



Faculté des Sciences Exactes et d'Informatique
Département de Mathématiques et informatique
Filière : Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'Obtention du Diplôme de Master en Informatique

Option : **Ingénierie des Systèmes d'Information**

Présenté par :

Benchikh Nesrine

Bekhedda Sara

THEME :

Allocation des ressources dans le Fog Computing

Soutenu le : 02 juin 2024

Devant le jury composé de :

Dr Midoun Mohamed	MCA	Université de Mostaganem	Président
Dr Filali Fatima.Z	MCB	Université de Mostaganem	Examinateur
Dr Benidris Fatima.Z	MCA	Université de Mostaganem	Encadreur

Année Universitaire 2023-2024

Résumé

Le nombre d'applications basées sur l'Internet des objets (IoT) est en augmentation, en plus, le transfert de toutes leurs données associées vers le Cloud nécessite plus de latence, d'énergie, de bande passante et de coût. Dans de tels cas, la couche de Fog Computing, en tant que nouveau paradigme informatique, vise à étendre l'architecture du Cloud computing vers un réseau de nœuds périphériques proches des utilisateurs et prend en charge les IoTs aux ressources limitées. Dans ce travail, nous proposons une stratégie d'allocation de ressources qui permet de répondre aux exigences des utilisateurs IoT en termes de disponibilité des ressources demandées et de performance tout en prenant compte de la contrainte de l'énergie qui peut influencer sur le processus d'allocation. . Nous avons évalué notre stratégie en utilisant le simulateur iFogSim qui offre une plateforme flexible ouverte pour créer et simuler des infrastructures Fog. Ainsi, nous avons réalisé plusieurs séries d'expériences en variant plusieurs paramètres et évaluant ainsi diverses métriques telles que la satisfaction des requêtes, le temps de réponse, le taux d'allocation des ressources et la consommation d'énergie.

Mots-clés : Allocation des ressources, Fog computing, IoT, iFogSim,

Abstract

The number of Internet of Things (IoT) based applications is increasing, in addition, transferring all their associated data to the Cloud requires more latency, energy, bandwidth and cost. In such cases, the Fog Computing layer, as a new computing paradigm, aims to extend the Cloud computing architecture to a network of edge nodes close to users and supports resource-constrained IoTs. In this work, we propose a resource allocation strategy which makes it possible to meet the requirements of IoT users in terms of availability of requested resources and performance while taking into account the energy constraint which can influence the allocation process. . We evaluated our strategy using the iFogSim simulator which offers a flexible open platform for creating and simulating Fog infrastructures. Thus, we carried out several series of experiments by varying several parameters and thus evaluating various metrics such as query satisfaction, response time, resource allocation rate and energy consumption.

Keywords: Resource allocation, Fog computing, IoT, iFogSim,

ملخص

عدد التطبيقات المعتمدة على الإنترنت للأشياء في ازدياد، ومع ذلك، فإن نقل جميع البيانات المتعلقة بها إلى السحابة Fog يتطلب المزيد من التأخير والطاقة وعرض النطاق الترددي والتكلفة. في مثل هذه الحالات، تهدف طبقة Computing ، كما يعتبر نموذجًا جديدًا للحوسبة، إلى توسيع هيكلية الحوسبة السحابية نحو شبكة من العقد الطرفية IoT ذات الموارد المحدودة. في هذا العمل، نقدم استراتيجية لتخصيص الموارد IoT المجاورة للمستخدمين ودعم أجهزة من حيث توفر الموارد المطلوبة والأداء مع مراعاة قيود الطاقة التي قد IoT التي تسمح بالرد على متطلبات مستخدمي الذي يوفر منصة مرنة مفتوحة iFogSim تؤثر على عملية التخصيص. لقد قمنا بتقييم استراتيجيتنا باستخدام محاكي وبالتالي، قمنا بإجراء سلسلة من التجارب باختلاف عدة معلمات وتقييم مقاييس متنوعة مثل Fog لإنشاء ومحاكاة بنى رضا الطلبات، وزمن الاستجابة، ومعدل تخصيص الموارد، واستهلاك الطاقة.

الكلمات الرئيسية: تخصيص الموارد، Fog computing, IoT, IFogSim

Remerciements

Nos remerciements s'adressent d'abord à Dieu le tout puissant pour nous avoir donné la santé et la volonté d'entamer et de terminer ce mémoire.

Aussi, ce travail ne serait pas aussi riche et n'aurait pas pu voir le jour sans l'aide et l'encadrement de Mme Benidris Fatima Zohra, on la remercie pour sa patience, sa rigueur et sa disponibilité durant notre préparation de ce mémoire.

Nous tenons à exprimer notre profonde gratitude à tous ceux qui ont contribué à la réalisation de notre projet de fins d'étude.

Nous tenons également à exprimer notre reconnaissance envers nos parents ; dont le soutien constant et les encouragements indéfectibles ont été une source inépuisable de motivation. Leurs amours inconditionnels et leurs sacrifices ont été les piliers sur lesquels on a pu construire notre parcours éducatif.

Sans oublier non plus nos frères et sœurs qui ont toujours été là pour nous encourager, nous prodiguer des conseils et partager nos joies et nos peines. Leur soutien infaillible a été d'une importance capitale dans notre réussite académique.

Enfin, nous souhaitons exprimer notre gratitude envers nos amis qui nous ont accompagné tout au long de cette aventure. Leurs encouragements leurs discussions stimulantes et leur présence réconfortante ont été des éléments clés pour traverser les défis et les moments de doute. Leurs amitiés précieuses rendu ce parcours d'études encore plus mémorable et significatif.

Liste des figures

Figure N°	Titre de la figure	Page
Figure 1	Cloud Computing	11
Figure 2	Historique de Cloud Computing	12
Figure 3	Software as a Service (SaaS)	13
Figure 4	Plateform as a Service (PaaS)	14
Figure 5	Infrastructure as a Service (IaaS)	15
Figure 6	Modèle de déploiement d'un Cloud public	16
Figure 7	Modèle de déploiement d'un Cloud privé	17
Figure 8	Modèle de déploiement d'un Cloud hybride	17
Figure 9	Architecture de Fog Computing	21
Figure 10	Architecture de l'Iot	31
Figure 11	Gestion des ressources en Fog Computing	33
Figure 12	Schéma de l'architecture proposée	45
Figure 13	Modèle d'application	46
Figure 14	Diagramme de cas d'utilisation	47
Figure 15	Diagramme de séquence	48
Figure 16	L'organigramme de processus d'allocation	50
Figure 17	Algorithme (1)	53
Figure 18	Diagramme de classes de simulateur iFogSim.	63
Figure 19	Diagramme de cas d'utilisation du système	64
Figure 20	Graphe de configuration des requêtes satisfaite	67
Figure 21	Graphe de variation de temps de réponse	69
Figure 22	Graphe de la consommation d'énergie	70
Figure 23	Graphe de l'utilisation du réseau	72
Figure 24	Graphe de consommation des ressources (Ram,Pes)	73

Liste des tableaux

Tableau N°	Titre du tableau	Page
Tableau 1	La comparaison entre le Cloud et le Fog Compting	29
Tableau 2	Tableau des ressources d'utilisateurs	56
Tableau 3	Tableau de calcul de l'entropie et du poids	57
Tableau 4	Tableau des requêtes satisfaites	66
Tableau 5	Tableau de variation de temps de réponse	67
Tableau 6	Tableau de la consommation d'énergie	69
Tableau 7	Tableau de l'usage du réseau	71
Tableau 8	Tableau des ressources consommé	72

Liste des abréviations

Abréviation	Expression Complète	Page
Iot	Internet des objets	4
SaaS	Software as a service	9
Paas	Plateforme as a service	10
IaaS	Infrastructure as a service	11
LISP	Locator ID Séparation Protocol	19
IA	Intelligent artificiel	20
AVC	Accident vasculaire cérébral	20
WLAN	Wireless Local Area Network	24
IP	Internet Protocol	24
SLA	Service Level Argument	33
QoS	Qualité de service	33
QR	Quick ResVopnse	34
VM	Virtuelle Machine	35
RACC	Ressources Allocation Dans Cloud Computing	35
RSMA	Ressources Allocation Multi-agent	35
RAFC	Rssources Allocation Dans Fog Computing	36
FCFS	First Come First served	36
SJF	Shortest Job First	36

Table des matières

Introduction Général	8
Chapitre 1 Un Aperçu sur Le Cloud et Le Fog Computing	10
1.1 Introduction	10
1.2 Aperçu sur le Cloud computing	11
1.2.1 Définition	11
1.2.2 Historique.....	12
1.2.3 Les types de service de Cloud computing.....	13
1.2.4 Les modèles de déploiement du Cloud Computing	15
1.2.5 Rôle du Cloud computing	18
1.2.6 Les avantages	18
1.2.7 Les Inconvénients	19
1.3 Aperçu sur le Fog computing.....	20
1.3.1 Définition	20
1.3.2 Architecture.....	20
1.3.3 Les caractéristiques.....	22
1.3.4 Applications du Fog Computing.....	23
1.3.5 Les Avantages.....	25
1.3.6 Les inconvénients.....	26
1.4 Comparaison entre le Cloud computing et le Fog computing.....	26
1.4.1 Capacité du processus de données	26
1.4.2 Latence	27
1.4.3 Réactivité	27
1.4.4 Sécurité	27
1.4.5 Emplacement du centre de données.....	27
1.4.6 Mode de communication.....	27
1.4.7 Dépendance au réseau central.....	28
1.4.8 Nœuds de serveur.....	28

1.5	Internet des objets.....	29
1.5.1	Architecture de l’IoT.....	29
1.6	Conclusion.....	31
Chapitre 2 Allocation des ressources dans le Fog Computing : « Approches et Méthodes »		32
2.1	Introduction	32
2.2	Gestion des ressources en Fog Computing	32
2.2.1	L'estimation des ressources.....	33
2.2.2	Découverte des ressources	34
2.2.3	Surveillance des ressources.....	34
2.2.4	Organisation des ressources.....	34
2.2.5	Allocation des ressources.....	34
2.3	Allocation des ressources dans le Cloud computing.....	35
2.4	Allocation des ressources dans le Fog computing	37
2.5	Conclusion.....	40
Chapitre 3 Stratégie proposé		41
3.1	Introduction	41
3.2	Objectif.....	42
3.3	Architecture.....	43
3.3.1	Couche 1	43
3.3.2	Couche 2	43
3.3.3	Couche 3	43
3.4	Modèle d’application.....	44
3.4.1	Client.....	44
3.4.2	Concentrateur	45
3.5	Processus d’allocation des ressources	46
3.5.1	Conditions d’allocation des ressources.....	47
3.5.2	Modèle d’allocation des ressources	48

3.5.3	Algorithme d'allocation des ressources	50
	Conclusion	58
Chapitre 4	59
4.1	Introduction	59
4.2	Environnement de Développement	59
4.2.1	Langage de programmation Java	60
4.2.2	Environnement de développement.....	60
4.2.3	Simulateur iFogSim	61
4.3	Évaluation des Performances	62
4.3.1	Configuration de la simulation.....	63
4.3.2	Analyse des résultats.....	64
4.4	Conclusion.....	73
Conclusion Générale	74
Bibliographie	76

Introduction Général

L'évolution rapide des technologies de l'information a donné naissance à de nouveaux paradigmes informatiques, parmi lesquels le Cloud Computing et le Fog Computing se distinguent comme des approches innovantes pour le traitement des données et la fourniture de services. Le Cloud Computing a révolutionné la manière dont les entreprises et les utilisateurs accèdent aux ressources informatiques, en offrant une infrastructure centralisée et évolutive pour le stockage, le traitement et la gestion des données à distance. Cependant, malgré ses avantages en termes de scalabilité et d'accessibilité, le Cloud Computing présente des limites en termes de latence, en particulier pour les applications en temps réel ou sensibles à la latence. C'est là qu'intervient le Fog Computing, une extension du Cloud Computing qui vise à réduire la latence et à améliorer l'efficacité du traitement des données en rapprochant les ressources de calcul et de stockage des utilisateurs finaux, au plus près du réseau.

Contrairement au Cloud Computing, où les données sont traitées dans des centres de données distants, le Fog Computing décentralise le traitement en utilisant des nœuds de calcul situés aux frontières du réseau, plus proches des périphériques et des utilisateurs finaux. Cette architecture distribuée permet de réduire les temps de réponse et d'améliorer les performances pour les applications nécessitant une faible latence et une grande réactivité. Dans cette situation, l'utilisation du Fog Computing est essentielle pour les appareils de l'Internet des Objets qui font référence à un réseau de dispositifs physiques connectés à des capteurs, des logiciels et d'autres technologies afin de communiquer et de partager des informations avec d'autres appareils et systèmes en ligne. Les équipements de ces objets produisent un grand volume de données qui nécessitent un traitement rapide et efficace pour des applications comme les villes intelligentes, les véhicules connectés, les soins de santé en temps réel et l'automatisation industrielle. En intégrant des nœuds de traitement situés à proximité des dispositifs IoT, le Fog Computing assure une gestion plus efficace des données, réduisant ainsi la latence et optimisant les performances de ces applications critiques. Dans cette étude,

nous allons proposer une approche adaptative et innovante pour l'allocation des ressources dans le Fog Computing. L'objectif de cette approche est de maximiser le nombre des utilisateurs en minimisant la consommation d'énergie et améliorer les performances des environnements de Fog Computing en optimisant l'utilisation des ressources, garantissant ainsi une qualité de service améliorée et une réactivité accrue avec les utilisateurs Iot.

Pour réaliser cet objectif notre rapport est structuré autour de quatre chapitres avec une introduction et une conclusion générale. Dans le premier chapitre, nous allons présenter de manière générale Un aperçu du Cloud et du Fog Computing, en abordant les utilisateurs IOT. Dans le deuxième chapitre, nous examinerons les travaux existants dans la littérature qui utilisent le paradigme du Fog Computing. Le troisième chapitre sera consacré à la présentation approfondie de notre stratégie et le dernier chapitre présente les résultats de l'évaluation des performances de notre contribution par simulation.

Chapitre 1

Un Aperçu sur Le Cloud et Le Fog Computing

1.1 Introduction

Les systèmes d'information sont constamment à la recherche de nouvelles technologies afin d'améliorer la performance, l'efficacité économique et le respect de l'environnement de leurs infrastructures informatiques. La recherche continue est due à la nature dynamique et évolutive de la scène technologique, ainsi qu'aux exigences croissantes de durabilité et d'efficacité opérationnelle. L'apparition de nouvelles technologies telles que le Cloud computing a changé la façon dont les systèmes informatiques sont gérés. De plus le Fog Computing vise à réduire la latence, à améliorer l'efficacité énergétique et à fournir une puissance de calcul plus proche des utilisateurs finaux en distribuant les ressources de manière décentralisée.

Dans ce chapitre nous aborderons le Cloud Computing en expliquant ses principes essentiels et son mode de fonctionnement, ensuite nous examinerons le Fog Computing en mettant en évidence ses différences clés par rapport au Cloud Computing Enfin, nous explorerons l'Internet des Objets (IoT), en décrivant son architecture et ses composants essentiels, afin de comprendre son rôle dans le contexte du Fog Computing.

1.2 Aperçu sur le Cloud computing

1.2.1 Définition

Le Cloud computing, également connu sous le nom d'informatique en nuage, est une infrastructure qui permet aux serveurs distants de gérer le calcul et le stockage auxquels les utilisateurs se connectent via une liaison Internet sécurisée [1].

Il s'agit de proposer des services informatiques sous forme de services à la demande, accessible via un PC et une connexion Internet de n'importe où, n'importe quand et par n'importe qui [1]

En ce qui concerne le grand public, le Cloud computing est représenté par des services de stockage et de partage de données numériques tels que Box, Drop box, Microsoft One Drive ou Apple i Cloud. Ces services permettent aux utilisateurs de stocker des contenus personnels tels que des photos, des vidéos, de la musique et des documents et d'y accéder depuis n'importe quel terminal connecté dans le monde entier [1].

Un Cloud est un ensemble de matériels, de raccordements réseau et de logiciels [1] ou sa souplesse permet aux fournisseurs d'adapter automatiquement les capacités de stockage et de calcul aux besoins des utilisateurs [1].

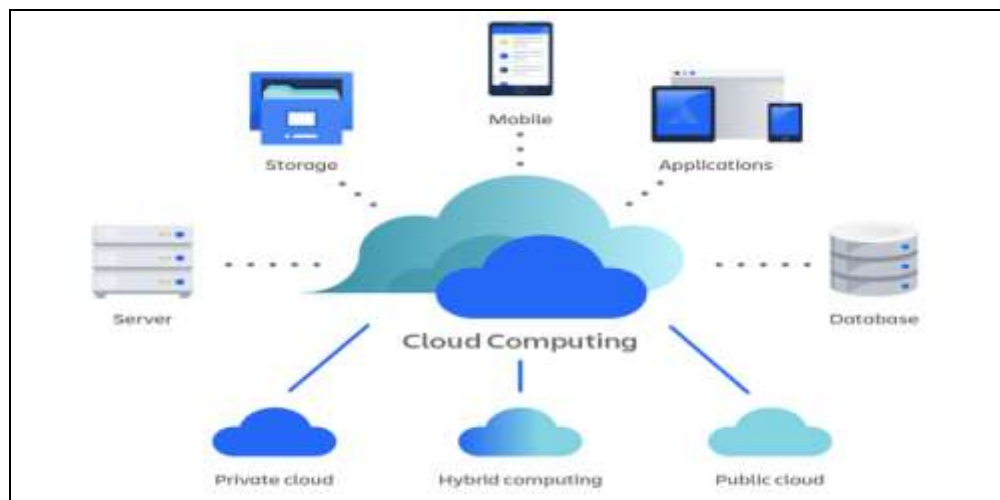


Figure 1 – Cloud Computing.

1.2.2 Historique

Le concept de Cloud computing a été introduit dans les années 2000. Cependant, le concept de "l'informatique comme service" remonte aux années 1960. À cette époque, des entreprises connues sous le nom de "bureau de téléphone" permettaient à d'autres organisations de louer du temps sur un "mainframe" (ordinateur central) plutôt qu'en acheter un.

Ces services ont peu à peu disparu avec l'essor des ordinateurs personnels, car les prix ont largement baissé. Suite à cela, la création de centres de données sur site a permis de stocker de nombreuses données en interne. Avec l'émergence du Cloud computing, la location de ressources informatiques a fait son grand retour. Cela a réellement décollé avec l'apparition de fournisseurs hyper cale comme Amazon Web Services et du modèle SaaS [2].

