République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études
Pour l'obtention du diplôme de Master en Informatique
Option: Réseaux et Systèmes Distribués (RSD)

Thème

Optimisation du placement des applications et équilibrage de charge dans un environnement Fog Computing

Réalisé par :

- CHEMLOUL Fadila Sarra
- FEROUANI Anfel

Présenté le 28/06/2025 devant le jury composé de :

- *SETTOUTI Ahmed Khalid Yassine*	*Président*
- *AMRAOUI Asma*	*Encadrante*
- *ARICHI Bochra*	*Co-Encadrante*
- *GAOUAR Nihal*	*Examinatrice*

Année universitaire : 2024 – 2025

Remerciements

Avec un immense plaisir, ces quelques lignes sont dédiées à exprimer une profonde gratitude et une sincère reconnaissance à toute personne ayant contribué, de près ou de loin, à l'accomplissement de ce travail.

Avant toute chose, nous souhaitons adresser notre profonde gratitude et nos remerciements les plus sincères à notre Encadrante Mme. AMRAOUI Asma. Son dévouement constant, sa patience exemplaire, ses conseils pertinents et son suivi rigoureux se sont avérés essentiels pour l'aboutissement de ce travail.

Nous tenons également à témoigner notre gratitude la plus sincère à notre co-encadrante Mme. ARICHI Bochra. Ses brillantes intuitions, son accompagnement indéfectible, ses orientations avisées, sa constante disponibilité, ses remarques et l'aide qu'elle nous a apportée pour surmonter les défis rencontrés.

Ainsi, nous souhaitons adresser nos plus vifs remerciements aux membres du jury Mr. SETTOUTI et Mme GAOUAR pour avoir consacré une partie de leur temps à la lecture de ce mémoire et pour l'intérêt qu'ils ont porté à ce travail.

Nous exprimons nos remerciements à tous nos enseignants du département d'Informatique de l'Université Abou Bakr Belkaid de Tlemcen pour l'encadrement et la formation de qualité qu'ils nous dispensent.

Nous ne saurions achever ce mémoire sans remercier chaleureusement toutes les personnes qui, de près ou de loin, ont apporté leur aide, leur soutien ou leurs encouragements.

Dédicaces

***J**e souhaite offrir humblement ce travail*

***A** mes très chers parents pour tous les sacrifices qu'ils ont faits et pour tout le soutien qu'ils ont offert tout au long de mes études.*

***A** ma chère tante Latifa, qui a su être pour moi une seconde maman, un cœur attentif et une présence rassurante,*

***A** mes sœurs Ines et Zineb, deux étoiles toujours présentes dans mon ciel,*

***A** ma première amie d'enfance, celle qui connaît mes silences, partage mes rires et soutient mes rêves,*

***E**nfin Je dédie ce travail à moi-même, C'est le temps D'y être !*

Fadila Sarra

***J**e dédie ce travail*

***A** celle qui m'a donné la vie, mon espoir, la lumière de mon existence, la source de mon bonheur et ma réussite, à ma mère*

