à forte dépendance.

les resultats apres le regroupment

P1 (d0, d1, d3)

P2 (d2, d4)

P3 (d5, d6)

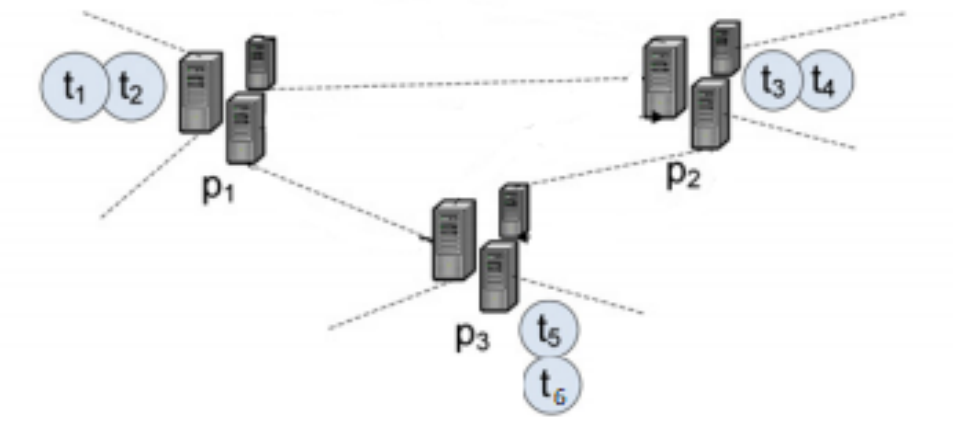

FIGURE 3.3 – l'exemple après le clustering avec k-means algorithme

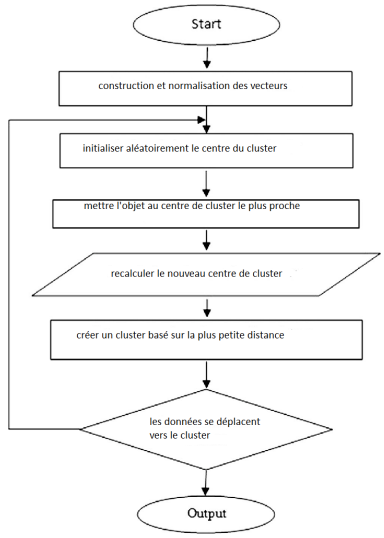

FIGURE 3.4 – organigramme de l'algorithme de k-means

3.3.3 PHASE DE RÉPLICATION DYNAMIQUE :

Pour réduire la transmission de données et améliorer notre proposition, Nous avons introduit un service de réplication dynamique qui permet la réplication de certains des ensembles de données les plus fréquemment utilisés pour réduire le trafic entre les nœuds. Son principe est décrit en deux étapes :

3.3.3.1 CRÉATION DES RÈPLIQUES

Pour calculer le nombre optimal de répliques à créer pour un objet, nous nous sommes appuyés sur la modélisation du taux d'utilisation de cet objet <seuil>, Nous avons étudié les stratégies existantes et choisi la stratégie de réplication racine carrée[33]. Dans cette stratégie, Le nombre de répliques d'un objet est proportionnel à la racine carrée du taux d'utilisation de cet objet. A chaque période, si cette condition est cochée, L'algorithme commence le processus de vérification du besoin de réplication : il calcule pour chaque ressource la racine carrée du nombre de demandes demandant cette ressource. Si cette valeur est supérieure au nombre total de répliques de cette ressource, le processus de réplication est lancé.

$$\lambda_i = \sqrt{AF_i}$$

Avec :
AF_i : le nombre d'accès de la donnée i ;
λ_i : seuil de réplication.

3.3.3.2 NOMBRE DE RÉPLIQUE

Dans cette partie le nombre de réplique est augmenté jusqu'à ce que le seuil soit inférieur de lui.

3.3.3.3 PLACEMENT DES RÉPLIQUES

Une fois les répliques créées, elles doivent être placées dans des fog, sous réserve de certains critères, Tout d'abord, nous devons déterminer quels nœuds ont suffisamment d'espace de stockage pour héberger la réplique qui a été créée, Ensuite, nous sélectionnons les nœuds qui ne contiennent pas déjà les données répliquées, Enfin, placez la réplique dans le nœud qui demande fréquemment ces données.