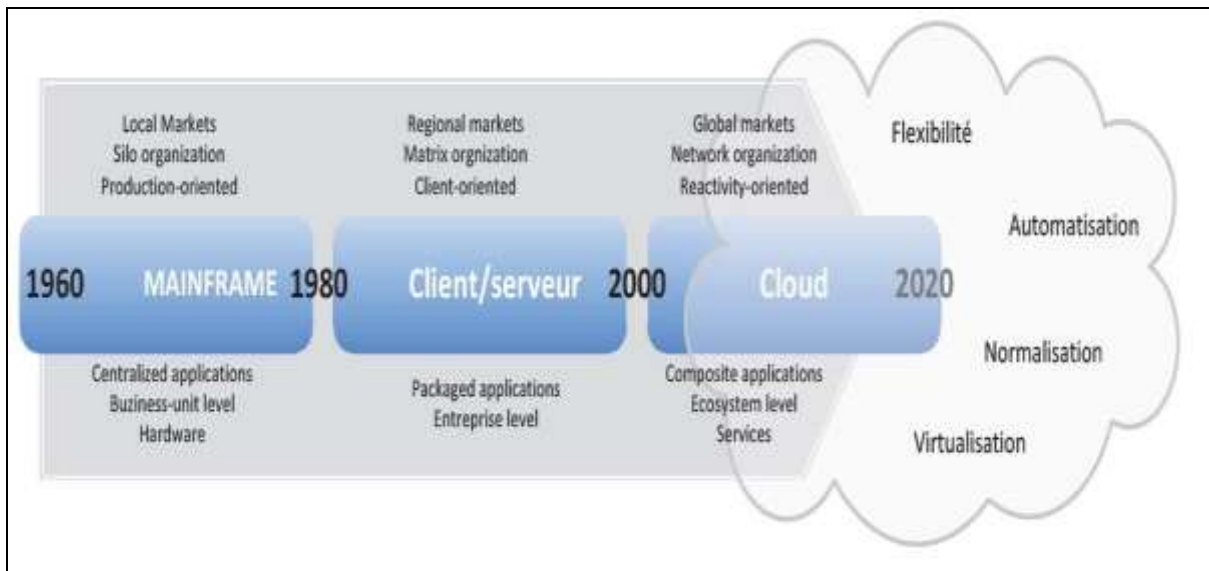


Figure 2– Historique de Cloud Computing.

1.2.3 Les types de service de Cloud computing

1.2.3.1 SaaS (Logiciel-service/Software as a Service)

Le logiciel en tant que service (SaaS) est un moyen de fournir des applications à distance via Internet plutôt que localement sur des machines (appelé logiciel « sur site »). Les applications SaaS sont également connues sous les noms suivants. Logiciel basé sur le Web, Logiciel à la demande et un Logiciel hébergé. Ce logiciel peut être converti d'un produit acheté (un produit que vous achetez une fois et que vous téléchargez sur votre appareil) en un service rendu : c'est à dire un ensemble de fonctionnalités auxquelles vous vous abonnez. Avec le SaaS (Software as a Service), les applications résident sur les serveurs des fournisseurs de logiciels, offrant ainsi plusieurs avantages aux utilisateurs. Ces avantages incluent la libération des contraintes liées aux licences et au matériel, ainsi que des responsabilités de maintenance [3].



Figure 3– Software as a Service (SaaS).

1.2.3.2 PaaS (Platform as a Service)

En tant que service (PaaS), la plateforme offre un environnement complet pour le développement et le déploiement dans le Cloud, avec des ressources qui vous offrent la possibilité de tout fournir, que ce soit des applications basiques basées sur le Cloud ou des applications sophistiquées basées sur le Cloud. Vous faites l'acquisition des ressources

nécessaires auprès d'un prestataire de services Cloud en effectuant un paiement à l'utilisation, puis vous y accédez grâce à une connexion Internet sécurisée [4].



Figure 4– Platform as a Service (PaaS).

1.2.3.3 IaaS (Infrastructure -service/Infrastructure as a Service)

L'IaaS (Infrastructure as a Service) est une approche de service Cloud qui permet aux entreprises et aux particuliers d'accéder à des ressources d'infrastructure à la demande, telles que le calcul, le stockage, la mise en réseau et la virtualisation, via le Cloud. Contrairement à l'approvisionnement traditionnel en ressources informatiques, qui peut prendre plusieurs mois et exiger des investissements importants en matériel et en expertise, l'IaaS offre une solution rapide et flexible. Les entreprises n'ont plus besoin d'acquérir et de gérer leur propre infrastructure physique, ce qui leur permet de se concentrer sur leurs activités principales [5].

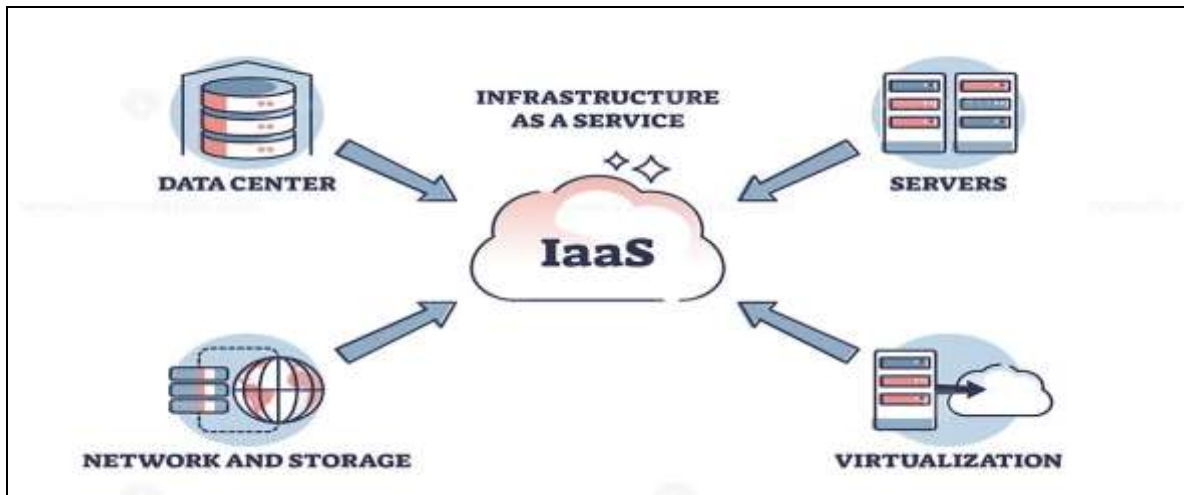


Figure 5– Infrastructure as a Service (IaaS).

1.2.4 Les modèles de déploiement du Cloud Computing

Les modèles de déploiement du Cloud computing sont des stratégies utilisées pour héberger et gérer des infrastructures Cloud.

1.2.4.1 Cloud public

Le Cloud public est la fourniture de services informatiques sur l'internet public par un fournisseur tiers. Ses services sont à la disposition de toute personne qui souhaite les utiliser et leur utilisation peut être gratuite ou payante.

Le modèle de déploiement du Cloud public réduit les coûts initiaux élevés d'achat, de gestion et de maintenance des infrastructures matérielles et logicielles sur site. De plus, grâce à la taille même des fournisseurs de technologies de Cloud public tels que Google, Microsoft et Amazon, il offre un déploiement rapide et une extensibilité presque infinie [6].

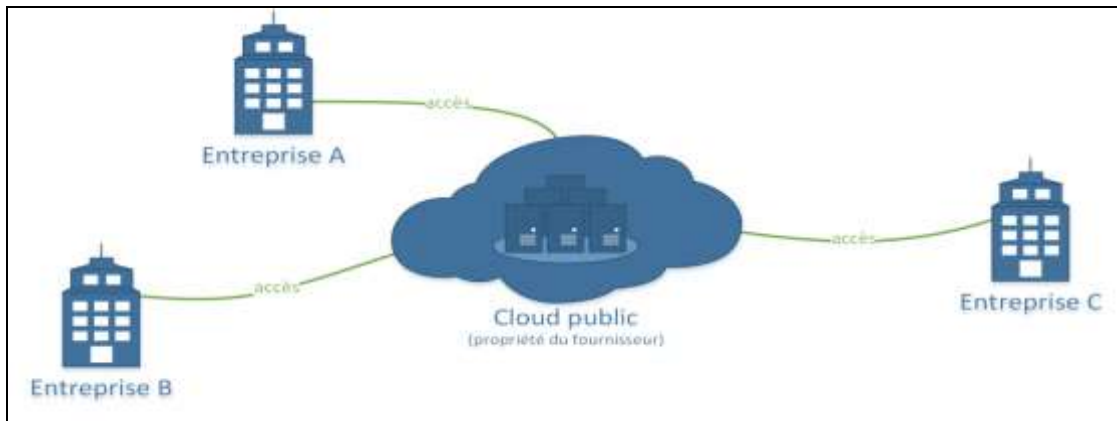


Figure 6– Modèle de déploiement d'un Cloud public.

1.2.4.2 Cloud privé

Un Cloud privé est un modèle de Cloud Computing dans lequel l'infrastructure est dédiée à une seule organisation. Il peut être hébergé dans le Data Center d'une entreprise, dans une installation de Co localisation tierce ou via un fournisseur de Cloud privé qui offre des services d'hébergement de Cloud privé et peut ou non offrir des services mutualisés de Clouds publics traditionnels [7].

Un déploiement de Cloud privé vous offrira bon nombre des mêmes avantages que le Cloud public, notamment le libre-service et l'évolutivité. Même si l'utilisation d'un Cloud public nécessite un niveau d'expertise informatique beaucoup plus élevé, les solutions de Cloud privé offrent un contrôle accru et une meilleure sécurité des serveurs de Cloud privé [7].

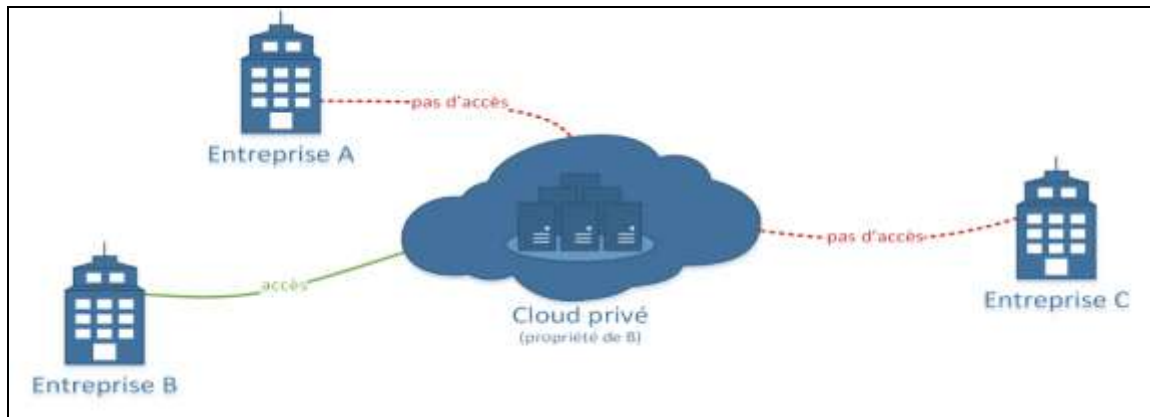


Figure 7– Modèle de déploiement d'un Cloud privé.

1.2.4.3 Cloud hybride

Dans un environnement informatique hybride, les applications sont déployées en utilisant à la fois des ressources de calcul de stockage et de services provenant de différents environnements, tels que des Clouds publics et privés ainsi que des centres de données [8]. Cette approche permet aux entreprises de migrer et de gérer leurs charges de travail entre ces divers environnements Cloud, offrant ainsi une flexibilité accrue pour répondre aux besoins spécifiques de l'entreprise. De nombreuses organisations optent pour des solutions de Cloud hybride afin de réduire les coûts, de minimiser les risques et de développer leurs capacités existantes pour soutenir leurs initiatives de transformation numérique [8].

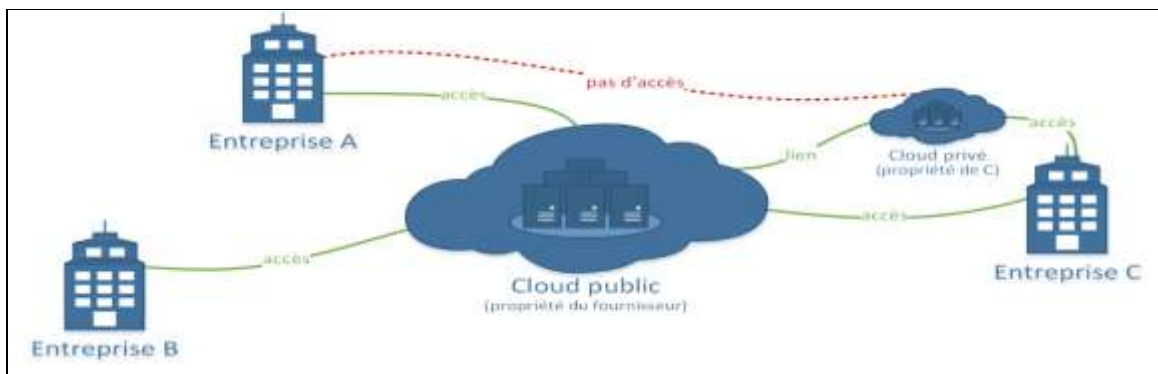


Figure 8– Modèle de déploiement d'un Cloud hybride.

1.2.5 Rôle du Cloud computing

Étant donné que les processus informatiques et le stockage se font sur des serveurs dans un datacenter plutôt que localement sur la machine utilisateur, le Cloud permet aux utilisateurs d'accéder aux mêmes fichiers et applications à partir de presque n'importe quel appareil. C'est ce qui permet à un utilisateur dont le téléphone est défaillant de se connecter à son compte Instagram à partir d'un nouveau téléphone et de récupérer toutes ses photos, vidéos et l'historique de ses conversations. Cela s'applique également aux fournisseurs de messagerie Cloud tels que Gmail ou Microsoft Office 365, ainsi qu'aux fournisseurs de stockage Cloud tels que Dropbox ou Google Drive. [9].

1.2.6 Les avantages

1.2.6.1 Flexibilité et évolution

Le Cloud computing offre plus de flexibilité. Il permet de faire évoluer rapidement les ressources et l'espace de stockage pour répondre aux besoins spécifiques [10].

1.2.6.2 Gain du temps

Permet à vos équipes informatiques de gagner du temps précieux qui peut être consacré à des tâches plus stratégiques [10].

1.2.6.3 Meilleure coopération

Les données sont accessibles aux utilisateurs du monde entier depuis n'importe quel appareil, à condition qu'ils soient connectés à Internet, plutôt qu'à un emplacement ou à un appareil spécifique [10].

1.2.6.4 Sécurité supérieure

Le Cloud computing peut en réalité renforcer votre stratégie de sécurité en raison de la profondeur et de la couverture des fonctionnalités de sécurité, de la maintenance automatique et de la gestion centralisée qu'il intègre [10].

1.2.6.5 Une excellente disponibilité des services

Contrairement aux préjugés, le Cloud computing permet de garantir l'accès et la disponibilité des services, ce qui est crucial aujourd'hui [10].

1.2.6.6 Un usage simple

Augmente l'élasticité et l'agilité en permettant un accès plus rapide aux ressources IT (serveur, stockage ou bande passante) via un portail web simple, sans investir dans des équipements matériels supplémentaires. Ainsi, la disponibilité est immédiate [10].

1.2.7 Les Inconvénients

1.2.7.1 Les pannes

Il existe toujours un risque d'arrêts et de pannes. Vous pouvez perdre du temps et des clients en raison d'interruptions fréquentes de processus importants. Ces risques d'interruption devraient être réduits en choisissant un bon fournisseur de services de Cloud computing et un bon partenaire d'hébergement, en conséquence de la conception de services avec une haute disponibilité et en tenant compte de la reprise après sinistre [11].

1.2.7.2 La sécurité

Le Cloud computing signifie que la sécurité et la confidentialité des données sont confiées aux fournisseurs de serveurs de Cloud. Ainsi, lorsque vous choisissez un fournisseur de Cloud computing, faites un choix judicieux et n'oubliez pas d'examiner attentivement les contrôles, mesures et outils de sécurité du fournisseur. [11].

1.2.7.3 L'enfermement propriétaire

La migration des services de Cloud computing d'un fournisseur à un autre peut être difficile dans certains cas. La migration d'une plateforme à une autre peut être compliquée en raison des différences entre les plateformes des fournisseurs, ce qui entraîne souvent des coûts supplémentaires et des difficultés de configuration. Les erreurs commises lors de la

migration peuvent également exposer vos données à des problèmes supplémentaires de sécurité et de confidentialité. [11].

1.2.7.4 Les dépenses cachées

Le Cloud computing permet aux petites structures de bénéficier de services de qualité sans avoir à investir des sommes importantes dans la mise en place d'une infrastructure informatique. Cependant, cela peut également augmenter les dépenses car cela implique plusieurs coûts cachés qui n'apparaissent qu'à un stade ultérieur. Les coûts de transfert de données, d'utilisation des ressources, de mise à jour logicielle ou de gestion des sauvegardes en sont des exemples. [11].

1.3 Aperçu sur le Fog computing

1.3.1 Définition

Le Fog computing est un réseau hautement virtualisé et décentralisé, constitué de nœuds qui fournissent des services de traitement et de stockage aux dispositifs finaux. Situés à la périphérie du réseau, près des appareils IoT, ces nœuds offrent un support informatique aux applications IoT avec une latence minimale. Bien que fréquemment présents sur le réseau de bord, les nœuds de fog peuvent se déployer de manière hiérarchique n'importe où entre la couche IoT et le Cloud distant. Cette architecture permet de répondre à de nombreuses demandes IoT via le fog, réduisant ainsi le volume de données acheminées vers les serveurs Cloud. Le Cloud reste indispensable pour les applications IoT nécessitant des calculs intensifs et un stockage à long terme. L'objectif du Fog est de soutenir les applications IoT à faible capacité et sensibles à la latence. [12].

1.3.2 Architecture

Pour le déploiement des services IoT au plus près des objets IoT, l'architecture Fog Computing est considérée comme l'un des schémas qui exploitent la capacité de calcul et de stockage de l'infrastructure réseau en plus de celle du Cloud.

L'architecture informatique Fog comprend des composants physiques et logiques sous forme de matériel et de logiciels. Il est composé de dispositifs IoT, de nœuds de brouillard, de nœuds d'agrégation de Fog à l'aide de services de données de Fog, de stockage Cloud à distance et de serveur/Cloud de stockage de données local, comme le montre la figure ci-dessous [13].

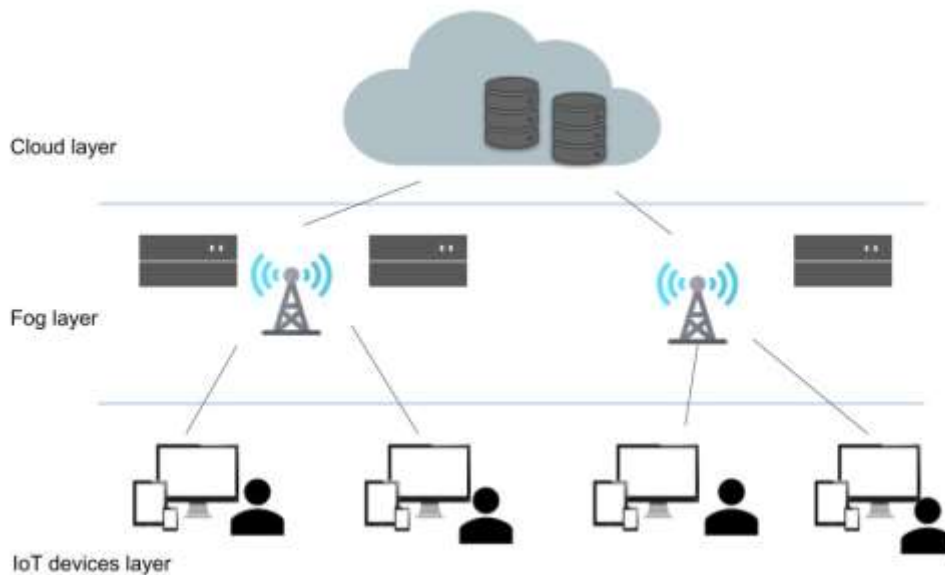


Figure 9–Architecture de Fog computing. [27]

1.3.2.1 Les appareils de l'Internet des objets (IoT)

L'Internet des objets (ou IoT pour Internet of Things) désigne le processus de connexion d'objets physiques à Internet et le réseau qui relie ces objets [13].