***A** mon père, mon protecteur depuis ma naissance, mon école depuis mon enfance,*

***A** mon frère Mohamed, mes chères sœurs Douàa et Abir pour leur amour et leur incontestable appui,*

***A** ma famille et à tous mes amies.*

***A** tous ceux qui m'aiment, et tous ceux que j'aime.*

Anfel

Table des matières

GLOSSAIRE

Acronyme	Signification
ACO	**A**nt **C**olony **O**ptimization
AR	**A**ugmented **R**eality
AWS	**A**mazon **W**eb **S**ervices
ARM	**A**dvanced **R**ISC **M**achine
BM	**B**roker **M**anagement
BP	**B**ande **P**assante
CB	**C**entralized **B**roker
CC	**C**loud **C**omputing
CE	**C**ost **E**fficiency
CH	**C**onsistent **H**ashing
CISCO	**C**omputer **I**nformation **S**ystem **CO**mpany
CPU	**C**entral **P**rocessing **U**nit
C-RAN	**C**loud **R**adio **A**ccess **N**etwork
DL	**D**eep **L**earning
DRL	**D**eep **R**einforcement **L**earning
EC	**E**nergy **C**onsumption
EWMP	**E**fficient **V**irtual **M**odule **P**lacement
FAAL	**F**og-based **A**mbient **A**ssisted **L**iving
FC	**F**og **C**omputing
FCL	**F**og **C**omputational **L**ayers
FGL	**F**og **G**ateway **L**ayers
FN	**F**og **N**odes
FNPA	**F**og **N**ode **P**lacement **A**lgorithm
F-RAN	**F**og **R**adio **A**ccess **N**etwork
GUI	**G**raphical **U**ser **I**nterface
GVMP	**G**lobal **V**irtual **M**odule **P**lacement
IA	**I**ntelligence **A**rtificielle
IDE	**I**ntegrated **D**evelopment **E**nvironment
IIC	**I**ndustrial **I**nternet **C**onsortium
INTEL	**INT**egrated **EL**ectronics

IOMT	**I**nternet **O**f **M**edical **T**hings
IOHT	**I**nternet **O**f **H**ealth **T**hings
IOT	**I**nternet **O**f **T**hings
JDK	**J**ava **D**evelopment **K**it
JFC	**J**ava **F**oundation **C**lasses
JSON	**J**ava**S**cript **O**bject **N**otation
JRE	**J**ava **R**untime **E**nvironment
JVM	**J**ava **V**irtual **M**achine
LAB	**L**oad **A**ware **B**alancing
LB	**L**oad **B**alancing
LBRM	**L**oad **B**alancing **R**esource **M**anagement
LC	**L**east **C**onnections
LRT	**L**east **R**esponse **T**ime
MICROSOFT	**MICRO**computer **SOFT**ware
NIST	**N**ational **I**nstitute of **S**tandards and **T**echnology
NP	**N**on **D**eterministic
OFC	**O**pen **F**og **C**onsortium
OLB	**O**nline **L**oad **B**alancing
OSGI	**O**pen **S**ervice **G**ateway **I**nitiative
QOS	**Q**uality **O**f **S**ervice
QSD	**Q**uality of **S**ervice **D**ifferentiation
RBAC	**R**ole **B**ased **A**ccess **C**ontrol
RHM	**R**emote **H**ealth **M**onitoring
RR	**R**ound **R**obin
SLA	**S**ervice **L**evel **A**greement
SPOF	**S**ingle **P**oint **O**f **F**ailure
TO	**T**ask **O**ffloading
UC	**U**biquitous **C**omputing
WRR	**W**eighted **R**ound **R**obin
5G	**C**inquième **G**énération

Listes des figures

Listes des tableaux

Listes des équations

Résumé:

Minimiser la latence et améliorer la qualité de service dans un environnement Fog Computing représentent des défis majeurs pour les applications temps réel comme la e-santé. Dans ce projet de fin d'études, nous avons étudié deux approches hybrides afin d'optimiser le placement des applications et équilibrer la charge entre les nœuds : la combinaison Weighted Round Robin (WRR) avec Least Connections (LC), ainsi que l'algorithme méta-heuristique Ant Colony Optimization (ACO) également associé à LC. L'objectif est de déterminer laquelle de ces approches offre les meilleures performances en termes de répartition des applications et de réactivité du système. Les simulations effectuées dans cette étude ont été réalisées à l'aide du simulateur iFogSim et les résultats obtenus sont jugés satisfaisants, montrant une amélioration significative en termes de latence et de temps de réponse.

Mots clés : Fog Computing, Équilibrage de charge ,WRR, ACO, LC, Réduction de la latence ,Qualité de Service ,iFogSim.

Abstract:

Minimizing latency and enhancing the QoS in a Fog Computing environment are major challenges for real-time applications such as e-health. In our graduation project, our work focuses on hybrid approaches to optimize the applications placement and load balancing between nodes: the combination Weighted Round Robin (WRR) with Least Connections (LC), and the metaheuristic algorithm Ant Colony Optimization (ACO) which is also associated with (LC). The objective is to determine which of these approaches offers the best performance in terms of the distribution of applications and the system's responsiveness. The simulations performed in this study were realized with the iFogSim simulator. The results obtained are considered satisfactory, showing a significant improvement in terms of latency and response time.