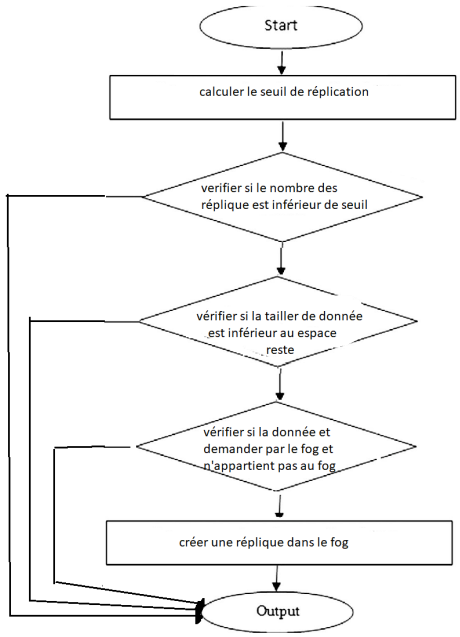

FIGURE 3.5 – organigramme de l'algorithme de réplication

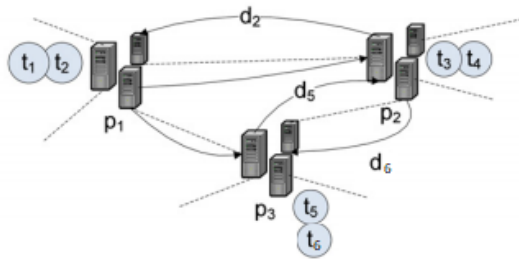

FIGURE 3.6 – l'example apres l'application de la réplication

3.4 CONCLUSION

Au cours de ce chapitre, nous avons décrit les méthodes qu'on s'est base et utilise dans notre projet, ou nous avons proposés les trois stratégies, un ensemble de pseudocodes pour bien illustré notre approche, Dans le dernier chapitre, on va présenter les résultats obtenus de plusieurs expérimentations et simulations.

Chapitre 4

IMPLÉMENTATION ET RÉSULTATS

4.1 INTRODUCTION

Afin d'incarner, de valider et d'évaluer notre approche réplication, nous avons conçu un simulateur qui reflète le fonctionnement de notre proposition, et réalisé une série d'expériences dont les résultats et les interprétations font l'objet de ce chapitre. Nous commencerons par décrire l'environnement dans lequel nous avons construit notre simulateur, puis nous discuterons et analyserons les résultats que nous avons obtenus.

4.2 Langage et environnement de développement

L'environnement de développement est un facteur important qu'il faut détailler pour savoir, dans quelles situations ?, le même travail peut être reproduit. La stratégie que nous avons proposée, dans le cadre de ce projet de fin d'étude, a été implémentée et testée dans l'environnement suivant :

- Caractéristiques matérielles et logicielles de l'ordinateur utilisés :

Nous avons développé notre application sur une machine avec un processeur Intel(R) Core(TM)i3-7100h CPU, d'une vitesse de 3.0Ghz et d'une capacité mémoire de 4GB. Le simulateur est sous Windows10 de 64 bits

- Simulateur utilisé

— iFogSim

- Langage utilisé

— java

- IDE utilisé

— eclipse

4.2.1 Langage de programmation Java :

Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.Sun a été racheté en 2009 par Oracle, qui possède et gère désormais Java.

Le langage JAVA a la particularité principale que les logiciels écrits dans ce langage peuvent être très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec ou sans modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java .

Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Java Runtime Environment - Environnement d'exécution Java). Cette dernière est constituée d'une JVM (Java Virtual Machine - Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes .

Java est devenu aujourd'hui une direction incontournable dans le monde de la programmation, parmi les différentes caractéristiques qui sont attribuées à son

succès :

- L'indépendance de toute plate-forme : le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine.
- • Le code est structuré dans plusieurs classes, dont chacune traite une partie différente de la simulation.
- Il assure la gestion de la mémoire.
- Java est multitâches : il permet l'utilisation de Threads qui sont des unités d'exécution isolées.

Ainsi, une des principales raisons de ce choix est que le simulateur iFogSim est développé avec ce langage.

4.2.2 Environnement de développement

Eclipse est un environnement de développement intégré (EDI), placé en Open Source par Sun en novembre 2001. En plus de Java, Eclipse permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML téléchargeable du site : https ://Eclipse.org/downloads. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web)
. Conçu en Java, Eclipse est disponible sous Windows, Linux, Solaris, MacOS ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java).
De plus, Eclipse est écrit en Open Source, téléchargeable directement du site http ://java.sun.com. Il est puissant et compatible avec toutes les nouvelles technologies Java (les technologies Java EE, les bases de données, UML, XML, ...).