1.3.2.2 Les nœuds Fog

Sont des appareils connectés à l'informatique, au stockage et au réseau. Les nœuds de brouillard sont installés à différents endroits pour diverses applications, comme sur le sol d'une usine, au sommet d'un poteau électrique, le long d'une voie ferrée, dans des véhicules, sur une plate-forme pétrolière, etc. [13].

1.3.2.3 Nœuds d'agrégation de brouillard

Chaque nœud de Fog possède un nœud de brouillard agrégé. Il analyse les informations en quelques secondes ou minutes. Ces nœuds peuvent stocker des données IoT pendant des heures ou des jours et ils sont utilisés pour traiter des données moyennes sensibles [13].

1.3.2.4 Cloud distant

Un terme généralement utilisé pour désigner un service de stockage en ligne ou un service informatique déployé sur des serveurs distants accessibles via Internet. Les données moins sensibles ou insensibles au temps est traité, analysées et stockées dans le Cloud [13].

1.3.2.5 Serveur local et Cloud

Le Fog computing utilise fréquemment un serveur local et un Cloud privé pour stocker les informations privées. Ce stockage local aide également à protéger les données [13].

1.3.3 Les caractéristiques

1.3.3.1 Localisation et latence faible

Le calcul du Fog prend en charge la connaissance de l'emplacement des nœuds du Fog. En plus, il réduit le temps de latence lors du traitement des données dans les terminaux car il est plus proche des terminaux. [14].

1.3.3.2 Assistance à la mobilité

La possibilité de se connecter directement à des périphériques mobiles et d'activer des méthodes de mobilité comme le protocole LISP (Locator ID Separation Protocol) [14].

1.3.3.3 Localisation géographique

Contrairement au Cloud centralisé, les applications et les services du Fog sont dispersés et peuvent être utilisés n'importe où [14].

1.3.3.4 Évolutivité

Le Fog fonctionne avec des réseaux de capteurs à grande échelle qui surveillent leur environnement et fournissent des ressources de calcul et de stockage dispersées qui fonctionnent dans ces environnements [14].

1.3.3.5 Communication en temps réel

Au lieu du traitement par lots utilisé dans le Cloud, les applications de Fog computing permettent aux nœuds de Fog d'interagir entre eux en temps réel [14].

1.3.3.6 Diversité

Il existe une variété de fabricants qui ont conçu les nœuds du Fog computing. Ainsi, ils se présentent sous diverses formes et doivent être utilisés selon leurs plateformes. Le Fog peut fonctionner sur diverses plateformes [14].

1.3.3.7 Compatibilité interopérable

Les éléments de le Fog peuvent interagir et fonctionner avec divers domaines et fournisseurs de services [14].

1.3.4 Applications du Fog Computing

Le Fog computing est un modèle qui élargit les capacités du Cloud computing jusqu'au bord du réseau, plus près de l'endroit où les données sont produites et consommées. Grâce à cette méthode, il est possible de concevoir de nouvelles applications et des situations d'usage qui tirent parti de la proximité des ressources informatiques avec les utilisateurs et les appareils.

1.3.4.1 Soins de santé

Le Fog computing peut être bénéfique pour les soins de santé, où le traitement et la réponse en temps réel sont essentiels. Le Fog computing dans le domaine de la santé prend également en charge les applications d'IA de pointe telles que l'analyse d'images en temps réel, la détection précoce des maladies et le traitement personnalisé. Il facilite la

télémédecine, les soins ambulatoires et la surveillance à distance des patients. Une méthode suggérée pour identifier, prévoir et empêcher les victimes d'un AVC de tomber est le Fog computing. Il s'agit d'un système de santé intelligent basé sur le brouillard informatique qui offre une faible latence, une assistance à la mobilité et une sensibilisation à la localisation [15].

1.3.4.2 Réseaux de véhicules

Le Fog computing peut être intégré aux réseaux de véhicules. Le calcul du Fog computing des véhicules est divisé en deux catégories : basé sur l'infrastructure et autonome, en fonction de la nécessité ou non d'infrastructure supplémentaire. Le premier, tout comme Tube, utilise des nœuds de Fog situés le long de la route. Ces nœuds sont chargés d'envoyer et de récupérer des informations des voitures qui passent. Ce dernier utilise des véhicules qui se déplacent à la volée pour produire du Fog et/ou des nuages afin de prendre en charge des événements ponctuels, chaque brouillard peut se connecter à son client à l'intérieur et à d'autres Fog [15].

1.3.4.3 Jeu, systèmes cognitifs et réalité augmentée

Le Fog computing est très utilisé par les applications utilisant la réalité augmentée et sensibles à la latence. Par exemple, EEG Tractor Beam, un jeu d'interaction cerveau-ordinateur multi-joueurs en ligne amélioré, classe continuellement les états cérébraux sur les appareils anti brouillard, puis affine les modèles de classificateur sur les serveurs Cloud en fonction des données recueillies via des capteurs d'électroencéphalogramme [15].

1.3.4.4 Services utilitaires smart

Les services publics intelligents, qui visent à améliorer la production, la livraison et la facturation de l'énergie, peuvent utiliser le Fog computing. Dans ces situations, les appareils de pointe peuvent fournir des informations plus précises sur la consommation d'énergie aux appareils mobiles des clients que les services publics intelligents conventionnels (par exemple, des relevés horaires et quotidiens plutôt que des relevés mensuels). Ces technologies modernes peuvent également estimer les coûts énergétiques quotidiens,

recommander la source d'énergie la plus rentable à tout moment et indiquer quand utiliser les appareils électroménagers pour réduire la consommation des services publics [15].

1.3.5 Les Avantages

1.3.5.1 Moins d'activité sur le réseau

Le trafic entre les dispositifs IoT et le Cloud est réduit par le Fog computing[16].

1.3.5.2 Latence réduite

Le Fog computing simplifie la communication et accélère les processus automatisés d'analyse et de prise de décision [16].

1.3.5.3 Disponible en ligne et hors ligne

Les données doivent être stockées principalement dans le réseau local et transférées au Cloud uniquement lorsque cela est nécessaire dans le Fog computing. Cela le rend idéal pour envoyer des données lorsqu'il n'y a pas de connexion Internet [16].

1.3.5.4 Quantité minimale de données envoyée au Cloud

En traitant les données localement, à proximité de leur source, il réduit la bande passante nécessaire et les coûts associés à la transmission de grandes quantités de données. [16]

1.3.5.5 Traitement immédiat des données

En effectuant des analyses et des opérations en temps réel au niveau des appareils ou des serveurs locaux, le Fog computing permet des réponses rapides et améliore l'efficacité des systèmes, crucial pour des applications nécessitant une faible latence comme l'Internet des objets (IoT) et les systèmes critiques. [16]

1.3.5.6 Sécurité des données

Le réseau local prétraite les données de l'appareil. Cela permet une mise en œuvre dans laquelle les données sensibles restent ou peuvent être chiffrées ou anonymisées avant d'être téléchargées sur le Cloud [16].

1.3.6 Les inconvénients

1.3.6.1 Plus de frais de matériel

Pour permettre le traitement de données local et la communication entre appareils, les terminaux et capteurs IoT de l'architecture Fog computing doivent être équipés d'une unité de traitement supplémentaire [16].

1.3.6.2 Les besoins de maintenance augmentent

Le traitement des données décentralisé nécessite plus de maintenance. La raison en est que les composants de traitement et de stockage sont dispersés sur l'ensemble du réseau et qu'ils ne peuvent ni n'être maintenus ni administrés de manière centralisée, contrairement aux solutions Cloud [16] ;

1.3.6.3 Des problèmes supplémentaires liés à la sécurité du réseau

Les attaques de type « homme du milieu » (ou intercepteur) affectent le Fog computing [16].

1.4 Comparaison entre le Cloud computing et le Fog computing

Le Fog computing et le Cloud computing sont deux méthodes complémentaires pour traiter et stocker les données, mais ils présentent des différences en ce qui concerne leur localisation, leur évolutivité, leur latence et leurs applications.

1.4.1 Capacité du processus de données

Le Fog Computing possède une capacité de traitement des données relativement réduite. Les applications qui nécessitent un maximum de bande passante devraient l'utiliser.

En revanche, le Cloud computing offre de nombreux avantages en termes de capacités de travail. Ainsi, il est adapté à l'analyse du Big Data et à la modélisation sophistiquée [17].

1.4.2 Latence

La latence désigne le temps requis pour que les données circulent d'un appareil à un serveur/appareil. Au milieu de l'obscurité informatique, la capacité est limitée puisque les données ne doivent pas être très éloignées de l'appareil [17].

1.4.3 Réactivité

En grande partie, le Fog repose sur le matériel local. Le temps de réponse diffère en raison des contraintes de bande passante et de la latence [17].

1.4.4 Sécurité

La sécurité du Fog computing est très élevée car il utilise des réseaux localisés ou distribués. Le Cloud computing garantit également une sécurité élevée grâce au cryptage des données et à d'autres techniques. En même temps, il est plus vulnérable aux attaques cybernétiques [17].

1.4.5 Emplacement du centre de données

La dispersion géographique du brouillard informatique peut varier, mais généralement, il est plus localisé et ne peut fonctionner qu'à partir d'un emplacement géographique.

Contrairement à cela, le Cloud computing est localisé car il utilise un réseau de serveurs Cloud situés dans différentes régions géographiques [17].

1.4.6 Mode de communication

Différentes communications sans fil (WLAN, Wifi, 3G, 4G) ou filaires sont requises pour le Fog Computing. Toutefois, le Cloud computing utilise un réseau IP afin de garantir sa fonctionnalité [17].

1.4.7 Dépendance au réseau central

En utilisant le Fog computing, on observe une approche décentralisée qui exploite les zones périphériques du réseau pour le stockage et le traitement des données. Ces éléments comprennent des dispositifs ou des capteurs individuels [17].

1.4.8 Nœuds de serveur

En raison de son caractère, le Fog computing nécessite l'utilisation d'un grand nombre de nœuds de serveur afin de traiter les données. Cependant, le Cloud computing nécessite moins de serveurs [17].

Le tableau ci-dessus représente les différences citées :

Tableau 1 – La comparaison entre le Cloud et le Fog computing

Fonctionnalité	Fog	Cloud
Latence	Latence faible	Haute latence
Réactivité	Dépend de la bande passante	Très réactif
Sécurité	Sécurité améliorée	Sécurité renforcée avec décryptions
Emplacement du centre de données	Utilisable Ment, il fonctionne depuis un seul endroit	Distribué dans divers endroits
Mode de communication	Utilise de réseaux sans fil ou câblés	Utilise le réseau IP
Dépendance au réseau central	N'a pas besoin d'un cœur de réseau solide	Un noyau de réseau solide est essentiel
Nœuds de serveur	Utilise de nombreux nœuds de serveur	Utilise moins de nœuds de serveur

1.5 Internet des objets

L'Internet des objets (IoT) est défini comme un paradigme dans lequel des objets équipés de capteurs, d'actionneurs et de processeurs communiquent entre eux pour servir un objectif significatif. Est une infrastructure de réseau mondiale en constante évolution dans laquelle les objets physiques, tels que les accessoires, les vêtements, les machines, les véhicules, etc., sont autonomes et modifiables. Ces choses devraient être capables, grâce à l'utilisation des technologies de l'information et de la télécommunication, à la visualisation et à la transmission de données sur Internet, à la surveillance de leur environnement et d'agir en se basant sur le partage d'informations. La domotique, la logistique, la télémédecine et l'apprentissage en ligne sont quelques exemples de scénarios d'application possibles dans lesquels l'Internet des objets joue un rôle important dans ce contexte [18].

Les fabricants d'équipements, les fournisseurs de services Internet et les développeurs d'applications bénéficient de grandes opportunités de marché grâce à l'Internet des objets. Selon Cisco, le nombre d'objets connectés dépassera les 50 milliards en 2020, générant un marché mondial d'environ 2.700 à 6.200 milliards de dollars par an. L'IoT offre également de bonnes opportunités d'emploi. En effet, 4,5 millions de développeurs contribueront à des projets IoT dans le monde d'ici 2020. En revanche, en 2020, plus de 220 millions de voitures connectées seront utilisées dans les secteurs de l'automobile et des transports.

Les applications et services de santé basés sur l'IoT, tels que d'ici 2025, la télémédecine et la santé mobile devraient générer une croissance annuelle de 1.100 à 2.500 milliards de dollars [18].

1.5.1 Architecture de l'IoT

Il n'y a pas de consensus universel sur l'architecture de l'IoT. Divers chercheurs ont proposé diverses architectures. Une architecture à trois couches est l'architecture la plus élémentaire. Il est composé de :

La couche perception, la couche réseau et la couche application. La couche physique connue sous le nom de couche perception comprend des capteurs capables de détecter et de collecter des informations sur l'environnement. Il détecte certains paramètres physiques ou identifie d'autres objets intelligents dans l'environnement. La couche réseau connecte les autres objets intelligents, les périphériques réseau et les serveurs. Les données des capteurs sont également transmises et traitées grâce à ses fonctionnalités. La couche application est chargée de fournir aux utilisateurs des services spécifiques à l'application. Il définit une variété d'applications d'Internet des objets, telles que les maisons intelligentes, les villes intelligentes et la santé intelligente.

L'architecture à trois couches au même rôle que les couches de perception et d'application. La couche transportable transfère les données du capteur à la couche de traitement. et vice versa via des réseaux [19].

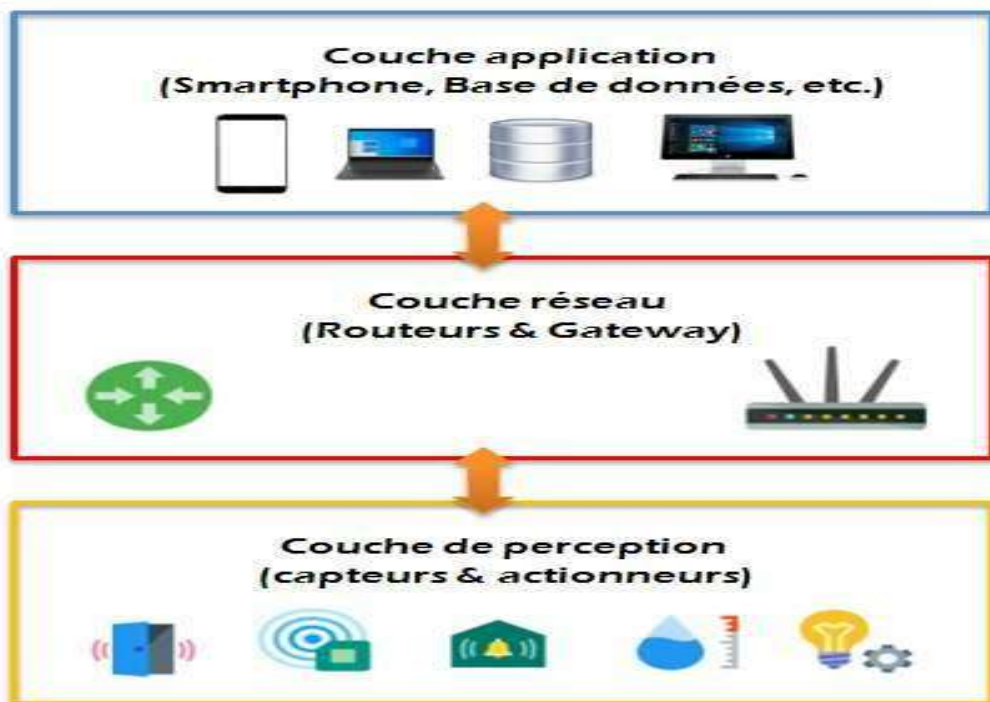


Figure 10– Architecture de l'IoT.

1.6 Conclusion

Dans ce chapitre, nous avons présenté deux paradigmes majeurs, le Cloud et le Fog computing avec leurs définitions, caractéristiques des avantages et des inconvénients, mettant en évidence les similitudes, les différences et les synergies entre ces deux paradigmes informatiques.

Dans le prochain chapitre, nous allons parler de la gestion des ressources en Fog computing ainsi que l'allocation des ressources dans le Cloud et le Fog computing. De plus, nous examinerons les stratégies et les méthodes utilisées pour allouer les ressources dans le Cloud computing.

Chapitre 2

Allocation des ressources dans le Fog Computing :

« Approches et Méthodes »

2.1 Introduction

Le besoin croissant des applications à accéder simultanément à des ressources réparties sur plusieurs sites a conduit la communauté informatique à s'intéresser à l'étude de l'allocation des ressources dans le Cloud computing et le Fog computing. Dans ce chapitre nous allons dans un premier temps définir la gestion des ressources en Fog computing et ses étapes, ensuite, nous allons présenter les méthodes développées dans la littérature qui traite l'allocation des ressources dans le Cloud et le Fog computing.

2.2 Gestion des ressources en Fog Computing

La gestion des ressources en Fog computing est cruciale pour optimiser les performances et l'efficacité des infrastructures de Fog computing. Le Fog computing étant un modèle de traitement des données décentralisé et distribué, la gestion des ressources implique la coordination des ressources informatiques disponibles au niveau local (dans les dispositifs périphériques, les passerelles, etc.) pour répondre aux besoins de traitement et de stockage des applications.

Dans ce qui suit, une brève description du processus de gestion des ressources sera fournie.