Keywords: Fog Computing, Load Balancing, WRR, ACO, LC, Latency reduction, QoS, iFogSim.

ملخص:

يعد تخفيض زمن الاستجابة وتحسين جودة الخدمة في بيئة الحوسبة الضبابية من التحديات الرئيسية لتطبيقات الوقت الحقيقي مثل الصحة الإلكترونية. في مشروع تخرجنا هذا، يرتكز عملنا على الاساليب الهجينة لتحسين وضع التطبيقات وموازنة توزيع الاحمال بين العقد: الجمع بين خوارزمية WRR مع LC، وخوارزمية التحسين بالاستدلال الفوقي ACO المرتبطة ايضا بLC .الهدف هو تحديد أي من هذه الأساليب يقدم أفضل أداء من حيث توزيع التطبيقات واستجابة النظام. تم إجراء المحاكاة في هذه الدراسة باستخدام محاكي iFogSim وتعتبر النتائج التي تم الحصول عليها مُرضية، حيث أظهرت تحسناً كبيراً من حيث زمن الاستجابة وزمن الاستجابة.

الكلمات المفتاحية: الحوسبة الضبابية، موازنة توزيع الاحمال،WRR، ACO، LC ، تقليل زمن الاستجابة، جودة الخدمة، iFogSim

Introduction Générale

Introduction générale

L'époque actuelle est marquée par une avancée technologique et progresse à un rythme fulgurant, donnant naissance à des solutions de plus en plus sophistiquées, parmi celles-ci, les appareils IoT jouent un rôle essentiel dans de nombreux domaines, y compris le secteur de santé. Ces dispositifs intelligents, capables de collecter, transmettre et analyser des données en temps réel, sont désormais utilisés pour surveiller à distance l'état de santé des patients, tel que le moniteur de signes vitaux portable. Grâce à cette évolution, la médecine devient plus connectée, mais cela entraîne la génération de vastes quantités de données. Cette croissance continue souligne à quel point il est crucial de gérer et de stocker efficacement les données générées.

Le Fog Computing (FC) est une solution très pertinente pour gérer les volumes massifs de données générées. Il a été créé pour répondre aux besoins croissants des systèmes IoT et améliorer la réactivité des applications connectées. Il a été conceptualisé autour de **2012** [19], c'est une extension du Cloud, qui vise à traiter et stocker les données plus près des utilisateurs finaux, en utilisant des dispositifs locaux comme : des passerelles ou des routeurs. Il permet de décentraliser le traitement des données.

L'intérêt principal du FC réside dans la réduction de la latence et l'optimisation de la bande passante [20], en particulier pour les applications en temps réel telles que : les transports intelligents, la santé intelligente, etc... L'équilibrage de charge est un mécanisme crucial qui permet au FC d'atteindre ses objectifs d'efficacité. Son rôle est de répartir les charges de travail de manière optimisée entre plusieurs dispositifs afin d'éviter la surcharge d'un seul élément et de garantir des performances constantes, une haute disponibilité et une utilisation efficace des ressources.

Dans la e-santé, la gestion des données patient présente un défi majeur lié au placement des applications et à l'équilibrage de charge. La problématique centrale réside dans la distribution optimale des patientes sur les nœuds fog (FN) disponibles. L'objectif est d'assurer une charge de travail équitablement répartie entre tous les nœuds, exploitant au maximum leurs ressources, tout en garantissant que les patients dont le niveau d'urgence est le plus élevé soient prioritairement traités par les FN les plus performants.

Notre visée est de minimiser la latence et d'améliorer la qualité de service, car tout cela converge vers la recherche d'une solution optimale garantissant une haute performance et une bonne scalabilité.

Notre objectif dans le cadre de ce projet de fin d'études est de développer une méthode pour l'optimisation du placement des applications et l'équilibrage de la charge dans un environnement FC, afin de garantir un traitement rapide et efficace des données. Cela doit également assurer la réactivité et la sécurité des informations sensibles, spécifiquement dans le domaine de e-santé, pour des applications telles que la télémédecine et la surveillance en temps réel des patients.