4.2.3 Simulateur iFogSim

iFogSim est un simulateur d'environnements de Fog et d'IoT . C'est une extension du simulateur d'environnements de Cloud : CloudSim . iFogSim est conçu pour pouvoir évaluer des stratégies de gestion des ressources applicables aux environnements de Fog et d'IoT en ce qui concerne leur impact sur la latence, la consommation d'énergie, la congestion du réseau et les coûts opérationnels.
iFogSim permet de simuler un environnement de Fog et d'IoT incluant des capteurs, des actionneurs, des nœuds de Fog et des centres de données de Cloud. Les applications d'IoT simulées dans iFogSim sont basées sur le modèle PerceptionTraitement-Action . Dans ce modèle, un ensemble de capteurs (ex. de température) collectent d'abord des informations relatives à l'environnement physique de l'application, puis les publient de manière périodique ou événementielle (ex. si la température dépasse un certain seuil). Ensuite, ces informations sont récupérées puis traitées par un ensemble d'instances de services d'IoT déployées dans le Fog et dans le Cloud. Enfin, conformément aux résultats du traitement, des messages sont envoyés aux actionneurs pour agir sur l'environnement physique de l'application (ex. éteindre le chauffage ou déclencher une alarme).
Afin de réaliser une simulation dans iFogSim, les utilisateurs spécifient l'infrastructure physique de système (i.e. capteurs, actionneurs, nœuds de Fog, liens réseaux, etc.), le scénario (i.e. services d'IoT, flux de données, etc.), leurs stratégies de placement et/ou d'ordonnancement de services, et le temps maximal de la simulation. iFogSim produit en sortie la valeur d'un ou de plusieurs critères (selon les objectifs des utilisateurs) parmi ceux qui sont mentionnés auparavant.
La Figure 4.1 montre l'architecture initiale d'iFogSim (i.e. sans l'extension proposée). Ce simulateur est composé : (i) d'un ensemble d'entités permettant de modéliser les équipements physiques de l'infrastructure, (ii) un ensemble d'entités permettant de modéliser les éléments logiques du scénario simulé (ex. les données et les instances des services de l'IoT), et (iii) un ensemble d'entités permettant de

gérer les ressources de traitement dans les nœuds de Fog et dans les centres de données. Ces ressources sont manipulées pour placer et ordonnancer les instances de services dans les entités physiques afin d'optimiser la latence du service, le trafic réseau, la consommation énergétique ou le coût opérationnel du système. Ci-après nous décrivons chaque ensemble d'entités définies dans iFogSim.

- Les entités physiques : ce sont les modèles des équipements physiques trouvés dans une infrastructure de Fog (ex. serveurs, capteurs, actionneurs etc.).
  1. fogdevice :cette entité modélise les nœuds de Fog (par exemple les switchs, les routeurs, les stations de base, les passerelles, ...). Elle spécifie les caractéristiques matérielles d'un nœud de Fog comme le modèle du processeur, la taille de la RAM, la capacité du stockage et la bande passante du réseau.
  2. Capteur : cette entité modélise les différents capteurs déployés dans l'infrastructure simulée. Elle spécifie les attributs d'un capteur allant de sa connectivité réseau jusqu'aux données capturées.
  3. Actionneur : cette entité modélise les actionneurs et les effets de leur action sur l'environnement simulé. Elle spécifie les attributs d'un actionneur comme sa connectivité, sa latence et la passerelle associée.

—

- un ensemble d'entités permettant de modéliser les éléments logiques du scénario simulé (par exemple les instances des services de l'IoT et leurs dépendances, ...),
  1. Tuple : cette entité représente l'unité fondamentale des communications entre les entités physiques iFogSim. Un Tuple est caractérisé par son type (par exemple température), le nœud source, le noeud destinataire, la capacité du traitement de données demandée (en millions d'instructions) et la taille des données encapsulées (en octets).
  2. AppModule : cette entité représente les unités de traitement du scénario simulé qui sont les instances de services de l'IoT. Ces instances de services sont déployées dans les nœuds de Fog. Pour chaque donnée d'entrée (ou Tuple), une AppModule s'occupe de son traitement et génère des données de sortie qui sont envoyées par la suite à une ou à plusieurs AppModules. Le taux de production des données de sortie est exprimé par un modèle de sélection des données déployé dans le simulateur iFogSim. Par exemple, un taux de sélection de 10Tuple) de sortie pour 10 données d'entrée.
  3. Actionneur : cette entité modélise les actionneurs et les effets de leur action sur l'environnement simulé. Elle spécifie les attributs d'un actionneur comme sa connectivité, sa latence et la passerelle associée.
  4. AppEdge : cette entité représente les dépendances de données existantes entre deux AppModules (c'est-à-dire les services). De plus, une AppEdge

spécifie le mode de production de données parmi les deux modes possibles : périodique ou événementiel. Le modèle de dépendance de données du scénario simulé est représenté par un graphe orienté acyclique composé d'un ensemble d'AppModules interconnectées entre elles. Les noeuds du graphe représentent les AppModules et les arêtes représentent les flux de données.