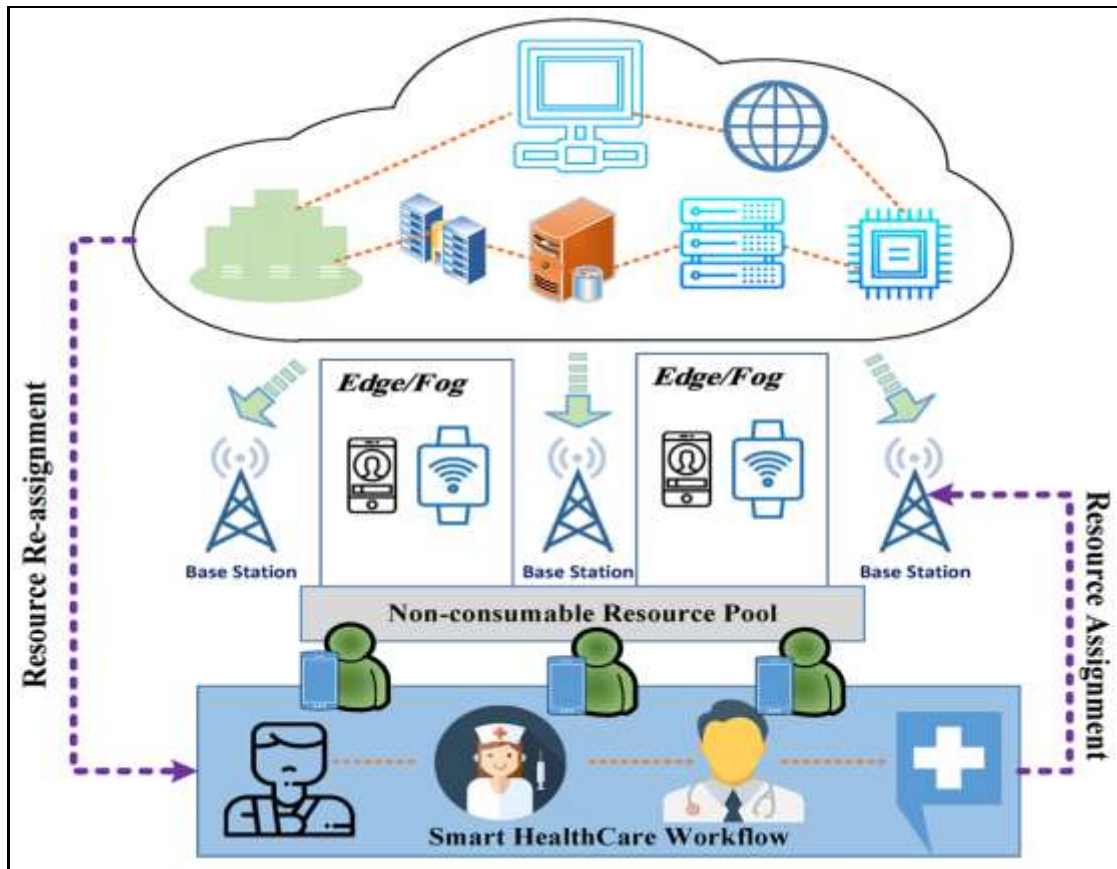


Figure 11- Gestion des ressources en Fog Computing.

2.2.1 L'estimation des ressources

Est un élément crucial du processus de gestion des ressources et fait référence au calcul de la quantité de ressources informatiques et du temps nécessaire pour effectuer des tâches dans des environnements de Fog computing.

L'une des principales exigences de la gestion des ressources informatiques est la capacité d'estimer combien de ressources seront nécessaires pour effectuer une tâche. La nature spécifique du projet, les exigences de traitement, la distribution des ressources informatiques et les contraintes de latence sont quelques-uns des nombreux facteurs qui l'affectent [29].

2.2.2 Découverte des ressources

Le processus de découverte des ressources est considéré comme un défi difficile et fondamental pour l'environnement en raison des caractéristiques du brouillard informatique, telles que la mobilité, la forte répartition géographique et l'hétérogénéité.

2.2.3 Surveillance des ressources

Dans le Fog computing, la surveillance des ressources est un élément essentiel de la gestion globale du système. Elle garantit des performances optimales, prévient les problèmes potentiels et maintient le système disponible.

La surveillance est une fonctionnalité importante et considérée comme essentielle pour orchestrer correctement les services de brouillard et collecter des informations d'état mises à jour sur les nœuds de brouillard et les liaisons de communication.

2.2.4 Organisation des ressources

Selon les articles examinés, l'étape d'orchestration du brouillard est définie comme une fonction de gestion qui gère le cycle de vie du service. Il doit surveiller l'infrastructure sous-jacente, réagir rapidement aux changements et respecter les règles de confidentialité et de sécurité afin de fournir les services demandés à l'utilisateur.

Cela peut être accompli grâce à des technologies telles que les conteneurs, qui facilitent la gestion et le déploiement d'applications sur des environnements distribués.

2.2.5 Allocation des ressources

L'étape d'Allocation joue un rôle essentiel dans le processus de gestion des ressources, englobant le plus grand nombre de termes et discuté par la plupart des auteurs.

Le Fog computing vise à répondre à un ensemble de tâches qui ont des exigences de QoS différentes (comme le coût et le temps d'exécution) dans un ensemble de ressources. Il représente une approche dynamique, adaptative et axée sur les besoins des applications. Pour tirer le meilleur parti du Fog Computing, il est essentiel d'avoir une compréhension

approfondie des exigences spécifiques de chaque application, ainsi que des mécanismes d'orchestration et de gestion des ressources.

L'allocation des ressources, doit fournir la ressource la plus appropriée en fonction des connaissances obtenues lors des étapes précédentes, telles que l'estimation ou le déchargement des ressources. Elle peut être considérée comme l'étape qui reçoit informations d'estimation et effectue la réservation efficace des ressources de calcul afin que la tâche puisse être exécutée dans l'environnement Fog computing, conformément aux exigences de QoS comme pour tout processus, l'allocation des ressources comporte des entrées et des sorties. Les entrées proviennent des précédentes étapes, qui dans cette proposition de gestion des ressources est l'étape d'estimation. Ces entrées indiquent Métriques de qualité de service ainsi que d'autres exigences (par exemple informatiques). Le résultat de l'allocation des ressources est composé de la technique d'allocation, de la méthode de virtualisation et des couches d'architecture à couvert. L'allocation des ressources fait référence à l'étape du processus de gestion des ressources qui vise à sélectionner, réserver et utiliser les meilleures ressources disponibles pour exécuter une charge de travail dans l'environnement de Fog, tout en respectant les paramètres de QoS ». Sur la base de cette définition et en comprenant qu'il est nécessaire de mieux connaître les propositions d'allocation des ressources existant dans la littérature, il a été possible de déterminer les questions de recherche (QR) auxquelles ce travail devrait répondre.

2.3 Allocation des ressources dans le Cloud computing

Il est courant de commencer par discuter de l'allocation des ressources dans le Cloud computing avant d'aborder le Fog computing en raison de la popularité et de la familiarité accrues du Cloud computing. En effet, le Cloud fournit souvent les services de traitement et de stockage distants utilisés par les nœuds de Fog pour compléter leurs propres capacités locales.

Nous nous concentrons principalement sur les Cloud Infrastructure-as-a-Service (IaaS) dans cette section car le modèle de service du Fog computing est IaaS qui représente un

modèle de service qui fournit directement aux agents d'infrastructure informatique. La majorité des fournisseurs de Cloud IaaS proposent désormais des instances de VM préconfigurées de types fixes. Cependant, les plans de réservation ne sont évidemment pas adaptés aux agents aux tâches incertaines, ce qui est courant dans les scénarios IoT. Même si les mécanismes de forfaitisation sont faciles à mettre en œuvre et sont évidemment véridiques, ils ne sont pas très efficaces en matière de protection sociale ou revenus.

Par exemple, une tâche de grande valeur n'obtiendrait pas une priorité élevée dans un mécanisme de prix fixe. Pour améliorer l'efficacité des revenus, la tarification spot est également adoptée par ces fournisseurs de Cloud, l'agent offre un prix pour sa tâche. Ensuite, la tâche sera exécutée lorsque le prix au comptant sera inférieur à ce prix, et l'agent paie le prix au comptant. Cependant, ce mécanisme n'a aucune garantie d'accord de niveau de service (SLA). Plus précisément, le prix au comptant est volatil, et ainsi les tâches peuvent être interrompues fréquemment, et il n'y a aucune garantie de finition au moment de la tâche.

Dans [28], les auteurs ont proposé un modèle de composition de services basé sur un environnement multi-Cloud utilisant des agents multiples. Ces agents correspondent aux centres de données Cloud adaptés aux services requis. Si un seul centre de données Cloud ne peut pas répondre aux exigences, il combine alors les services d'autres centres de données Cloud. Cette approche, pilotée par la mémoire des agents, effectue une recherche par force brute avec moins de calculs en répartissant la charge entre différents agents. De cette manière, l'efficacité et la rapidité de la composition des services sont améliorées.

Dans un autre ouvrage [29], Raju et al ont proposé une méthode d'approvisionnement et d'allocation des ressources en deux étapes pour l'environnement Cloud, en tenant compte des délais. Dans leur modèle, deux types différents de machines virtuelles sont requis de manière séquentielle pour terminer les travaux. L'allocation des machines virtuelles est réalisée en considérant les délais de traitement, tandis que l'allocation des tâches prend en compte les délais d'attente et de réponse.

Dans [30], les auteurs ont présenté un algorithme d'allocation de services dans les réseaux 5G, visant à répondre aux exigences de service en segmentant le réseau en tranches basées sur des délais limites et en ajustant les paramètres en fonction de la priorité du service. Cependant, cet algorithme d'allocation n'a pas pris en considération les besoins dynamiques des utilisateurs dans un environnement à ressources limitées.

Dans une autre approche, Liu et ses collègues [31] ont développé un algorithme basé sur la méthode biologique évolutive, en se concentrant spécifiquement sur l'utilisation de l'algorithme génétique. Ils ont incorporé une fonction de pénalité adaptative afin de tenir compte des délais stricts lors de l'allocation des ressources. L'objectif principal de cet algorithme était de faciliter l'allocation efficace des ressources dans un environnement Cloud. Plus précisément, leur attention s'est portée sur l'optimisation des flux de travail scientifiques [21].

2.4 Allocation des ressources dans le Fog computing

Bien que le Fog computing soit un concept assez nouveau, de nombreuses études ont été menées sur l'allocation des ressources dans ce domaine. En particulier, RAFC a de nombreuses mesures de performance, et diverses études peuvent essayer d'optimiser chacune d'entre elles. Les résultats des mesures couramment utilisées sont énumérés ci-dessous :

- **Temps d'exécution** : Le temps d'exécution est le temps nécessaire pour terminer les tâches, à l'exclusion de tout temps d'attente.
- **Consommation d'énergie** : la consommation d'énergie est la quantité d'énergie utilisée pour traiter toutes les tâches.

Ensuite, nous passerons en revue les études sur l'allocation des ressources dans ce domaine. Cette discipline examine comment allouer et gérer efficacement les ressources, telles que la puissance de calcul et le stockage, dans un environnement de Fog computing distribué. De nombreuses recherches ont été menées pour développer des stratégies d'allocation des ressources dans le Fog computing, avec pour objectif d'optimiser l'utilisation

des ressources disponibles, de minimiser la latence et de garantir une qualité de service élevée

Dans [27], Ni et al. ont proposé une politique d'allocation des ressources pour le calcul en Fog computing basée sur des réseaux de Petri temporisés (PTPN). Cette approche permet aux utilisateurs de choisir de manière autonome les ressources satisfaisantes à partir d'un groupe de ressources pré-allouées. La sélection des ressources prend en compte la mesure du coût pour accomplir les tâches.

Dans une étude [32], Brogi et al ont développé un outil appelé FogTorch, réalisé en Java, qui vise à déployer des applications composites IoT dans un environnement Fog. Cet outil intègre un algorithme spécifique conçu pour prendre en compte les exigences des applications ainsi que les capacités des ressources disponibles afin de faciliter le déploiement. L'algorithme proposé comprend une phase de prétraitement visant à restreindre l'espace de recherche des déploiements possibles, ainsi qu'une phase de rétrogradation et d'heuristiques permettant de sélectionner un déploiement optimal parmi les candidats éligibles.

Dans [33], les auteurs ont rapporté les résultats de l'application de simulations Monte Carlo à l'outil FogTorch, en tenant compte de la latence, de la consommation de bande passante et des ressources telles que le processeur, la mémoire RAM et le stockage.

Dans [34], Skarlat et les autres auteurs ont présenté une description hiérarchique des infrastructures de Fog, composées de colonies constituées d'un nœud de contrôle d'orchestration et de plusieurs cellules. Chaque cellule représente un composant logiciel s'exécutant sur des appareils pour contrôler et surveiller un groupe spécifique d'appareils IoT, tandis que le nœud de contrôle d'orchestration est chargé de gérer les cellules au sein de la même colonie. Pour ce faire, ils ont utilisé un algorithme génétique pour calculer un plan d'allocation des ressources, prenant en compte la consommation de ressources et les contraintes de temps d'exécution afin de déterminer où un service peut être déployé au sein d'une colonie.

Dans une étude distincte, [35] Du et al ont développé un algorithme d'allocation de ressources pour l'environnement Fog computing. Ce travail vise à optimiser la puissance de transmission, la bande passante radio, et les ressources de calcul, tout en garantissant un délai maximum tolérable et l'équité entre les utilisateurs. Leur algorithme d'allocation de ressources, connu sous le nom d'Algorithme de Déchargement et d'Allocation des Ressources (CORA), a été proposé pour répondre à ces exigences.

Zhang et Li [36] ont exposé une approche d'allocation des ressources pour préserver la confidentialité des nœuds Fog contre les attaques malveillantes, même si la clé privée de tous les nœuds Fog est compromise. Dans cette configuration, un ensemble de dispositifs Fog génère une clé publique de groupe, garantissant ainsi la sécurité du système contre les écoutes indiscretes et servant de passerelle intelligente [40].

Une autre étude menée par Yu et al dans [37] expose l'utilisation de la méthode des multiplicateurs à direction alternée (ADMM), les algorithmes Branch-and-Bound, ainsi que la décomposition de Joint Benders et l'algorithme de Dinkelbach pour l'allocation de ressources à grande échelle dans les environnements Fog. Cette approche tient compte du débit pour cette allocation à grande échelle, en prenant en considération la distance entre les nœuds Fog [40].

Ilva et Fonseca [38] ont proposé une approche d'allocation de ressources dans le Fog computing qui prend en considération les ressources du système disponible. Cette méthode, appelée Régression du Processus Gaussien pour l'Allocation des Nuages de Fog (GPRFCA), repose sur une étude des demandes de soumission antérieures et de la latence du réseau pour l'attribution des ressources. En outre, sa vocation est de réduire au maximum la consommation d'énergie des nœuds Fog.

Xu et al. [39] proposent l'emploi de la méthode d'allocation dynamique des ressources (DRAM) afin de rétablir l'équilibre de la charge de travail dans l'environnement Fog. Cette méthode tient compte des exigences en ressources et de l'espace disponible dans les nœuds de

calcul afin de définir les plans de ressources dynamiques, en plus des allocations de ressources statiques [40].

2.5 Conclusion

Au cours de ce chapitre d'état de l'art, nous avons d'abord exploré la gestion des ressources dans le contexte spécifique du Fog computing. Nous avons mis en évidence l'importance cruciale de cette gestion, axée principalement sur la réduction de la latence du système. Par la suite, nous avons abordé en détail l'allocation des ressources dans le Fog et le Cloud computing en mettant en lumière les différentes méthodes et approches utilisées pour gérer efficacement les ressources dans ce contexte complexe et en constante évolution.

Chapitre 3

Stratégie proposé

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. En outre, nous avons examiné certaines études qui ont été présentées dans la littérature et qui partagent des similitudes avec notre projet en ce qui concerne le contexte, les objectifs et l'environnement.

Le présent chapitre permet de détailler notre stratégie d'allocation des ressources dans un environnement Fog computing pour des utilisateurs IoT. Tous d'abord, nous exposerons l'architecture et le modèle d'application que nous avons considéré. Ensuite, nous allons montrer notre approche en présentant les différents diagrammes et un algorithme que nous avons proposés de les intégrer dans le processus d'allocation des ressources au niveau du Fog computing.

3.2 Objectif

L'objectif principal de notre travail est de proposer une approche d'allocation des ressources dans le Fog computing basée sur la maximisation du nombre d'utilisateurs et la minimisation de consommation d'énergie. Pour atteindre ces objectifs de manière équilibrée et efficace, nous avons adopté une stratégie de pondération où chaque demande d'utilisateur est évaluée en fonction de plusieurs critères clés en attribuant un poids à ces derniers selon son importance relative. Nous allons formaliser notre problème comme suit :

- La maximisation du nombre d'utilisateurs : l'objectif est de maximiser l'utilisation des ressources disponibles afin de satisfaire un maximum d'utilisateurs tout en optimisant l'utilisation des ressources du Fog en tenant compte de l'utilité globale du système.
- La minimisation de consommation d'énergie : Les utilisateurs IoT sont des nœuds mobiles donc ils sont susceptibles de consommer beaucoup d'énergie afin d'allouer les ressources et accomplir leurs tâches. Donc la consommation d'énergie par un utilisateur IoT doit être minimisée.
- Minimisation du temps de réponse : le temps est un facteur important dans le processus d'allocation des ressources. Alors, nous formulons le temps de réponse pour l'utilisateur IoT comme suit :

$$TR_{u_i} = TD_{u_i} + TH_{u_i} \quad \forall i \in M \quad (4.1)$$

$M = \{1, 2, \dots, m\}$ tel que m est le nombre des utilisateurs IoT dans le réseau. TD_u est le temps d'attente d'un utilisateur pour allouer les ressources disponibles. TH_u est le temps d'utilisation des ressources demandées par l'utilisateur. Cependant, nous visons à minimiser TD_u .

3.3 Architecture

Dans notre travail, nous avons proposé une plateforme basée sur le Fog Computing. Le Fog présente une infrastructure dense et géo-distribuée à grande échelle [21]. Nous avons subdivisé notre architecture horizontalement sur trois couches (Cloud, Fog , IOT)

3.3.1 Couche 1

La couche Cloud se compose de ressources informatiques puissantes situées dans un centre de données. Ces ressources sont responsables du traitement avancé des données, de l'analyse et du stockage qui permettent de traiter un grand volume de données hétérogènes et de déterminer des prédictions et des actions à réaliser.

3.3.2 Couche 2

La couche Fog computing est une extension du concept de Cloud computing. Elle vise à améliorer l'efficacité et la réactivité des systèmes IoT en utilisant des machines virtuelles (VM) qui jouent un rôle crucial dans le Fog .

3.3.3 Couche 3

La couche IoT se compose des appareils et des capteurs physiques qui collectent des données. Ces appareils peuvent inclure des capteurs de température, des capteurs de pression, des caméras, des microphones, etc. Les données collectées par ces appareils sont envoyées à la couche Fog pour le traitement.