Afin de répondre à ces besoins et d'atteindre les objectifs fixés, nous avons opté pour une méthode hybride qui est Weighted Round Robin (WRR) combinée avec Least Connexion (LC). En effet, vu que WRR ne s'adapte pas dynamiquement à la charge réelle, car il suppose que les poids reflètent la capacité de traitement, nous avons ajouté LC, parce qu'il s'adapte à l'état actuel des nœuds, car il est spécialement conçu pour équilibrer la charge entre les nœuds.

Ensuite, nous avons utilisé une méthode méta-heuristique, l'algorithme bio-inspiré Ant Colony Optimisation (ACO), en raison de son efficacité à identifier des chemins optimaux dans des environnements complexes. À partir du meilleur chemin obtenu, nous effectuons un rééquilibrage de charge en le combinant avec l'algorithme LC.

Notre rapport est construit autour de trois chapitres ; les deux premiers sont théoriques et le dernier est pratique représentant notre contribution concrète à ce PFE.

Dans le chapitre 1, nous avons introduit la notion de Fog Computing : définition de l'architecture, son fonctionnement, ses avantages, la qualité de service dans le FC et son application dans le domaine de la santé (Healthcare).

Le chapitre 2 aborde la notion et les problématiques liées à la gestion des ressources dans le Fog Computing, en particulier le placement des applications. Il traite également du concept d'équilibrage de charge, en présentant les différentes techniques utilisées. Enfin, il examine l'état de l'art sur l'équilibrage de charge dans le Fog.

Le dernier chapitre est consacré à notre participation à la mise en œuvre de ce PFE, notamment les simulations faites avec iFogSim. Il présente en détail la solution proposée, les outils utilisés et l'évaluation des résultats obtenus en utilisant la méthode WRR en combinaison avec LC et l'algorithme ACO en association avec LC dans le contexte de Fog Computing.

Chapitre I

FOG COMPUTING

I. Chapitre I : Fog Computing

I.1 Introduction

L'augmentation des appareils connectés et des dispositifs intelligents crée un besoin croissant de traitement des données en temps réel. Cette tendance conduit à l'émergence du Fog Computing, qui offre des solutions innovantes pour répondre à ces enjeux, tout en ouvrant de nouvelles perspectives, notamment dans le secteur de la santé.

Dans ce chapitre nous allons introduire le Fog Computing, son architecture, sa qualité de service puis le Fog dans la santé et enfin la relation entre le Fog, la 5G et le F-RAN.

I.2 Définition

Plusieurs définitions ont été données dans la littérature à ce terme, parmi les plus utilisées :

CISCO : « Le Fog Computing étend le Cloud Computing jusqu'aux périphéries du réseau, là où les données sont générées. Il fournit un cadre pour offrir des services de calcul, de stockage et de mise en réseau plus proches de l'endroit où les données sont créées, plutôt que de dépendre uniquement du cloud. Cette approche permet de réduire la latence, d'améliorer la performance des applications et d'optimiser l'utilisation de la bande passante en traitant les données localement avant de les envoyer vers le Cloud».[1]

NIST : « Le Fog Computing est une architecture horizontale au niveau du système qui distribue les ressources et services de calcul, de stockage, de contrôle et de mise en réseau, de l'infonuagique jusqu'à la source des données. Le Fog Computing rapproche l'infonuagique de l'endroit où se trouvent les objets, permettant une plus grande proximité avec les dispositifs finaux ». [2]

IIC : « Le Fog Computing est une infrastructure de calcul décentralisée dans laquelle les services de données, de stockage et d'application sont distribués à l'endroit le plus logique entre la source des données et le cloud, généralement plus près des dispositifs finaux.» [30]

Le FC est un concept qui désigne une architecture intermédiaire entre le cloud et les appareils des utilisateurs (une extension du Cloud). Souvent assimilé à l'Edge Computing, il consiste à exécuter des tâches à la périphérie du réseau plutôt que dans le Cloud. La notion de "Edge" (périphérie) varie selon le contexte :

Dans le domaine de santé, elle peut faire référence à des dispositifs médicaux connectés, des capteurs ou des passerelles locales qui traitent les données des patients en temps réel, réduisant ainsi la latence et améliorant la réactivité des systèmes. Cela permet de répondre aux exigences critiques du secteur, comme la confidentialité des données et la nécessité de traitements rapides pour des applications telles que la surveillance continue ou les diagnostics à distance.