- Les entités de gestion de ressources : ce sont des classes représentant les stratégies de gestion de ressources physiques et logiques dans iFogSim (par exemple allocation, ordonnancement, migration, ...).
  1. AppModule Placement : cette entité définit un ensemble de méthodes permettant de gérer le placement des AppModules dans l'infrastructure simulée. Ainsi, les utilisateurs utilisent ces méthodes pour définir leur stratégie de placement des AppModules afin d'optimiser un ou plusieurs critères parmi ceux qui sont cités au début de cette section. Une fois les AppModules placés dans les nœuds de Fog, ils sont ordonnancés suivant l'entité AppModule Scheduler qui
  2. AppModule Mapping : cette entité définit pour chaque instance de service d'IoT, le nœud de Fog ou le centre de données qui l'héberge. Cette entité est configurée suivant la politique implantée dans l'entité de gestion de ressources AppModule Placement.
  3. AppModule Scheduler : cette entité définit un ensemble de méthodes pour gérer l'ordonnancement des AppModules dans un nœud de Fog ou un centre de données. Les utilisateurs utilisent ces méthodes pour définir leur stratégie d'ordonnancement des AppModules. Le mode d'ordonnancement mis en oeuvre par défaut dans iFogSim subdivise équitablement les ressources de traitement entre l'ensemble des AppModules hébergés dans une entité physique de l'infrastructure. est définie par la suite.

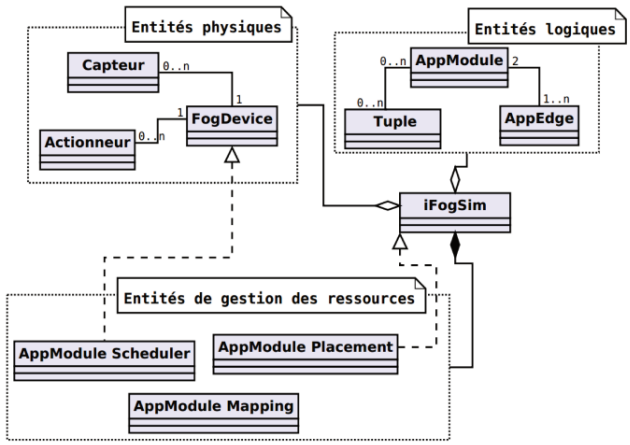

FIGURE 4.1 – Diagramme de classes de simulateur iFogSim.

4.3 IMPLEMENTATION :

Dans cette partie, nous allons nous intéresser à la démonstration de notre application à travers un exemple en faisant référence à quelques interfaces graphiques. le diagramme de cas d'utilisation suivant décrit la méthode d'utilisation de notre application.

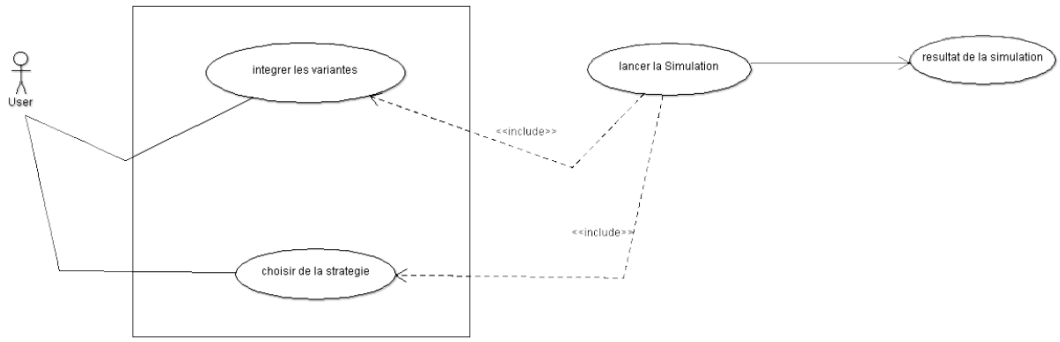

FIGURE 4.2 – Diagramme de cas d'utilisation.