Ci-dessous nous présentons notre architecture schématisée d'une façon simple dans la Figure 13 :

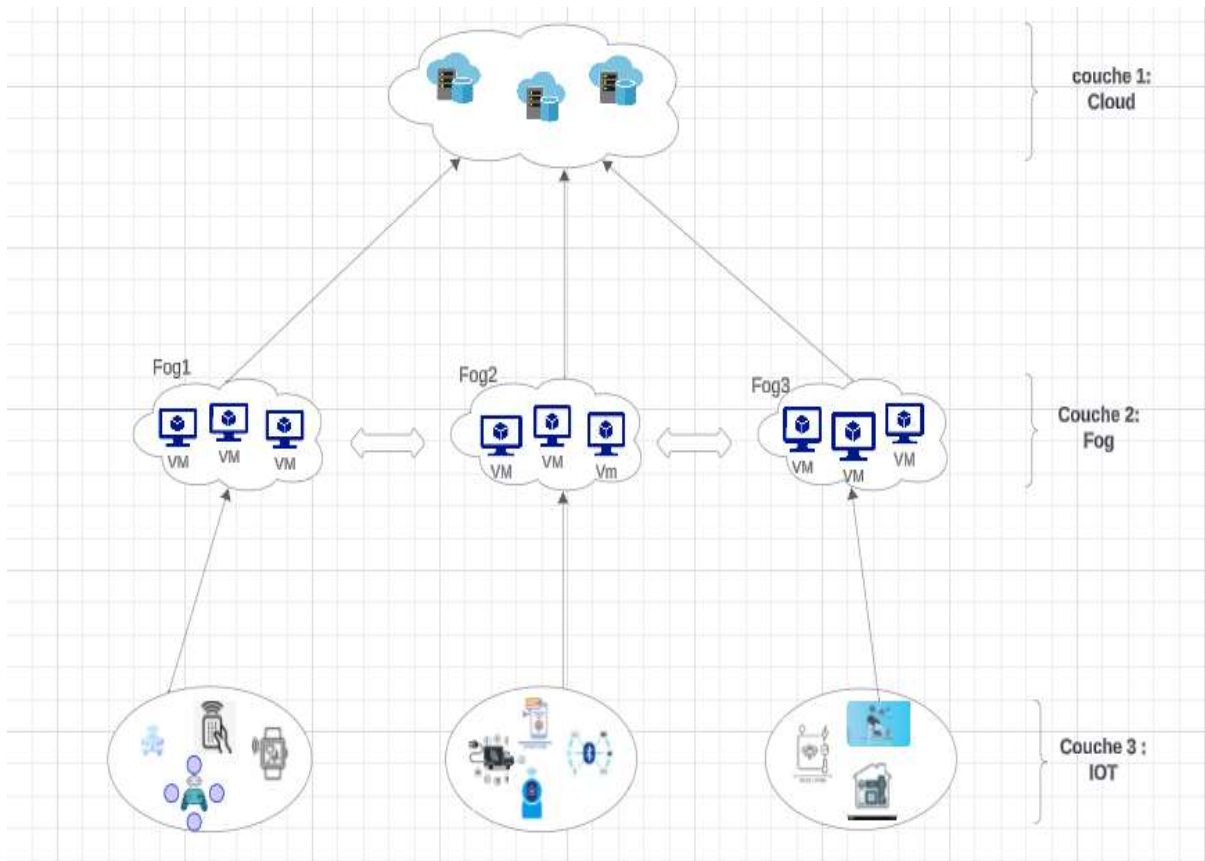


Figure 12 - Schéma de l'Architecture Proposée.

3.4 Modèle d'application

Le traitement de l'application est effectué par deux modules principaux : client et concentrateur, comme il est montré dans la figure 14.

3.4.1 Client

Le module client établit une connexion avec le capteur et reçoit les signaux bruts EEG. Il procède à la vérification des valeurs des signaux reçus afin de détecter toute divergence et refuse toute lecture qui semble étrange. En cas de cohérence de la valeur du signal détecté, il envoie la valeur au Module Concentrateur afin d'obtenir le niveau de concentration de l'utilisateur à l'aide du signal. Lorsqu'il reçoit la concentration niveau, il la montre en envoyant la valeur à l'actionneur DISPLAY.

3.4.2 Concentrateur

Le module concentrateur est responsable du traitement et de l'analyse des données reçues du module client. Il effectue des opérations de concentration et d'analyse sur les données pour extraire des informations pertinentes, telles que le niveau de concentration de l'utilisateur. Une fois les données traitées, le module concentrateur les diffuse aux modules clients connectés pour une utilisation ultérieure. Il joue un rôle crucial dans la prise de décision et dans la fourniture d'informations précieuses aux utilisateurs finaux.

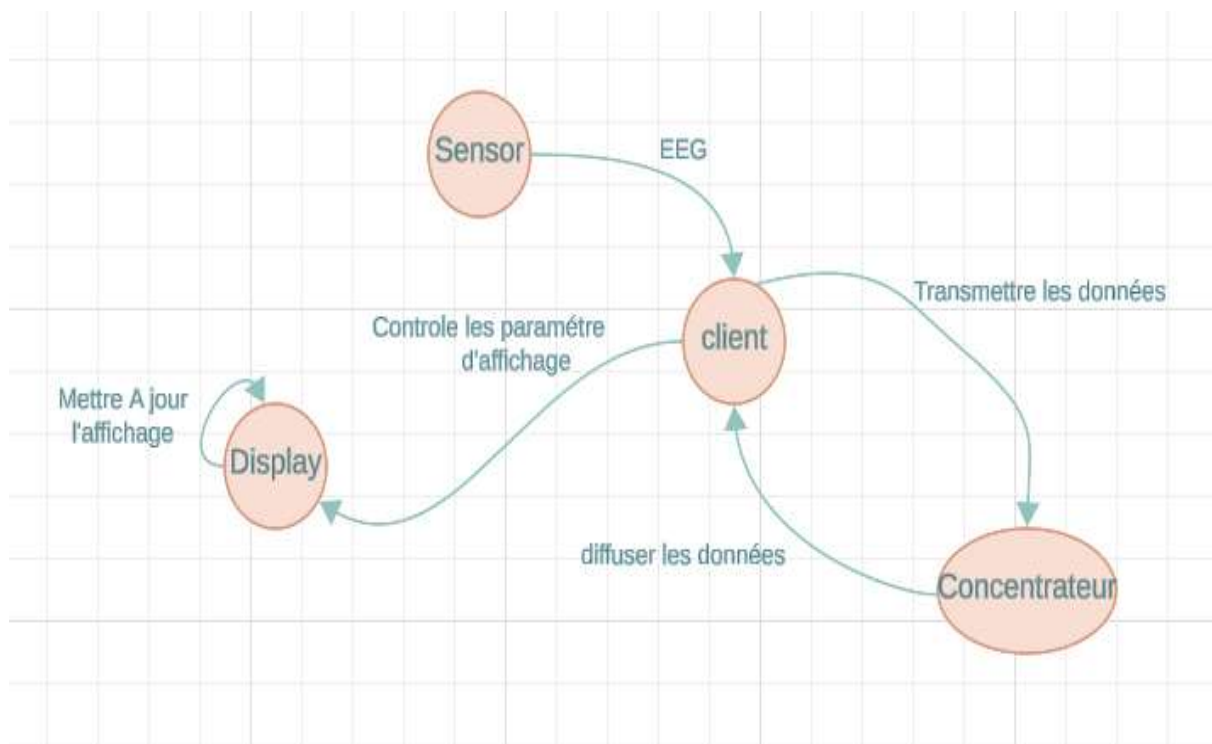


Figure 13- Modèle d'application.

Après avoir défini notre modèle d'application, la figure suivante présente le diagramme de cas d'utilisation qui illustre le modèle d'allocation des ressources dans notre stratégie. Ce diagramme décrit les fonctionnalités principales de l'utilisateur IoT et du nœud Fog qui représentent respectivement le client et le concentrateur.

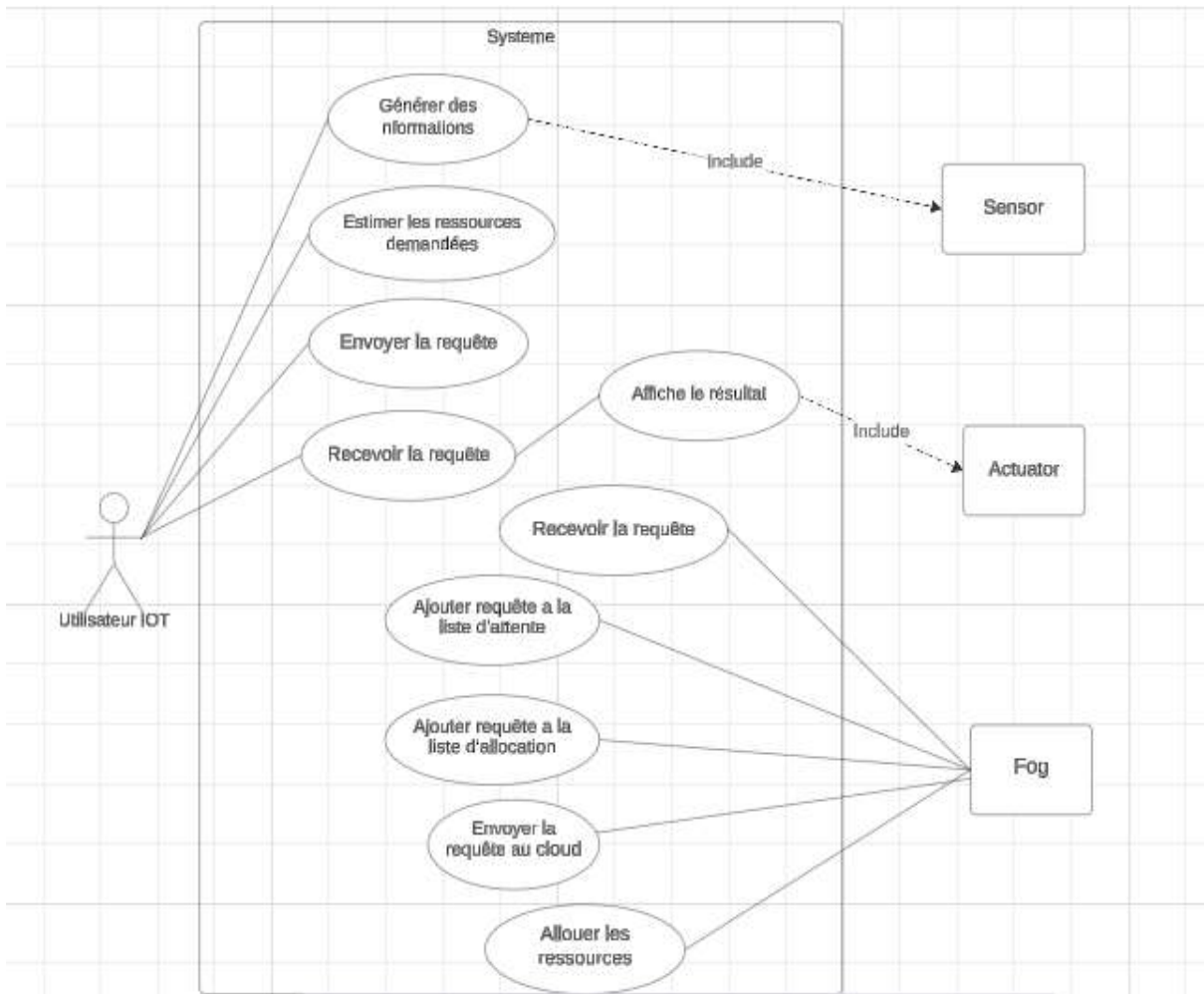


Figure 14– Diagramme de cas d’utilisation

3.5 Processus d’allocation des ressources

Le processus d’allocation des ressources dans le Fog computing comporte plusieurs étapes essentielles. Pour illustrer ce processus d’allocation, nous avons créé un diagramme de séquence (Figure 16) qui représente l’ordre chronologique des messages échangés entre les différents objets d’un système. Il permet de visualiser les interactions entre les différents composants d’un système et de comprendre le comportement de celui-ci dans un contexte spécifique.

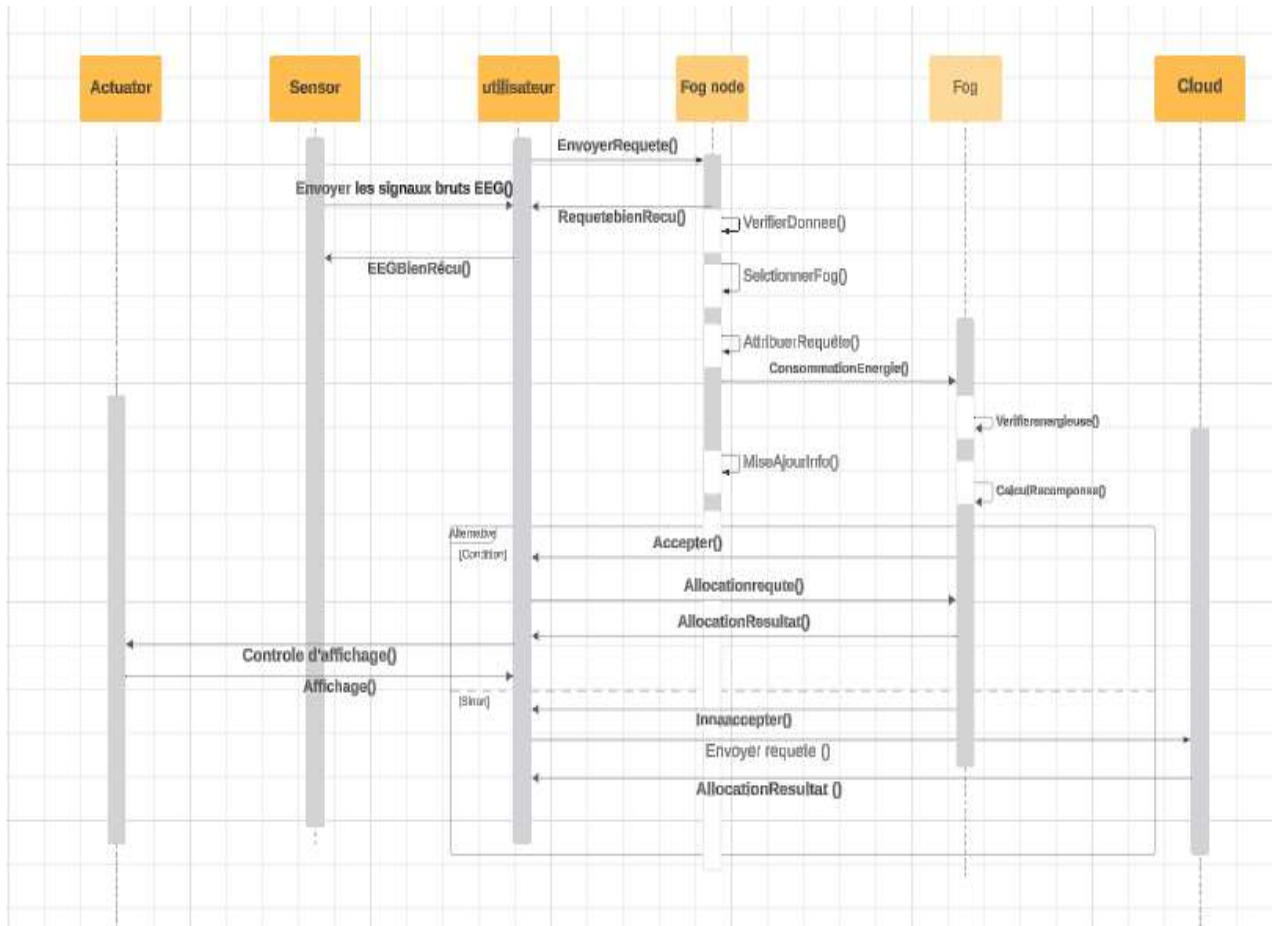


Figure 15-Diagramme de séquence.

3.5.1 Conditions d'allocation des ressources

Plusieurs critères définissent les conditions d'allocation des ressources comme il est indiqué dans l'équation suivante

$$\text{Conditions d'allocation des ressources} \Leftrightarrow \begin{cases} \text{Ram demandée} \leq \text{Ram du fog} \\ \text{Pes demanée} \leq \text{Pes du fog} \\ \text{Stockage demandé} \leq \text{Stockage du fog} \\ \text{Bande passante demandée} \leq \text{BW fog} \end{cases}$$

- RAM (Mémoire vive) : La quantité de RAM demandée par une requête doit être inférieure ou égale à la quantité de RAM disponible dans le Fog. Cela garantit que les ressources allouées peuvent être gérées efficacement sans compromettre les performances du système.
- Bande passante : La bande passante requise par les ressources demandées doit être inférieure ou égale à la bande passante disponible dans le Fog. Cela assure des transferts de données fluides et rapides entre les ressources et les utilisateurs finaux.
- PES (Puissance de calcul) : La puissance de calcul nécessaire pour traiter les demandes doit être inférieure ou égale à la capacité de calcul du Fog. Cela garantit que les opérations requises peuvent être effectuées de manière efficace sans surcharger le Fog.
- Stockage : La capacité de stockage demandée par les ressources doit être inférieure ou égale à la capacité de stockage disponible dans le Fog. Cela permet de garantir un accès rapide et fiable aux données nécessaires pour répondre aux demandes des utilisateurs.

3.5.2 Modèle d'allocation des ressources

Après avoir défini ce processus d'allocation et créé un diagramme de séquence pour visualiser les interactions entre les différents composants du système, nous pouvons maintenant mettre en place notre modèle d'allocation qui est présenté dans l'organigramme suivant :

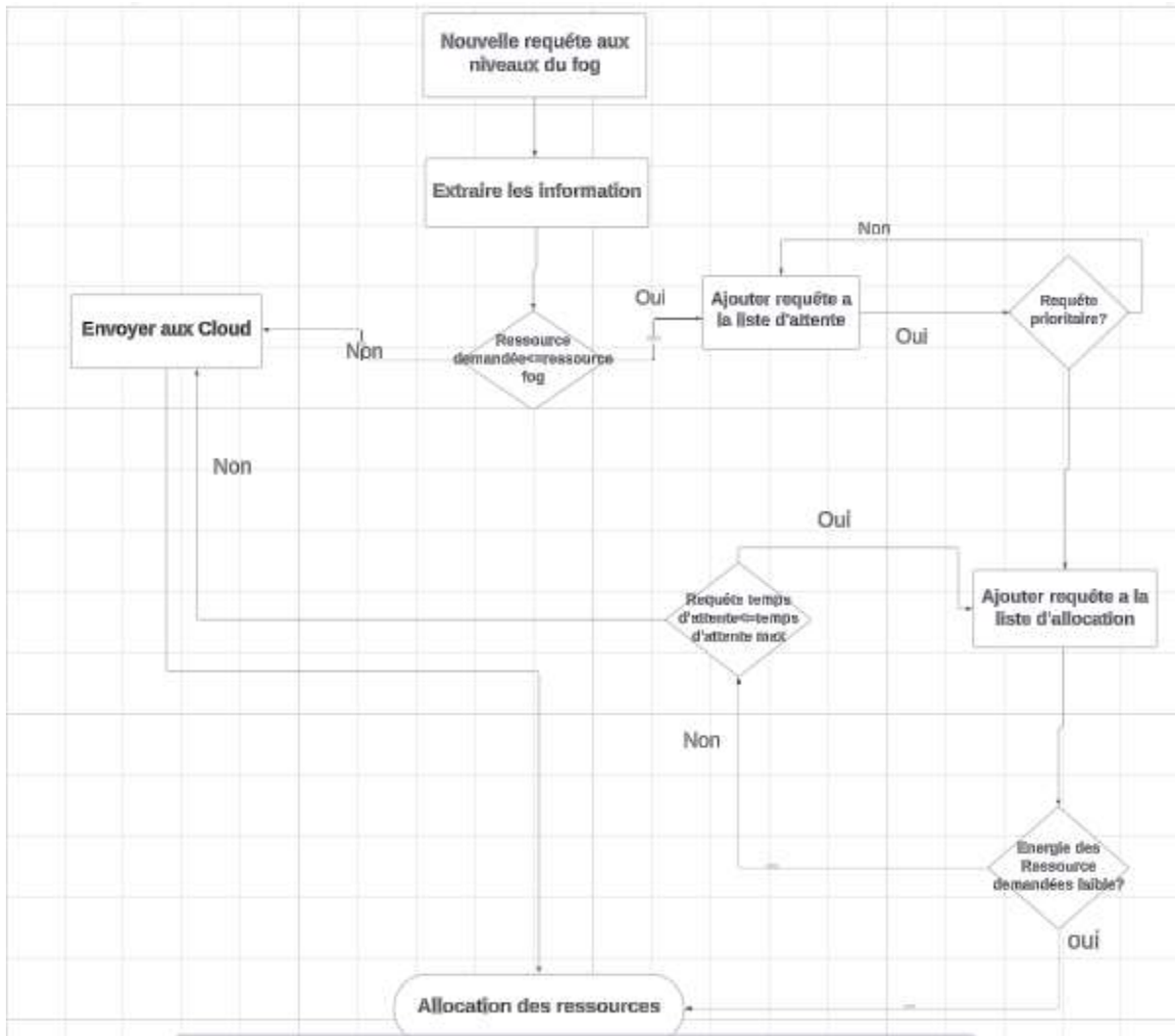


Figure 16– L’organigramme de processus d’allocation.