I.3 Historique

Le Fog Computing est un concept relativement récent qui a émergé en réponse aux défis posés par l'explosion de « Internet Of Things » (IoT) et la croissance exponentielle des données.

I.3.1 Origines et émergence (2010-2014)

Le terme "Fog Computing" a été popularisé par **Cisco** en 2012, mais l'idée fondamentale de rapprocher le traitement des données de l'edge du réseau a commencé à germer quelques années auparavant. Les chercheurs ont réalisé que le modèle traditionnel de CC, où toutes les données sont centralisées et traitées dans des centres de données distants, n'était pas toujours optimal pour les applications IoT.

I.3.2 Fondation du consortium OpenFog (2015)

En novembre 2015, un consortium industriel du nom de 'OFC- Open Fog Consortium' a été créé par ARM, Cisco, Dell, Intel, Microsoft et l'Université de Princeton. L'objectif principal de ce consortium était de développer un cadre architectural standardisé pour le Fog, afin de faciliter son adoption et son interopérabilité. Il était une étape clé pour structurer le FC.

I.3.3 Fusion avec l'Industriel Internet Consortium (2019)

En janvier 2019, l'organisation OFC a fusionné avec l'IIC, une autre organisation axée sur l'avancement de l'IoT industriel. Cette fusion a permis de rassembler les efforts des deux consortiums pour promouvoir l'adoption du FC dans divers secteurs.

I.2.3 Développements récents et tendances actuelles

Depuis sa création, le FC a connu un développement rapide, avec de nombreuses entreprises et organisations qui adoptent cette approche pour améliorer les performances de leurs applications, notamment dans le domaine de la santé. Les tendances actuelles dans ce secteur incluent :

- **Diagnostics plus rapides et précis**

L'analyse en temps réel des données de santé, grâce à des algorithmes d'IA (Intelligence Artificielle) déployés localement, permet d'alerter rapidement les médecins en cas d'anomalie.

- **Surveillance à distance efficace**

La collecte et l'analyse en temps réel des données de santé à domicile facilitent la surveillance des patients et la détection précoce de problèmes de santé.

- **Gestion sécurisée des données de santé**

Le stockage et le traitement locaux des données réduisent les risques de violation de données et garantissent la confidentialité des informations personnelles.

I.4 Architecture

L'architecture du Fog se distingue par une organisation hiérarchique en plusieurs niveaux, chacun contribuant spécifiquement au traitement et au flux des données. Elle repose sur les éléments clés suivants :

I.4.1 Appareils Intelligents (IoT)

Ce sont les appareils situés à l'extrémité du réseau qui génèrent des données, tels que les capteurs, les actionneurs et les smartphones.

I.4.2 Fog Nodes (FN)

Ce sont les appareils qui constituent la couche Fog, qui peut avoir de nombreux niveaux de FN [27]. Les couches de passerelle de Fog (FGL - Fog Gateway Layers) et les couches computationnelles de Fog (FCL - Fog Computational Layers) sont les deux sous-couches qui les constituent. Cette couche est située entre les appareils IoT et le cloud. Les FN peuvent être des routeurs, des serveurs ou tout autre appareil doté de ressources de calcul et de stockage. Ils effectuent le traitement, l'agrégation et l'analyse des données plus près de l'edge.

I.4.3 Réseau Fog (Fog Network)

C'est le réseau qui connecte les appareils en périphérie (edge) et les nœuds Fog. Il peut s'agir d'un réseau local, d'un réseau étendu (WAN) ou d'une combinaison des deux.