4.3.1 INTERFACE PRINCIPALE :

Nous avons créé une interface qui facilite l'accès au simulateur. L'interface doit faire appel à iFogSim ainsi qu'aux différentes approches qui se trouvent dans différents packages.

4.3.2 Configuration de simulation :

Avant toute simulation, la première action à effectuer est la configuration de l'infrastructure simulée. Nous avons implémenté un panneau de configuration. Ce

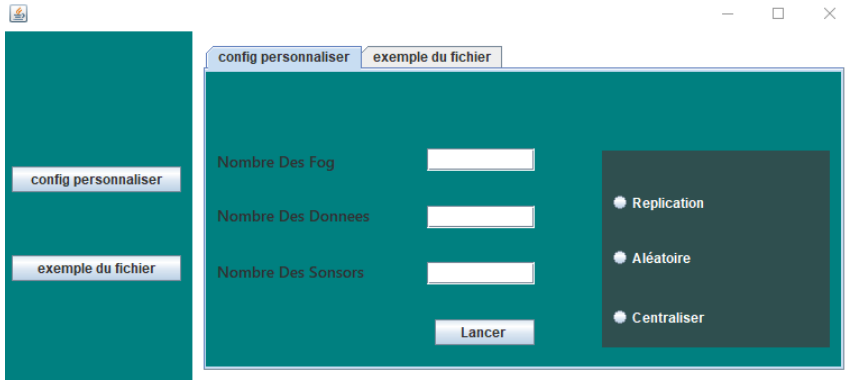

FIGURE 4.3 – Panneau de configuration de la topologie.

panneau permet la personnalisation des différents paramètres de configuration, le nombre de fog, le nombre de sonsors par fog et le nombre de donnees.

4.3.3 Lancement de simulation et visualisation des résultats

Pour mettre en valeur les apports de notre approches, nous allons se focaliser sur la métrique de temps de réponse. Dans le but d'étudier le comportement de notre propositions et d'analyser ses résultats obtenus par la simulation, nous allons les comparer d'autre stratégie. Plusieurs séries de simulation ont été lancées.

4.3.3.1 SIMULATION 1 :

Dans cette simulation, nous avons créé 5 fog , le nombre de requête est fixé a 100 requêtes et un nomber des données est fixé a 50 . Cette simulation consiste à varier la taille de la donnée (200-400,400-800,800-1200,1200-2400) .

	200-400	400-800	800-1200	1200-2400
stratégie de replication	202	208	339	336
stratégie de placement aléatoir	540	463	739	720
cloud	580	530	739	800

TABLE 4.1 – Impact de la taille du donnée sur le temps d'exécution

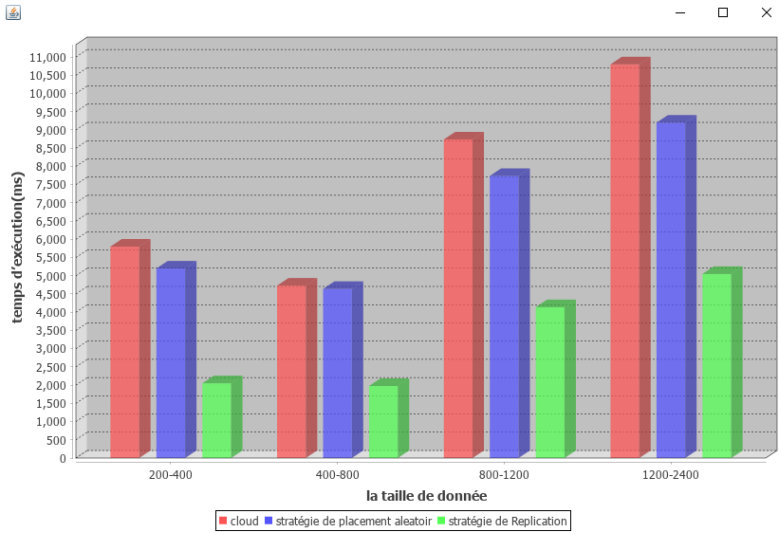

FIGURE 4.4 – Impact des tailles des donnees sur le temps d'exécution.

Le tableau 4.1 est un tableau de comparaison qui se repose sur trois stratégies (Placement aléatoire , la recuperation des donnees du cloud , Réplication) pour une seul variant (la taille du fichier), ou nous pouvons déduire que le temps de réponse de notre stratégie est beaucoup plus réduit par rapport aux deux autres stratégies.