- **Description**

Lorsqu'une requête arrive au niveau du Fog, le processus débute par l'extraction de toutes les informations pertinentes nécessaires à l'évaluation de ses besoins en termes de ressources. Cela comprend les spécifications de la demande telles que la RAM, la puissance de calcul, le stockage, la bande passante, et les contraintes énergétiques spécifiques de l'utilisateur.

Une fois les détails de la requête capturée, le système effectue une comparaison

minutieuse entre les ressources demandées et celles disponibles localement dans le Fog. Si les ressources demandées sont suffisamment disponibles pour satisfaire la requête, celle-ci est placée dans une file d'attente pour être traitée. Pendant ce temps, le système classe les requêtes en fonction de leur priorité. Les requêtes critiques ou urgentes sont immédiatement dirigées vers une liste d'allocation, tandis que les autres sont conservées dans la file d'attente pour un traitement ultérieur.

Une étape essentielle du processus est la vérification de la consommation d'énergie associée aux ressources demandées. Si l'énergie de dispositif IOT est jugée insuffisante, la requête est considérée comme prioritaire pour une allocation rapide.

En cas d'indisponibilité des ressources le système prend alors la décision de transférer la requête vers le Cloud. Cette transition vers le Cloud permet d'obtenir les ressources nécessaires pour satisfaire la demande de manière efficiente, bien que cela puisse entraîner un léger délai supplémentaire en raison de la latence associée aux communications avec le Cloud.

En résumé, ce modèle qui est basé sur la priorisation des requêtes vise à assurer une efficacité opérationnelle optimale, permettant ainsi de répondre de manière adéquate et efficiente aux besoins d'un nombre maximal d'utilisateurs dans un environnement distribué.

3.5.3 Algorithme d'allocation des ressources

Pour une allocation efficace des ressources et pour améliorer la satisfaction des utilisateurs dans un environnement Fog computing, nous avons développé un algorithme innovant d'allocation de ressources avec gestion de file d'attente, visant à maximiser du nombre d'utilisateurs en minimisant les temps d'attente et la consommation d'énergie. Avant de présenter cet algorithme, nous allons d'abord exposer la structure d'une requête IoT puisqu'elle est considérée comme un élément indispensable pour déclencher le processeur d'allocation des ressources au niveau du Fog.

3.5.3.1 Requête IOT

Une requête IoT correspond à une communication organisée entre un appareil IoT et un serveur ou une plateforme Fog ou Cloud, généralement initiée par l'appareil afin de transmettre des données de capteurs ou de recevoir des instructions. La requête IoT se caractérise par différents attributs qui sont essentiels pour le processus d'allocation de ressource de notre approche. La requête est sous la forme suivante :

Requête_{Allocation} (ID, RAM, Stock, Pes, BW, Eng)

- ID : L'identification de l'utilisateur associé à l'appareil ou à l'application qui génère la requête.
- RAM : La quantité de mémoire vive (RAM) nécessaire pour traiter la requête.
- BW : La quantité de bande passante réseau requise pour transmettre ou recevoir les données de la requête.
- Stock : La quantité d'espace de stockage nécessaire pour stocker les données de la requête ou les résultats du traitement.
- Pes : Cette ressource se rapporte à la capacité des nœuds du Fog à exécuter les tâches de manière économe en énergie.
- Eng : les appareils IoT ont souvent des contraintes énergétiques. Il est important de gérer efficacement cette ressource pour prolonger la durée de vie des appareils.

3.5.3.2 Algorithme

```
1  Arrivée de Reqi; // Reqi est la requête qui demande l'allocation des ressources
2  ReqListfog ← ReqListfog ∪ Reqi; // Ajouter la requête à la liste des requêtes du Fog
3  foreach Reqi ∈ ReqListfog do
4  | RécupérerReqi informations ;// Récupérer la valeur du critère nécessaire
5  | AllocationList ← Reqi; //l'ajout se fait après le tri selon la valeur minimale du critère
6  | Maj ReqListfog ;// Supprimer la requête
7  end
8  foreach Reqj ∈ AllocationList do
9  | Allouer ressource à la requête prioritaire ;
10 | Supprimer Reqj de AllocationList ;
11 | Maj des ressources du Fog et informer l'utilisateur de Reqj ;
12 end
```

Figure17-Algorithme 1. Allocation des ressources.

L'algorithme ci-dessous représente un processus de gestion des demandes (Req) dans un environnement Fog.

- **Description**

À l'arrivée d'une nouvelle requête (Req), celle-ci est ajoutée à la liste des demandes en attente (ReqList_Fog), ensuite tant qu'il existe des demandes en attente dans ReqList_Fog, le système procède comme suit

- a) Il récupère de la requête les informations sur le critère nécessaire tel que la RAM ou le Pes ou l'énergie ou bien l'ensemble de ces critères afin de calculer la priorité de cette requête.
- b) Il ajoute la requête concernée à la liste d'allocation selon son ordre de priorité (tri par valeur minimale par exemple).

- c) Enlever cette requête de la liste des requêtes du Fog.
- d) Attribution des ressources à la requête prioritaire.
- e) Retirer la requête traitée de la liste d'allocation triée.
- f) Mettre à jours les ressources disponibles au niveau du Fog.
- g) Informer l'utilisateur IoT concerné par la décision d'allocation.

3.5.3.3 Pondération des contraintes

Il est crucial d'établir un système de scores basé sur plusieurs critères. Ces critères peuvent inclure la priorité de la requête, la consommation énergétique attendue, et d'autres facteurs pertinents. Voici une méthodologie détaillée pour calculer les pondérations et classer les requêtes :

- a) Calcul de poids : Pour calculer les poids des critères (RAM, Stockage, Bande passante, PES, Consommation d'énergie) selon la méthode de l'entropie, nous allons suivre les étapes suivantes :

1. Calcul de la proportion des critères :

- Proportion de RAM : $RAM_i = \frac{RAM_i}{\sum RAM}$
- Proportion de Stockage : $Stockage_i = \frac{Stockage_i}{\sum Stockage}$
- Proportion de bande passante : $Bande\ passante_i = \frac{Bande\ passante_i}{\sum Bande\ passante}$
- Proportion de puissance de traitement : $PES_i = \frac{PES_i}{\sum PES}$
- Proportion de l'énergie : $Energie_i = \frac{Energie_i}{\sum Energie}$

- 2. Calcul de l'entropie pour chaque critère : ensuite, calculons l'entropie pour chaque critère en utilisant la formule :

$$H(x_i) = - \sum_{j=1}^n p_{ij} \log(p_{ij})$$

Où p_{ij} est la proportion du critère i pour l'utilisateur j .

3. Calcul des poids des critères : enfin, calculons les poids des critères en utilisant la formule :

$$W_i = \frac{1 - H(x_i)}{\sum_{i=1}^n 1 - H(x_i)}$$

b) Normalisation des critères : appliquons la normalisation min max pour mettre les valeurs de chaque critère dans la plage de 0 à 1 :

$$\text{Valeur Normalisée} = \frac{\text{Valeur originale} - \text{Valeur minimale}}{\text{Valeur maximale} - \text{Valeur minimale}}$$

c) Calcul de la priorité pondérée

Pour chaque utilisateur i , calculez la priorité pondérée en utilisant les valeurs normalisées et les poids attribués à chaque critère. La formule de la priorité pondérée peut être exprimée comme suit :

$$\text{Priorite}_i = W_{\text{Rami}} \times \text{ValeurNormaliséRami} + W_{\text{stockage}_i} \times \text{ValeurNormaliséStockage}_i + W_{\text{BW}_i} \times \text{ValeurNormaliséBW}_i + W_{\text{Pesi}} \times \text{ValeurNormaliséPesi} + W_{\text{Energie}_i} \times \text{ValeurNormaliséEnergie}_i$$

➤ **Décision d'allocation de la requête**

Avec cette priorisation basée sur la fonction objective pondérée, allouez les requêtes des utilisateurs au nœud Fog commence par la requête ayant les valeurs de fonctions objectif les plus bases.

➤ **Exemple illustratif**

Voici les détails des ressources pour chaque utilisateur :

Tableau 2-Tableau des ressources d'utilisateurs

Utilisateurs	Ram	Stockage	Bande passante	Puissance de traitement	Consommation énergétique
Utilisateur 1	4	100	50	60	250
Utilisateur 2	2	80	70	80	200
Utilisateur 3	3	120	60	50	400
Utilisateur 4	5	150	80	70	320
Utilisateur 5	6	200	100	90	280

1. Calcul des proportions des critères

$$\text{Utilisateur1} \left\{ \begin{array}{l}
 \textit{Proportion RAM} = 0,20 \\
 \textit{Proportion Stockage} = 0,154 \\
 \textit{Proportion bande passante} = 0,139 \\
 \textit{Proportion PES} = 0,171 \\
 \textit{Proportion consommation d'energie} = 0,172
 \end{array} \right.$$

$$\text{Utilisateur2} \left\{ \begin{array}{l}
 \textit{Proportion RAM} = 0,10 \\
 \textit{Proportion Stockage} = 0,123 \\
 \textit{Proportion bande passante} = 0,194 \\
 \textit{Proportion PES} = 0,229 \\
 \textit{Proportion consommation d'energie} = 0,138
 \end{array} \right.$$

$$\text{Utilisateur3} \left\{ \begin{array}{l}
 \textit{Proportion RAM} = 0,15 \\
 \textit{Proportion Stockage} = 0,185 \\
 \textit{Proportion bande passante} = 0,167 \\
 \textit{Proportion PES} = 0,143 \\
 \textit{Proportion consommation d'energie} = 0,276
 \end{array} \right.$$

$$\text{Utilisateur4} \left\{ \begin{array}{l} \text{Proportion RAM} = 0,25 \\ \text{Proportion Stockage} = 0,231 \\ \text{Proportion bande passante} = 0,222 \\ \text{Proportion PES} = 0,200 \\ \text{Proportion consommation d'energie} = 0,221 \end{array} \right.$$

$$\text{Utilisateur5} \left\{ \begin{array}{l} \text{Proportion RAM} = 0,30 \\ \text{Proportion Stockage} = 0,308 \\ \text{Proportion bande passante} = 0,278 \\ \text{Proportion PES} = 0,257 \\ \text{Proportion consommation d'energie} = 0,193 \end{array} \right.$$

2. Les proportions des critères

Tableau 3–Tableau de calcul de l’entropie et de poids.

Ressources	Ram	Stockage	Bande passante	Puissance de traitement	Consommation énergétique
Entropie	2,578	1,560	0,467	1,602	1,802
Poids	0,785	0,391	0,372	0,420	0,560

3. Normalisation des critères

$$\text{Utilisateur1} \left\{ \begin{array}{l} \text{Valeur normalisée(RAM)} = 0,5 \\ \text{Valeur normalisée(Stockage)} = 0,167 \\ \text{Valeur normalisée(bande passante)} = 0 \\ \text{Valeur normalisée(PES)} = 0,25 \\ \text{Valeur normalisée(consommation d'energie)} = 0,25 \end{array} \right.$$

$$\text{Utilisateur2} \left\{ \begin{array}{l} \text{Valeur normalisée(RAM)} = 0 \\ \text{Valeur normalisée(Stockage)} = 0 \\ \text{Valeur normalisée(bande passante)} = 0,4 \\ \text{Valeur normalisée(PES)} = 0,75 \\ \text{Valeur normalisée (consommation d'energie)} = 0 \end{array} \right.$$

$$\text{Utilisateur3} \left\{ \begin{array}{l} \text{Valeur normalisée(RAM)} = 0,25 \\ \text{Valeur normalisée(Stockage)} = 0,33 \\ \text{Valeur normalisée(bande passante)} = 0,2 \\ \text{Valeur normalisée(PES)} = 0 \\ \text{Valeur normalisée (consommation d'energie)} = 1 \end{array} \right.$$

$$\text{Utilisateur4} \left\{ \begin{array}{l} \text{Valeur normalisée(RAM)} = 0,75 \\ \text{Valeur normalisée(Stockage)} = 0,58 \\ \text{Valeur normalisée(bande passante)} = 0,6 \\ \text{Valeur normalisée(PES)} = 0,5 \\ \text{Valeur normalisée (consommation d'energie)} = 0,6 \end{array} \right.$$

$$\text{Utilisateur5} \left\{ \begin{array}{l} \text{Valeur normalisée(RAM)} = 1 \\ \text{Valeur normalisée(Stockage)} = 1 \\ \text{Valeur normalisée(bande passante)} = 1 \\ \text{Valeur normalisée(PES)} = 1 \\ \text{Valeur normalisée (consommation d'energie)} = 0,4 \end{array} \right.$$

4. La priorité pondérée

Après avoir appliqué la pondération aux différents utilisateurs, nous obtenons les résultats suivants pour la priorité pondérée :

- Pondération pour l'utilisateur1 ≈ 0.70
- Pondération pour l'utilisateur2 ≈ 0.46
- Pondération pour l'utilisateur3 ≈ 0.96
- Pondération pour l'utilisateur4 ≈ 1.58
- Pondération pour l'utilisateur5 ≈ 2.192

En se basant sur ces résultats, nous allons classer les utilisateurs pour l'allocation des ressources selon leur priorité pondérée, de la plus basse à la plus élevée. Cela signifie que les utilisateurs ayant une pondération plus faible seront servis en premier, suivis par ceux ayant une pondération plus élevée. Voici l'ordre de priorité pour l'allocation :

Utilisateur 2, utilisateur1, utilisateur3, utilisateur 4, utilisateur 5

Conclusion

Ce chapitre a été consacré à la présentation de notre approche proposée pour l'allocation des ressources dans un environnement Fog computing. Cette stratégie vise à maximiser le nombre d'utilisateurs en optimisant les ressources, le temps de réponse, et la consommation d'énergie on prend en considération la priorité pondérée. Nous avons défini le processus d'allocation des ressources, expliqué les conditions spécifiques qui régissent ce processus, et illustré ces concepts à l'aide de diagrammes et de modèles d'application. De plus, nous avons introduit et décrit un algorithme pour prioriser et allouer efficacement les ressources disponibles. Dans le chapitre suivant, nous mettrons en œuvre notre stratégie en utilisant le simulateur iFogSim pour évaluer son efficacité.

Chapitre 4

Implémentation et résultats

4.1 Introduction

Dans ce chapitre, nous allons procéder à la phase de réalisation et d'implémentation de notre approche d'allocation des ressources dans un environnement Fog computing pour évaluer et valider son fonctionnement. Pour cela nous commencerons par décrire l'environnement de simulation dans lequel nous avons étendu le simulateur IFogSim pour réaliser plusieurs séries d'expérimentations puis nous discuterons et analyserons les résultats que nous avons obtenus.

4.2 Environnement de Développement

Il est crucial de prendre en compte l'environnement de développement afin de déterminer dans quelles situations le travail peut être reproduit. Nous avons mis en place et testé la stratégie que nous avons suggérée dans le cadre de ce projet de fin d'études 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) i3-4005U CPU @ 1.70GHz et d'une capacité mémoire de 4Go. Le simulateur est sous Windows10 64 bits.

- **Simulateur utilisé :** Nous avons développé le simulateur IFogSim afin de supporter notre proposition.
- **Langage utilisée :** Java.
- **Version Java Développement Kit utilisée :** JDK 11.
- **IDE utilisé :** Eclipse.

4.2.1 Langage de programmation Java

Java est un langage de programmation de haut niveau 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. La société Sun est rachetée en 2009 par la société Oracle qui détient et maintient désormais Java. Le code Java est d'abord écrit dans un Kit de développement Java, qui est disponible pour Windows, Linux et macOS. Les programmeurs écrivent dans le langage de programmation Java, que le kit traduit en code informatique qui peut être lu par n'importe quel appareil avec le logiciel approprié. Cela est obtenu avec un composant logiciel appelé compilateur. Un compilateur prend du code informatique de haut niveau comme Java et le traduit dans un langage que les systèmes d'exploitation comprennent appelé bytecode. Java est devenu aujourd'hui une direction incontournable dans le monde de la programmation [24]

Java est devenu aujourd'hui une direction incontournable dans le monde de la programmation et une des principales raisons de ce choix est que le simulateur iFogSim est développé avec ce langage. [24]

4.2.2 Environnement de développement

Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques.

Eclipse possède de nombreux points forts qui sont à l'origine de son énorme succès dont les principaux sont :

- Une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug-ins.
- Plusieurs versions d'un même plug-in peuvent cohabiter sur une même plate-forme.
- Un support multi langage grâce à des plug-ins dédiés : Cobol, C, PHP, C# , ...
- Support de plusieurs plateformes d'exécution : Windows, Linux, Mac OS X, ... [25]

4.2.3 Simulateur iFogSim

iFogSim est un simulateur d'environnements de Fog et d'IoT, Il s'agit d'une variante du simulateur de Cloud : CloudSim. La conception d'iFogSim vise à évaluer les stratégies de gestion des ressources utilisées dans les environnements de Fog et d'IoT en ce qui concerne leur influence sur la latence, la consommation d'énergie, la congestion du réseau et les dépenses opérationnelles. Les capteurs, les actionneurs, les nœuds de Fog et les centres de données de Cloud sont inclus dans un environnement de Fog et d'IoT avec iFogSim [26] .

L'architecture initiale d'iFogSim est illustrée dans la figure suivante (c'est-à-dire sans l'extension proposée). Ce simulateur comprend : (i) un ensemble d'éléments qui permettent de représenter les équipements physiques de l'infrastructure, (ii) un ensemble d'éléments qui permettent de représenter les éléments logiques du scénario simulé (comme les données et les instances des services de l'IoT), et (iii) un ensemble d'éléments qui gèrent les ressources de traitement dans les nœuds de Fog et les centres de données. On utilise ces ressources pour organiser et situer les instances de services dans les entités physiques, dans le but d'améliorer la latence du service, le trafic réseau, la consommation d'énergie ou les coûts opérationnels du système.