I.4.4 Cloud

C'est le centre de données centralisé qui fournit des ressources de calcul, de stockage et de réseau évolutives. Il est utilisé pour le stockage de données à long terme et le déploiement d'applications.

La figure (1) ci-dessous illustre le **Fog** :

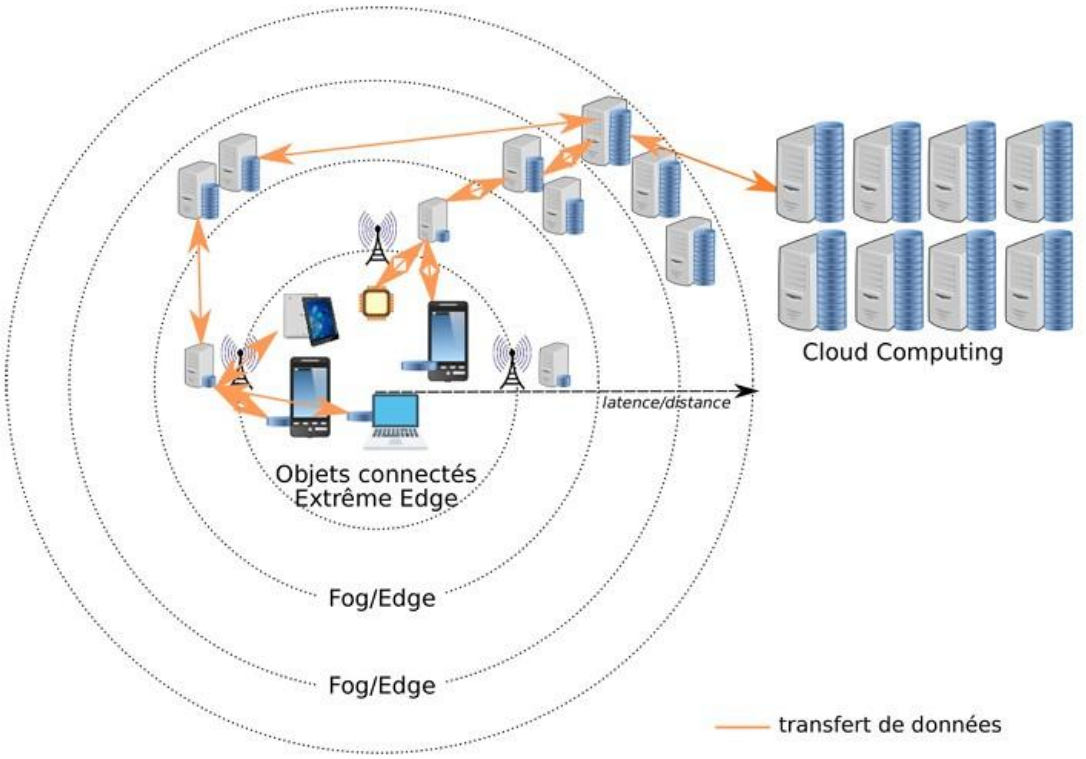

Figure I- 1: Architecture Du Fog Computing [18]

I.5 Fonctionnement

Le FC opère à travers une série d'étapes logiques essentielles pour une gestion efficiente des données et des services. Ce processus assure une réponse rapide aux besoins des utilisateurs. Voici ces étapes :

- **Collecte de données :** Les appareils IoT collectent des données de leur environnement.
- **Traitement des données :** Les données sont envoyées au nœud Fog le plus proche pour traitement. Les Nœuds Fog (FN) effectuent des tâches telles que le filtrage, l'agrégation et l'analyse des données.
- **Stockage des données** : Les FN peuvent stocker les données temporairement ou à long terme, selon les exigences de l'application.
- **Transfert de données** : Les données traitées peuvent être envoyées au CC pour une analyse ou un stockage plus approfondi.
- **Exécution d'applications :** Les applications peuvent être déployées sur des FN pour fournir des services aux appareils en Edge.