4.3.3.2 SIMULATION 2 :

Dans cette simulation,nous avons créé 5 fegs, nous avons créé aussi 50 données avec des tailles entre(1200,2400) . Cette simulation consiste à varier le nombre des requétes par fog (50, 100, 200, 300) ;

	50	100	200	300
stratégie de replication	2060	3836	11072	32018
stratégie de placement aléatoir	4436	6289	22390	54836
cloud	4363	7099	23390	52804

TABLE 4.2 – Impact du nombre des requétes sur le temps d'exécution.

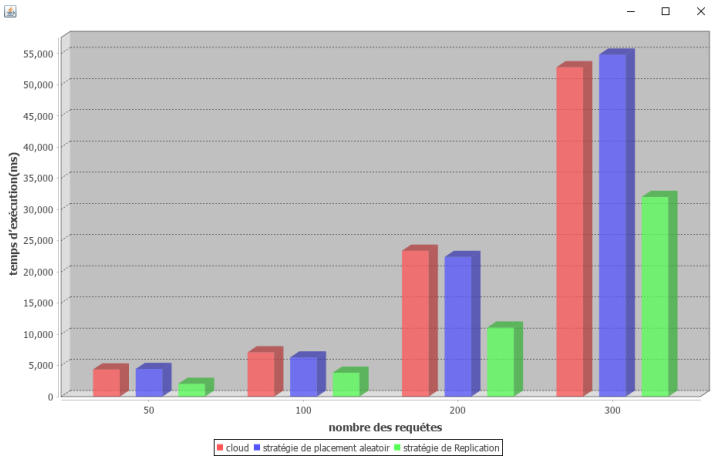

FIGURE 4.5 – Impact du nombre des requêtes sur le temps d'exécution.

4.3.4 Utilisation totale du réseau

Dans cette série de simulations représentée par la Figure 4.7, où l'axe des x représente la variation du nombre des requêtes et l'axe des y représente la quantité de données transmises sur le réseau mesurée en KO. Nous remarquons une augmentation de l'utilisation du réseau dans les trois approches. Notant que cette métrique est disponible dans le simulateur iFogSim. Nous pouvons justifier cela par l'augmentation du nombre des sauts empruntés par une requête utilisateur. Nous remarquons aussi qu'avec l'augmentation du nombre de requêtes la différence entre les courbes augmente et la courbe de l'approche proposée «stratégie de replication» se trouve au-dessous des deux courbes des autres approches avec une faible pente. Ce qui permet de confirmer l'efficacité de notre algorithme proposé de placement de répliques.

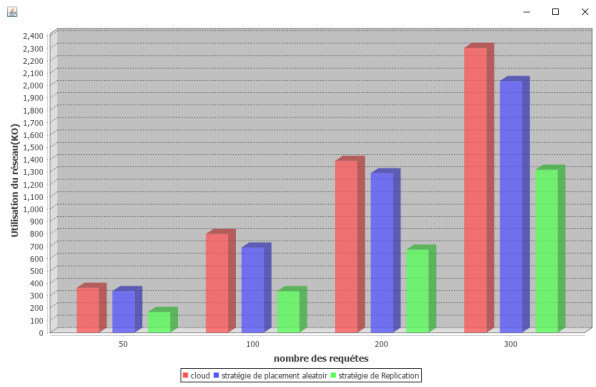

FIGURE 4.6 – Impact du nombre des requêtes sur Utilisation du réseau.

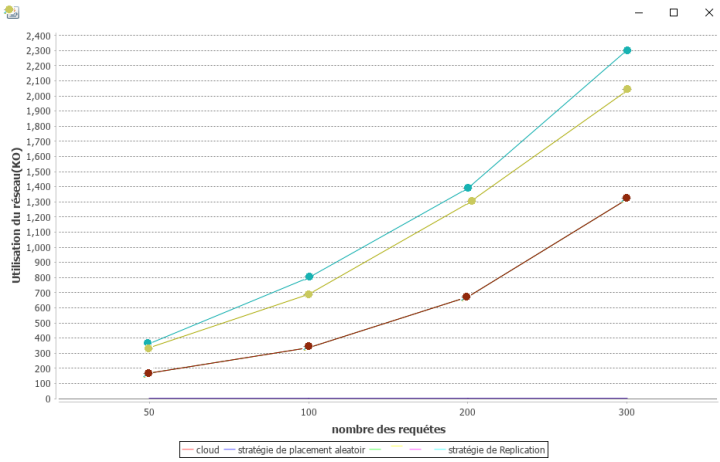

FIGURE 4.7 – Utilisation du réseau.