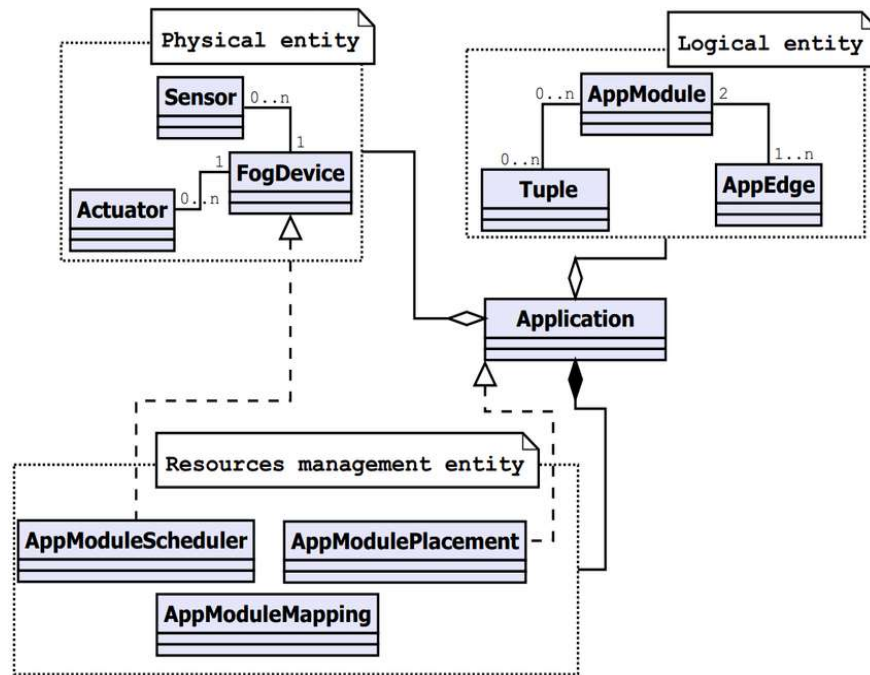


Figure 18 - Diagramme de classes de simulateur iFogSim.

4.3 Évaluation des Performances

Dans cette section nous réaliserons des simulations approfondies dans un environnement informatique Fog afin d'évaluer les différentes stratégies de notre approche d'allocation des ressources proposées à travers différentes études de cas d'applications IoT. Quatre paramètres clés ont été étudiés pour évaluer l'efficacité des stratégies : la mémoire vive (RAM), les unités de traitement (PES), la consommation d'énergie et la priorité pondérée de ces paramètres tels que les valeurs minimales du paramètre choisi doivent se traiter en priorité.

Le diagramme de cas d'utilisation suivant décrit la méthode d'utilisation de notre application.

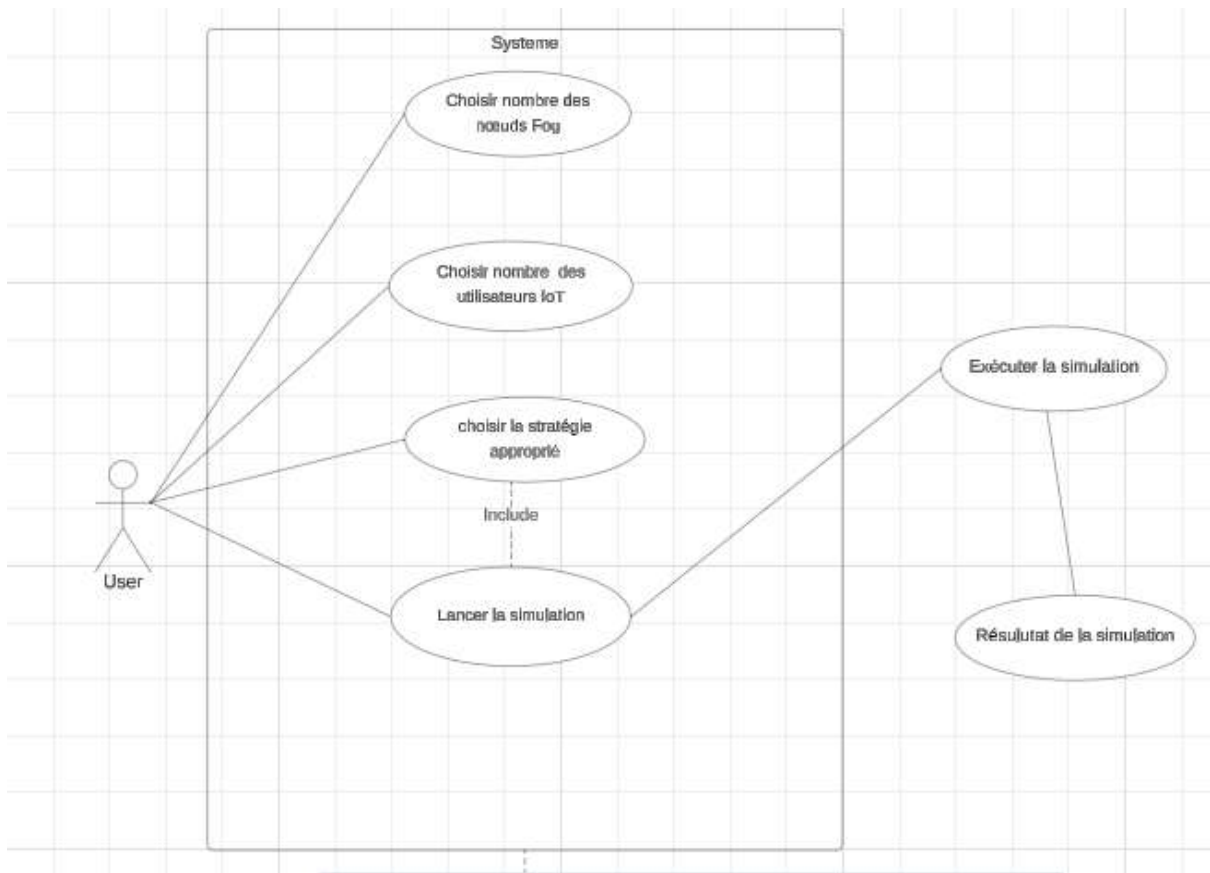


Figure 19– Diagramme de cas d’utilisation du système.

4.3.1 Configuration de la simulation

Nous avons mis en place une série de simulations qui ont utilisé divers scénarios d'utilisateurs IoT. Cinq scénarios différents ont été développés, chacun représentant un nombre croissant d'utilisateurs :

- Scénario 1 : 10 utilisateurs avec un seul Fog.
- Scénario 2 : 20 utilisateurs avec deux Fogs.
- Scénario 3 : 30 utilisateurs avec trois Fogs.
- Scénario 4 : 40 utilisateurs avec quatre Fogs.
- Scénario 5 : 50 utilisateurs avec cinq Fogs.

Il faut noter que pour chaque simulation a été réalisée pour un total de 5 exécutions et les valeurs moyennes sont prises pour tracer les graphiques. Pour évaluer le comportement de l'approche proposé, nous avons utilisé les métriques suivantes :

- Requête satisfaites : Le nombre de demandes de ressources IoT qui ont été pleinement satisfaites.
- Temps d'exécution : La durée totale nécessaire pour traiter toutes les demandes dans le scénario.
- Consommation d'énergie : La quantité d'énergie consommée des nœuds Fog pendant la simulation.
- Utilisation du réseau : Le total des données échangées au sein du réseau Fog, reflétant l'utilisation du réseau.
- Taux des ressources utilisées : La proportion de RAM et de PES (Processing Elements) utilisés pendant la simulation.

4.3.2 Analyse des résultats

Il est essentiel d'analyser les résultats des simulations. En analysant les informations concernant différentes mesures, comme le nombre de requêtes satisfaites, le temps de réponse total, la consommation d'énergie, l'utilisation du réseau et le taux d'utilisation des ressources (RAM et PES), nous pouvons identifier les points forts de notre approche. Cette section présente une analyse approfondie de ces résultats, accompagnée de graphiques comparatifs, dans le but de fournir des informations précieuses pour l'amélioration ultérieure des systèmes de Fog Computing.

4.3.2.1 Le nombre de requêtes satisfaites

Les tableaux ci-dessous présentent le nombre de requête satisfaites pour chaque paramètre à travers les cinq scénarios de configuration :

Tableau 4– Tableau des requêtes satisfaites

Configuration de Scénario	Min Ram	Min Pes	Min énergie	Min Pondération
Scénario1	6	7	8	8
Scénario2	11	11	12	14
Scénario3	18	18	20	21
Scénario4	20	22	23	25
Scénario5	27	29	29	33

Afin d'approfondir notre compréhension de la variation de la satisfaction des requêtes en fonction des différents paramètres et scénarios, nous avons converti ce tableau en un graphique. Chaque situation de scénario correspond à une augmentation du nombre d'utilisateurs, et le graphique permet de voir comment chaque paramètre répond aux demandes dans ces diverses situations.

Le graphique ci-dessous présente le nombre de requêtes satisfaites en fonction de nombre des utilisateurs IOT. Les différents scénarios sont déterminés par le nombre d'utilisateurs, variant de 10 à 50, et sont illustrés par les différentes barres colorées correspondant aux stratégies d'allocation qui réduisent respectivement la RAM, les PES (puissance de calcul), l'énergie et une pondération combinée de la RAM, PES et l'énergie. Nous pouvons observer que, globalement, le nombre de requêtes satisfaites augmente en fonction du nombre d'utilisateurs pour toutes les stratégies d'allocation. Selon cette tendance, il est possible que le système de Fog Computing puisse gérer une charge utilisateur croissante tout en continuant à fournir des performances satisfaisantes. Selon les résultats, la stratégie de pondération combinée (en vert) semble être la plus efficace, ce qui suggère une approche équilibrée qui optimise simultanément plusieurs ressources. Les différentes approches présentent des

différences selon les situations, mais toutes contribuent à une augmentation significative des demandes satisfaites en raison de l'augmentation du nombre d'utilisateurs.

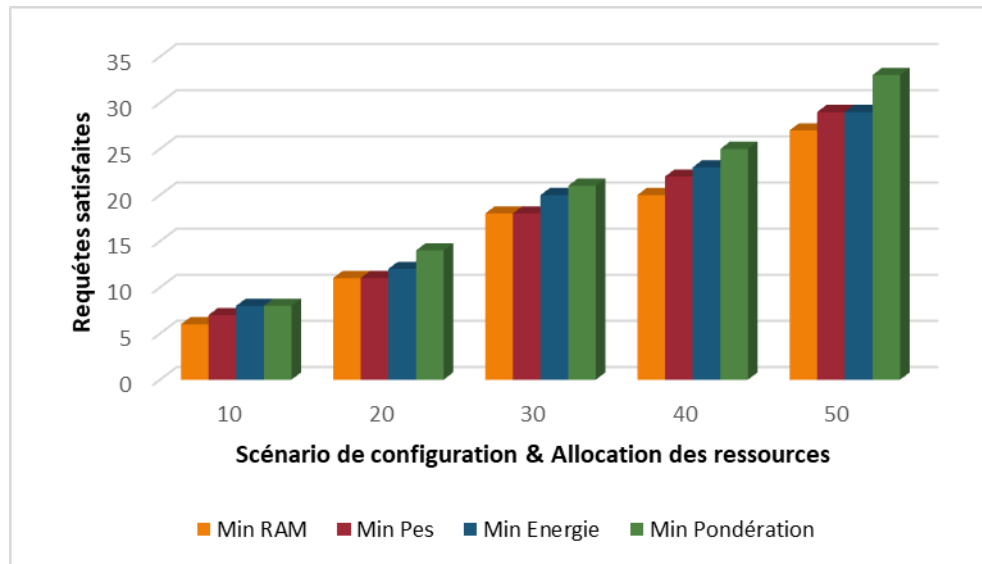


Figure 20- Graphe de configuration des requêtes satisfaites.

4.3.2.2 Le temps de réponse total

Les cinq scénarios de configuration montrent les variations du temps de réponse pour chaque paramètre dans le tableau ci-dessous

Tableau 5– Tableau de variation de temps de réponse

Configuration de Scénario	Min Ram	Min Pes	Min énergie	Min Pondération
Scénario1	118	128	106	108
Scénario2	148,6	145,8	148,4	150
Scénario3	142	191	194,6	192,4
Scénario4	208,5	195	211,6	214 ,5
Scénario5	265	206,2	266	265,8

Pour illustrer plus clairement les tendances observées, nous avons transformé ce tableau en un graphique pour mieux comprendre la variation du temps de réponse en fonction des différents paramètres et scénarios. Le nombre d'utilisateurs augmente dans chaque scénario et le graphique montre comment chaque paramètre affecte le temps de réponse dans ces différentes situations.

Le graphique ci-dessous représente les courbes de variation du temps de réponse en fonction du nombre d'utilisateurs IOT. On observe une tendance générale à l'augmentation du temps de réponse avec les différents paramètres. En particulier, la courbe associée à la stratégie Min Pondération présente les temps de réponse les plus élevés, variant de 600 à 1000, par rapport aux autres courbes. De plus, on remarque que le temps de réponse diminue considérablement dans le scénario à 10 utilisateurs par rapport aux scénarios à 20, 30, 40 et 50 utilisateurs. Cette corrélation directe entre le nombre d'utilisateurs et le temps de réponse souligne l'impact significatif de la charge utilisateur sur les performances du système. Dans toutes les configurations, le Fog Computing joue un rôle essentiel pour ces allocations. Par exemple, dans un cas de 10 utilisateurs, il est possible d'allouer de manière efficace les ressources à un seul nœud Fog. Toutefois, lorsqu'il y a 30 ou 40 utilisateurs, il est possible que les capacités d'un seul nœud Fog ne soient pas adéquates pour répondre à la demande, ce qui nécessite l'ajout de nouveaux nœuds Fog dans le but d'améliorer les capacités donc augmenter la possibilité d'allocation en minimisant le temps de réponse.

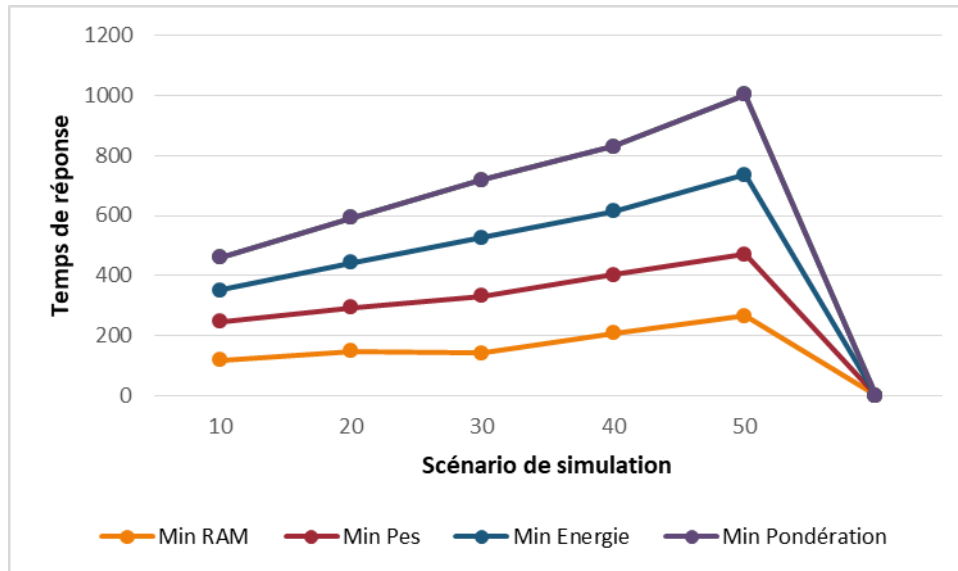


Figure 21 – Graphe des variations de temps de réponse.

4.3.2.3 La consommation d'énergie

Voici le tableau de différentes moyennes de consommation d'énergie au niveau des nœuds Fog :

Tableau 6– Tableau de la consommation d'énergie

Configuration de Scénario	Min Ram	Min Pes	Min énergie	Min Pondération
Scénario1	214625,506	214540,667	214437,727	214655,607
Scénario2	175910,132	115020,367	195264,142	214646,277
Scénario3	214615,6074	201776,807	214050,631	214649,389
Scénario4	214650,876	214651,441	202558,442	214588,807
Scénario5	246211,959	206043,833	246122,333	205042,922

En transformant le tableau en graphe, la figure ci-dessus présente un graphique qui analyse la moyenne de consommation d'énergie au niveau des nœuds Fog à travers divers scénarios, allant de 10 à 50 utilisateurs. Quatre configurations ont été comparées : Min Ram, Min Pes, Min énergie et Min pondération. La consommation d'énergie moyenne est généralement stable, se situant autour de 200 000 unités dans la plupart des cas. Toutefois, dans le deuxième scénario (avec 20 utilisateurs), la configuration Min Pes se caractérise par une consommation nettement inférieure, proche de 150 000 unités, tandis que les autres configurations sont proches de 200 000 unités. Malgré l'objectif de la configuration Min énergie d'optimiser la consommation d'énergie, elle ne présente pas toujours la consommation la plus faible, ce qui laisse entendre que d'autres facteurs ont également une influence sur la consommation totale. Dans les différents scénarios, la configuration Min pondération, qui équilibre plusieurs facteurs, a tendance à avoir une consommation légèrement réduite.

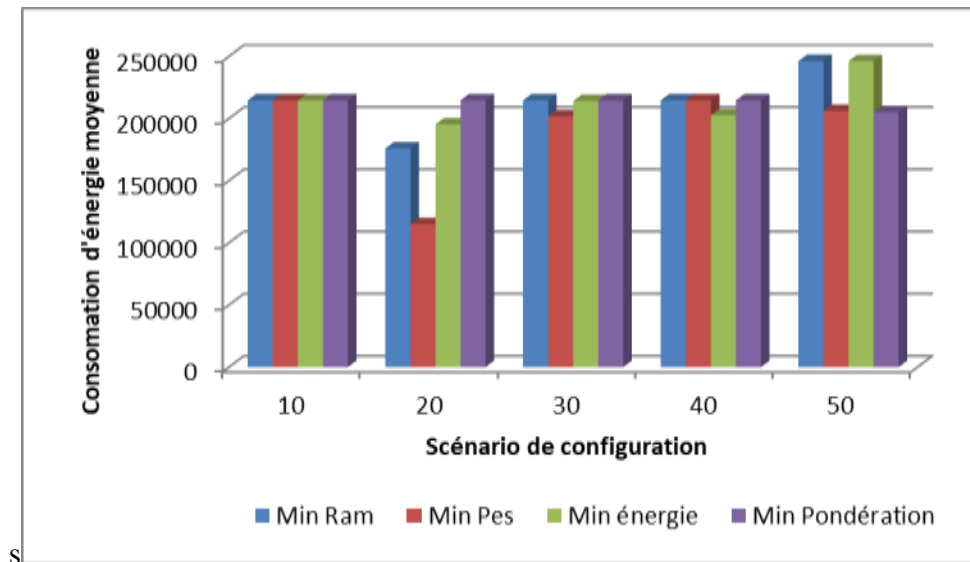


Figure 22–Graphe de consommation d'énergie.

4.3.2.4 L'utilisation du réseau

Voici un tableau récapitulatif présentant les variations de l'utilisation totale du réseau en fonction des différentes configurations de scénario :

Tableau 7 – Tableau de l’usage de réseau

Configuration de Scénario	Min Ram	Min Pes	Min énergie	Min Pondération
Scénario1	73	39	39	40
Scénario2	58	58	70	61
Scénario3	64,5	78,6	77	79,7
Scénario4	104,3	101	96,6	104,5
Scénario5	142	114,5	98	142

Après avoir modifié le tableau de calcul, on obtient le graphique mentionné précédemment.