Nous remarquons la différence entre la stratégie de replication avec les deux autres approches en pourcentage (%), où nous pouvons déduire que notre approche de replication a réduit l'Utilisation du réseau avec un gain moyen de $22,94\%$ par rapport à l'approche centralisée et un gain moyen de $20,58\%$ par rapport à l'approche de placement aléatoir . Donc notre proposition est bien meilleure par rapport aux autres approches puisqu'elle permet de Utilise moins le réseau.

4.4 Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre application ainsi que les résultats obtenus. Aussi, Nous avons effectué plusieurs séries de simulations pour comparer notre approche avec l'approche Aléatoire et l'approche centralisée tout en variant différents paramètres comme : le nombre des donnees , le nombre de fogs. Les résultats de la comparaison ont montré que notre approche permettant de réduire temps de réponse, de minimiser l'utilisation du réseau.

Conclusion générale

A travers cette thèse, nous étudions le problème de la réplication des données dans le fog computing. Plusieurs stratégies de réplication ont été proposées dans la littérature : elles visent principalement à créer et à placer de manière efficace les répliques afin qu'une tâche puisse trouver les données nécessaires à l'exécution de sa requête. Les approches sur lesquelles se base ces stratégies diffèrent pour pouvoir offrir un compromis entre les objectifs conflictuels de la réplication, à savoir, l'augmentation du niveau de disponibilité et la tolérance aux pannes.
Le problème de réplication de données dans le Cloud computing est assez complexe, il présente plusieurs défis, tels que la capacité de stockage . Une stratégie efficace de réplication doit s'adapter aux comportement dynamique des utilisateurs en évitant de nuire aux performances du système en le surchargeant de transfert de données superflu.
D'autre part, le choix de l'architecture du Fog est un facteur crucial dont dépendent les performances attendues du système. Dans ce contexte nous avons proposé sur une solution qui permet de garantir la disponibilité des donnée dans le Fog computing .

PERSPECTIVES

Ce travail a permis d'ouvrir quelques perspectives intéressantes que nous récapitulons dans les points suivants :

1. Traiter le coût de payement.
2. Intèger notre approche dans un environnement de fog réel.
3. Amélioration de la stratégie.

Bibliographie

[1] Infso d.4 networked enterprise rfid infso g.2 micro nanosystems, in : Cooperation with the working group rfid of the etp eposs, "internet of things in 2020, roadmap for the future," 2008.

[2] Mohammad Aazam and Eui-Nam Huh. E-hamc : Leveraging fog computing for emergency alert service. In *2015 ieee international conference on pervasive computing and communication workshops (percom workshops)*, pages 518–523. IEEE, 2015.

[3] Younes Ait Mouhoub, Mawloud Omar, Fatah Bouchebbah, et al. *Proposition d'un modèle de confiance pour l'internet des objets.* PhD thesis, Université A/Mira de Bejaia, 2015.

[4] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. Internet of things : A survey on enabling technologies, protocols, and applications. *IEEE communications surveys & tutorials*, 17(4) :2347–2376, 2015.

[5] Lorenzo Alvisi and Keith Marzullo. Message logging : Pessimistic, optimistic, causal, and optimal. *IEEE Transactions on Software Engineering*, 24(2) :149–159, 1998.

[6] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things : A survey. *Computer networks*, 54(15) :2787–2805, 2010.

[7] Pierre-Jean Benghozi, Sylvain Bureau, and Françoise Massit-Folea. L'internet des objets. quels enjeux pour les européens ? 2008.

[8] Pierre-Jean Benghozi, Sylvain Bureau, and Françoise Massit-Folea. L'internet des objets. quels enjeux pour les européens ? 2008.

[9] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16, 2012.

[10] Nader Daneshfar, Nikolaos Pappas, Valentin Polishchuk, and Vangelis Angelakis. Service allocation in a mobile fog infrastructure under availability and qos constraints. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, 2018.

[11] Amir Vahid Dastjerdi, Harshit Gupta, Rodrigo N Calheiros, Soumya K Ghosh, and Rajkumar Buyya. Fog computing : Principles, architectures, and applications. In *Internet of things*, pages 61–75. Elsevier, 2016.

[12] Chunye Gong, Jie Liu, Qiang Zhang, Haitao Chen, and Zhenghu Gong. The characteristics of cloud computing. In *2010 39th International Conference on Parallel Processing Workshops*, pages 275–279. IEEE, 2010.