Le graphique en bas illustre l'usage du réseau pour différents scénarios. On observe que l'usage du réseau est relativement équilibré pour tous les scénarios. Cependant, à mesure que le nombre d'utilisateurs augmente, l'usage total du réseau augmente également. Pour les différents scénarios, l'usage total du réseau est d'environ 150 pour le scénario 10, 250 pour le scénario 20, 350 pour le scénario 30, 400 pour le scénario 40 et 500 pour le scénario 50. Cette augmentation de l'usage du réseau s'explique par l'accroissement du nombre d'utilisateurs. Plus il y a d'utilisateurs, plus la demande en ressources réseau augmente.

En résumé, ce graphique permet d'évaluer comment différents paramètres de configuration affectent l'usage du réseau, et il met en lumière que l'usage du réseau augmente avec le nombre d'utilisateurs en raison de la demande croissante en ressources réseau. De plus le Fog Computing a la capacité d'équilibrer cet usage grâce à une augmentation proportionnelle du nombre de nœuds de Fog, ce qui permet de maintenir un usage efficace et équilibré des ressources réseau à chaque scénario.

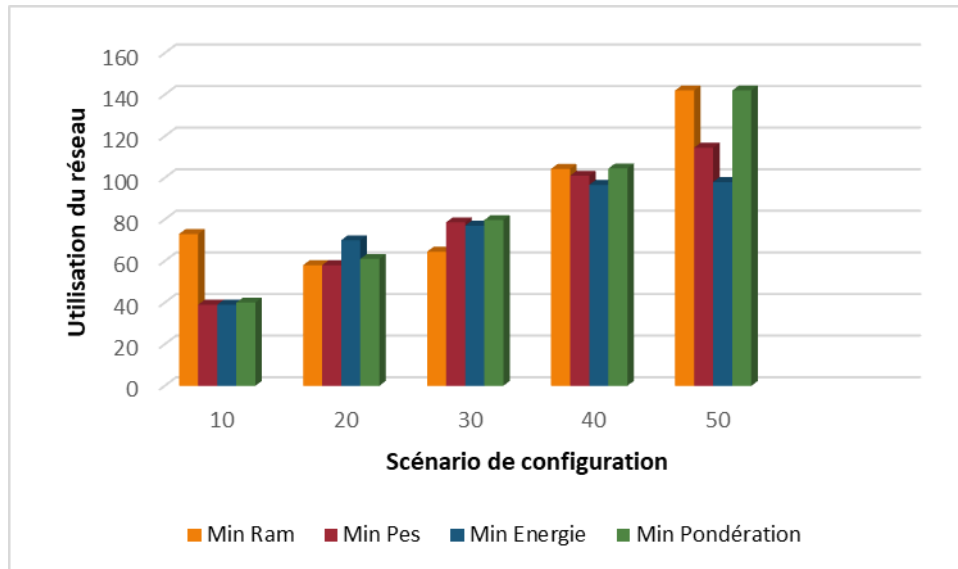


Figure 23-Graphe de l'utilisation du réseau.

4.3.2.5 Le taux d'utilisation des ressources (RAM et PES)

Voici un tableau présentant les taux d'utilisation des ressources (RAM et PES) en fonction des différentes configurations de scénario :

Tableau 8– Tableau des ressources consommées

Configuration de Scénario	Min Ram	Min Pes
Scénario1	72,74	75,05
Scénario2	76,57	72,7
Scénario3	78,29	73,35
Scénario4	74,98	72,96
Scénario5	73,32	73,26

En transformant le tableau de calcul en obtient le graphe ci-dessous.

Cet histogramme à barres verticales illustre une comparaison entre deux ensembles de ressources, à savoir "Min Ram" en rouge et "Min Pes" en orange, à travers divers scénarios de simulation numérotés. Dans le premier scénario de 10 utilisateurs, il est observé que la consommation de RAM est inférieure à celle de PES, ce qui indique que les ressources nécessaires pour ces utilisateurs en termes de PES sont supérieures à celles de RAM. Il est également observé que dans les scénarios 2, 3, et 4, le taux de consommation de RAM est supérieur au taux de consommation de PES. Cela suggère que les utilisateurs de 10, 20 et 30 nécessitent beaucoup plus de RAM que de PES. D'autre part, dans le dernier cas, les deux ressources (RAM et PES) sont équilibrées. En bref, ce graphique offre la possibilité d'examiner et de comparer la consommation des ressources pour chaque type (Min Ram et Min Pes) à travers diverses simulations. En résumé, il est possible de conclure que les besoins en ressources diffèrent selon les scénarios, avec une tendance générale à une sollicitation accrue de la RAM dans les scénarios intermédiaires, tandis que la consommation des ressources devient plus équilibrée dans les scénarios plus extrêmes.

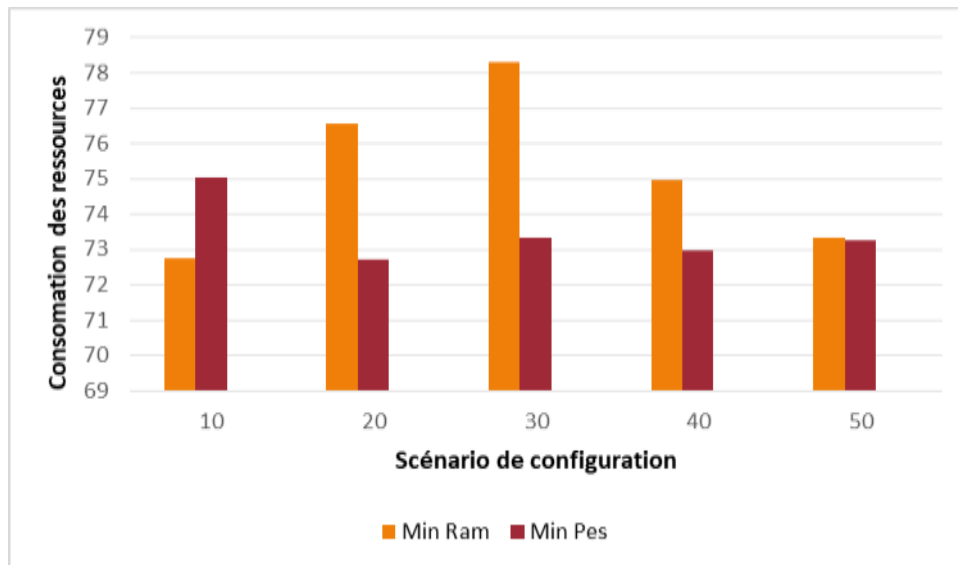


Figure 24-Graphe de consommation des ressources (Ram,Pes) .

4.4 Conclusion

Au cours de ce chapitre, nous avons présenté notre environnement de travail et nous avons exposé l'outil de simulation iFogSim que nous avons étendu pour simuler notre approche d'allocation des ressources dans un environnement Fog Computing, ainsi que les différentes méthodes utilisées. Plusieurs scénarios ont été simulés pour évaluer notre proposition en utilisant différents paramètres de configuration. Selon l'analyse des résultats des simulations, il est évident que l'approche suggérée à maximiser le nombre d'utilisateurs Iot, tout en optimisant la consommation d'énergie et en réduisant l'utilisation de la puissance de traitement (Pes), la Ram et la priorité pondérée des applications malgré l'augmentation de la taille des données demandées. Les résultats obtenus nous ont donné la possibilité de repérer les stratégies les plus efficaces et adaptées à différents scénarios d'application, ce qui permet d'optimiser l'allocation des ressources dans un environnement de Fog Computing.

Conclusion Générale

Les applications de traitement de données intensives traitent des volumes extrêmement importants. La gestion traditionnelle de ces données s'appuie sur des solutions basées sur le Cloud pour fournir un ensemble centralisé et riche. Les avantages des frameworks basés sur le Cloud sont leur omniprésence, leur capacité de ressources illimitée, leur rapport coût / efficacité, ainsi que leur élasticité. Cependant, accéder aux données depuis le Cloud implique un trafic réseau important, des latences de données élevées et des risques de sécurité plus élevés. A la différence du Cloud Computing, le Fog Computing est au plus près des objets connectés. Il facilite le stockage et l'analyse de données.

Dans le travail présent, nous avons étudié le problème d'allocation des ressources dans le Fog computing. Dans ce domaine l'allocation efficace des ressources est essentielle pour optimiser les performances du système, elle fournit les moyens nécessaires. Deux stratégies clés souvent utilisées sont la maximisation du nombre d'utilisateurs servis et la minimisation de la consommation d'énergie.

En adoptant une approche axée sur la maximisation du nombre d'utilisateurs servis, les fournisseurs de services Fog peuvent garantir une meilleure satisfaction client en offrant des temps de réponse plus courts et en réduisant les temps d'attente. Cela peut conduire à une utilisation plus efficace des ressources disponibles et à une meilleure utilisation de la capacité de traitement du Fog, contribuant ainsi à une utilisation plus efficiente des infrastructures distribuées. D'autre part, la minimisation de la consommation d'énergie est cruciale pour réduire les coûts opérationnels associés à l'infrastructure Fog.

Une approche équilibrée combinant la maximisation du nombre d'utilisateurs servis et la minimisation de la consommation d'énergie est essentielle pour garantir l'efficacité et la

durabilité des systèmes de Fog computing. Cela nécessite un algorithme d'allocation adaptatif pour répondre aux besoins croissants des utilisateurs en termes des ressources.

Pour mettre en évidence notre approche, nous avons étendu le simulateur iFogSim pour évaluer notre stratégie d'allocation des ressources qui représente une plateforme de simulation pour les environnements de calcul en périphérie dans les Fogs computing. Cette plateforme nous a permis d'évaluer les diverses métriques notamment Min-RAM, Min-PES, Min-Energie et la priorité pondérée. À partir de cette évaluation des métriques, nous avons pu obtenir des résultats concernant les requêtes satisfaites, les variations de temps de réponse, l'utilisation du réseau, la consommation d'énergie, et les ressources consommées. Ces résultats nous permettent de visualiser et d'analyser les compromis entre les différentes mesures pour optimiser notre approche.

Pour enrichir notre approche, nous envisageons quelque perspective pour l'avenir tels que l'intégration de l'aspect économique, En incorporant un modèle économique, il serait possible d'améliorer l'allocation des ressources en tenant compte des coûts d'exploitation, de la tarification dynamique et du retour sur investissement, ce qui rendrait notre plateforme plus rentable sous rapport financier. De plus, nous envisageons de mettre en place un système de coopération entre nœuds de Fog voisins. Ce système faciliterait le partage efficace de la charge de travail entre ces nœuds. Par exemple, en l'absence de ressources adéquates, un nœud de Fog pourrait confier les tâches à un nœud voisin possédant les compétences nécessaires, ce qui améliorerait l'efficacité globale et diminuerait les délais de réponse. Grâce à ces progrès, nous serons en mesure de mieux comprendre et étudier les compromis entre diverses mesures, ce qui permettra d'optimiser de manière globale notre stratégie d'allocation des ressources.

Bibliographie

Articles de revue

- [3] SATYANARAYANA, S. Cloud computing: SAAS. *Computer Sciences and Telecommunications*, 2012, no 4, p. 76-79.
- [4] LANJEWAR, Madhusudan G. ET PANCHBHAI, G. Convolutional neural network based tealeaf disease prediction system on smart phone using paas Cloud. *Neural Computing and Applications*, 2023, vol. 35, no 3, p. 2755-2771.
- [5] HAY, Brian, NANCE, Kara, ET BISHOP, Matt. Storm Clouds rising: security challenges for IaaS Cloud computing. In: 2011 44th Hawaii International Conference on System Sciences. IEEE, 2011. p. 1-7.
- [6] ELETTRE, Christian, BOUDAUD, Karima, et RIVEILL, Michel. Cloud Computing, Sécurité et Dissimulation de Données. In : 7ème Conférence sur la Sécurité des Architectures Réseaux et Systèmes d'Information (SARSSI'12). 2012.
- [8] Les tendances, et les developpements du, computing, Cloud, et sur-site, et les solutions IT. traduction officielle d'un document écrit en anglais. 2021.
- [9] HUTH, Alexa et CEBULA, James. The basics of Cloud computing. *United States Computer*, 2011, p. 1-4
- [10] ISLAM, Mohammad Manzurul, MORSHED, Sarwar, et GOSWAMI, Parijat. Cloud computing: A survey on its limitations and potential solutions. *International Journal of Computer Science Issues (IJCSI)*, 2013, vol. 10, no 4, p. 159.
- [11] KACHA, Lynda et ZITOUNI, Abdelhafid. An overview on data security in Cloud computing. *Cybernetics Approaches in Intelligent Systems: Computational Methods in Systems and Software 2017*, vol. 1, 2018, p. 250-261.
- [12] MARTINEZ, Ismael, HAFID, Abdelhakim Senhaji, et JARRAY, Abdallah. Design, resource management, and evaluation of Fog computing systems: A survey. *IEEE Internet of Things Journal*, 2020, vol. 8, no 4, p. 2494-2516
- [13] BOURHIM, El Houssine. La communication et le placement de conteneurs Docker dans le Fog computing. 2020.
- [15] YI, Shanhe, HAO, Zijiang, QIN, Zhengrui, et al. Fog computing: Platform and Applications. In : 2015 Third IEEE workshop on hot topics in web systems and Technologies (HotWeb). IEEE, 2015. p. 73-78

- [16] MEZAAL, Yaqeen S., SHAREEF, Mustafa S., ALAMERI, Ban M., et al. Emerging wireless communication technologies in Iraqi government: Exploring Cloud, edge, and Fog computing. *Heritage and Sustainable Development*, 2024, vol. 6, no 1, p. 169-182.
- [17] ALHWAITAT, AHMAD K., MANASEER, SAHER, et AL-SAYYED, RIZIK MH. A survey of digital forensic methods under advanced persistent threat in Fog computing environment. *Journal of Theoretical and Applied Information Technology*, 2019, vol. 97, no 18, p. 4934-4954.
- [21] Ranesh Kumar Naha, M.Sc., University Putra Malaysia School of Information and Communication Technology College of Sciences and Engineering (CoSE)
- [27] Lina Ni, Jinqun Zhang, Changjun Jiang, Chungang Yan, et Kan Yu. Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets. *IEEE Internet of Things Journal*, 4(5):1216–1228, 2017.
- [28] Phillip Kendrick, Thar Baker, Zakaria Maamar, Abir Hussain, Rajkumar Buyya, and Dhiya Al-Jumeily. An efficient multi-Cloud service composition using a distributed multiagent-based, memory-driven approach. *IEEE Transactions on Sustainable Computing*, pages 1–13, 2018.
- [29] Indukuri R Krishnam Raju, Penmasta Suresh Varma, MV Rama Sundari, and G Jose Moses. Deadline aware two stage scheduling algorithm in Cloud computing. *Indian Journal of Science and Technology*, 9(4), 2016.
- [30] Mohammed Dighriri, Ali Saeed Dayem Alfoudi, Gyu Myoung Lee, Thar Baker, and Rubem Pereira. Resource allocation scheme in 5g network slices. In *2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 275–280. IEEE, 2018.
- [31] Li Liu, Miao Zhang, Rajkumar Buyya, and Qi Fan. Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in Cloud computing. *Concurrency and Computation: Practice and Experience*, 2016.
- [32] Antonio Brogi and Stefano Forti. Qos-aware deployment of iot applications through the Fog. *IEEE Internet of Things Journal*, 4(5):1185–1192,2017.
- [33] Antonio Brogi, Stefano Forti, and Ahmad Ibrahim. How to best deploy your Fog applications, probably. In *1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017, Madrid, Spain, May 14-15*, pages 105–114, 2017.
- [34] Olena Skarlat, Matteo Nardelli, Stefan Schulte, Michael Borkowski, and Philipp Leitner. Optimized IoT service placement in the Fog. *Service Oriented Computing and Applications*, 11(4):427–443, 2017.

- [35] Jianbo Du, Liqiang Zhao, Jie Feng, and Xiaoli Chu. Computation offloading and resource allocation in mixed Fog/Cloud computing systems with min-max fairness guarantee. *IEEE Transactions on Communications*, 66(4):1594–1608, 2018.
- [36] Lei Zhang and Jiangtao Li. Enabling robust and privacy-preserving resource allocation in Fog computing. *IEEE Access*, 6:50384–50393, 2018.
- [37] Ye Yu, Xiangyuan Bu, Kai Yang, Zhikun Wu, and Zhu Han. Green large-scale Fog computing resource allocation using joint benders decomposition, dinkelbach algorithm, admm, and branch-and-bound. *IEEE Internet of Things Journal*, pages 1–12, 2018. 2018.
- [38] Rodrigo AC da Silva and Nelson LS da Fonseca. Resource allocation mechanism for a Fog-Cloud infrastructure. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2018.

Rapports de thèse

- [7] CYLIA, Guellal. Etude et mise en oeuvre d'un portail web pour la gestion d'un Cloud Privé de type IaaS cas Ooredoo. 2016. Thèse de doctorat. Université Mouloud Mammeri.
- [14] DJEMAI, Tanissia. Placement optimisé de services dans les architectures Fog computing et internet of things sous contraintes d'énergie, de QoS et de mobilité. 2021. Thèse de doctorat. Université Paul Sabatier-Toulouse III.
- [18] BITAILLOU, Alexis, PARREIN, Benoît, et ANDRIEUX, Guillaume. Synthèse sur les protocoles de communication pour l'Internet des objets de l'industrie 4.0. 2019. Thèse de doctorat. LS2N, Université de Nantes; IETR, Université de Nantes.
- [19] CHERRIER, Sylvain. Architecture et protocoles applicatifs pour la chorégraphie de services dans l'Internet des objets. 2013. Thèse de doctorat. Université Paris-Est.
- [20] NAAS, Mohammed Islam. Placement des données de l'internet des objets dans une infrastructure de Fog. 2019. Thèse de doctorat. Université de Bretagne occidentale-Brest.
- [22] BI, Fan. Resource allocation methods for Fog computing systems. 2022. Thèse de doctorat. University of Southampton
- [23] MEBREK, A. Fog Computing pour l'Internet des objets. 2020. Thèse de doctorat. Université UTT.
- [39] Mohamed Islam NAAS, Rapport technique n°601, Département de mathématiques et Science et Technologie de l'information. Université Bretagne Loire.

Documents web

- [1] <https://www.futura-sciences.com/tech/definitions/informatique-Cloud-computing-11573>
consulté le 8.02.2024
- [2] <https://datascientest.com/Cloud-computing-tout-savoir>, consulté le 8.02.2024
- [5] <https://Cloud.google.com/learn/what-is-iaas?hl=fr>, consulté le 16.02.2024
- [24] <https://azure.microsoft.com/fr-fr/resources/Cloud-computing-dictionary>, consulté le 29.03.2024
- [25] <https://jmdoudoux.developpez.com/cours/developpons/eclipse>, consulté le 05.04.2024
- [26] <https://fr.slideshare.net/omrimed/i-Fog-sim>, consulté le 05.04.2024
- [27] <https://www.e-spincorp.com/fog-computing-architecture-as-the-extension-of-cloud-computing/>, consulté le 4.06.2024