[13] Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, and Tie Qiu. Survey on fog computing : architecture, key technologies, applications and open issues. *Journal of network and computer applications*, 98 :27–42, 2017.

[14] Lihong Jiang, Li Da Xu, Hongming Cai, Zuhai Jiang, Fenglin Bu, and Boyi Xu. An iot-oriented data storage framework in cloud computing platform. *IEEE Transactions on Industrial Informatics*, 10(2) :1443–1451, 2014.

[15] Redowan Mahmud, Kotagiri Ramamohanarao, and Rajkumar Buyya. Latency-aware application module management for fog computing environments. *ACM Transactions on Internet Technology (TOIT)*, 19(1) :1–21, 2018.

[16] Naima Makhloufi, Raouf Achour, Abdellah Boukerram, et al. *Authentificaiton Dans L'iot*. PhD thesis, Université abderrahmane mira béjaia, 2017.

[17] Apostolos Malatras. State-of-the-art survey on p2p overlay networks in pervasive computing environments. *Journal of Network and Computer Applications*, 55 :1–23, 2015.

[18] Ruben Mayer, Leon Graser, Harshit Gupta, Enrique Saurez, and Umakishore Ramachandran. Emufog : Extensible and scalable emulation of large-scale fog computing infrastructures. In *2017 IEEE Fog World Congress (FWC)*, pages 1–6. IEEE, 2017.

[19] Daniele Miorandi, Sabrina Sicari, Francesco De Pellegrini, and Imrich Chlamtac. Internet of things : Vision, applications and research challenges. *Ad hoc networks*, 10(7) :1497–1516, 2012.

[20] Vasileios Moysiadis, Panagiotis Sarigiannidis, and Ioannis Moscholios. Towards distributed data management in fog computing. *Wireless Communications and Mobile Computing*, 2018, 2018.

[21] Vinod Pande, Chetan Marlecha, and Sangramsing Kayte. A review-fog computing and its role in the internet of things. *International Journal of Engineering Research and Applications*, 6(10) :2248–96227, 2016.

[22] Kavitha Ranganathan and Ian Foster. Identifying dynamic replication strategies for a high-performance data grid. In *International Workshop on Grid Computing*, pages 75–86. Springer, 2001.

[23] T Venkat Narayana Rao, Amer Khan, M Maschendra, and M Kiran Kumar. A paradigm shift from cloud to fog computing. *International Journal of Science, Engineering and Computer Technology*, 5(11) :385, 2015.

[24] Yogesh Simmhan. Big data and fog computing. *arXiv preprint arXiv :1712.09552*, 2017.

[25] I Smith, K Sakamura, A Furness, R Ma, YW Kim, E Walk, C Harmon, P Chartier, P Guillemin, and D Armstrong. Rfid and the inclusive model for the internet of things. *Final report, CASAGRAS EU Framework*, 7, 2009.

[26] Luis M Vaquero and Luis Rodero-Merino. Finding your way in the fog : Towards a comprehensive definition of fog computing. *ACM SIGCOMM computer communication Review*, 44(5) :27–32, 2014.

[27] Aidmar Wainakh, Alejandro Sanchez Guinea, Tim Grube, and Max Mühlhäuser. Enhancing privacy via hierarchical federated learning. In *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 344–347. IEEE, 2020.

[28] Mingjun Wang, Jinghui Zhang, Fang Dong, and Junzhou Luo. Data placement and task scheduling optimization for data intensive scientific workflow in multiple data centers environment. In *2014 Second International Conference on Advanced Cloud and Big Data*, pages 77–84. IEEE, 2014.

[29] Mark Weiser. The computer for the 21 st century. *Scientific american*, 265(3) :94–105, 1991.

[30] Shanhe Yi, Zijiang Hao, Zhengrui Qin, and Qun Li. Fog computing : Platform and applications. In *2015 Third IEEE workshop on hot topics in web systems and technologies (HotWeb)*, pages 73–78. IEEE, 2015.

[31] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing : concepts, applications and issues. In *Proceedings of the 2015 workshop on mobile big data*, pages 37–42, 2015.

[32] Esma Yildirim, Engin Arslan, Jangyoung Kim, and Tevfik Kosar. Application-level optimization of big data transfers through pipelining, parallelism and concurrency. *IEEE Transactions on Cloud Computing*, 4(1) :63–75, 2015.

[33] Bellounar Fatima Zohra. *Stratégies Efficaces de Réplication de données sur les Grilles*. PhD thesis, Université Badji Mokhtar-Annaba, 2014.