I.6 Avantages du Fog Computing

Le recours au FC présente plusieurs bénéfices qui en font une solution efficace face aux limites du Cloud. Parmi les principaux avantages, on peut citer :

- **Réduction de la latence** : Le traitement des données plus près des appreils réduit la latence, ce qui est essentiel pour les applications en temps réel.
- **Amélioration de l'utilisation de la bande passante :** Le FC réduit la quantité de données qui doivent être envoyées au Cloud, ce qui améliore l'utilisation.
- **Sécurité renforcée :** Il peut améliorer la sécurité en traitant les données localement et en réduisant le risque de violations de données.
- **Réduction des coûts** : En réduisant la quantité de données envoyées au Cloud, le Fog peut contribuer à réduire les coûts du CC.

I.7 Inconvénients du Fog Computing

- **Complexité de gestion :** La distribution des nœuds sur un vaste réseau augmente la complexité du déploiement, de la configuration et de la maintenance.
- **Défis de sécurité :** Un plus grand nombre de points de traitement en périphérie accroît la surface d'attaque potentielle et rend la sécurité plus difficile à garantir.
- **Gestion de la charge de travail :** L'optimisation de la répartition des tâches et de l'analyse entre des FN aux capacités et emplacements variés est un défi, pouvant entraîner des déséquilibres dans l'utilisation des ressources.
- **Coûts d'infrastructure initiaux :** L'acquisition et l'installation des nombreux FN et équipements réseau représentent un investissement initial souvent élevé.

I.8 Qualité de service dans le FC

La Qualité de Service **(**QoS - Quality of Service**)** dans le cadre du FC englobe l'ensemble des critères et des mécanismes déployés pour assurer que les applications hébergées au sein de ce système répondent précisément aux exigences de performance.

Dans le FC, la QoS elle impacte la gestion des ressources, la latence, la BP et la consommation d'énergie dans des environnements distribués[3].

Bien que la QoS admette plusieurs définitions, dans notre cas elle repose sur :

I.8.1 La latence

L'un des indicateurs les plus critiques pour les systèmes Fog, notamment pour les applications sensibles au temps, telles que la réalité augmentée, les systèmes de santé intelligents, etc...

La latence dans un environnement FC est influencée par plusieurs facteurs, dont la distance entre le périphérique source et le nœud de calcul, ainsi que la congestion du réseau. La latence peut être formalisée comme la somme de la latence de propagation (temps de transmission des données entre les nœuds) et de la latence de traitement (temps nécessaire pour traiter les données sur les nœuds de périphérie).

La gestion de la latence se fait souvent par l'optimisation de la sélection des nœuds pour minimiser les distances et par l'intelligence distribuée, qui permet de faire des décisions en temps réel sur l'endroit où traiter les données pour réduire cette latence.

I.8.2 La bande passante

Elle est souvent limitée et une mauvaise gestion peut entraîner des goulets d'étranglement et affecter la performance globale.

On utilise des algorithmes d'optimisation pour allouer la BP de manière efficace, souvent en utilisant des stratégies comme la compression de données, la priorisation du trafic (QSD - Quality of Service Differentiation en anglais), ou encore l'agrégation de données avant leur envoi vers le Cloud.

Des techniques comme la programmation linéaire, l'optimisation de la congestion du réseau et l'élagage de données inutiles permettent de maximiser l'utilisation de la bande passante tout en respectant les contraintes de QoS.

I.8.3 La consommation d'énergie

Un problème clé car de nombreux appareils IoT sont utilisés dans des environnements à ressources limitées.

La consommation d'énergie (EC) peut être modélisée en tenant compte des caractéristiques des nœuds de périphérie. Les algorithmes de gestion de l'énergie cherchent à minimiser la consommation en utilisant des stratégies telles que :

- Le mode veille dynamique des nœuds.
- La répartition dynamique des tâches pour optimiser les périodes d'activité des nœuds, en prenant en compte leurs états de consommation.

Les techniques d'optimisation telles que la programmation par contraintes ou les algorithmes génétiques sont utilisées pour attribuer les tâches aux nœuds de manière à minimiser la consommation d'énergie tout en garantissant la performance.

I.8.4 La gestion de la sécurité

Le contrôle nécessite une attention particulière, surtout lorsqu'il s'agit de données sensibles, telles que celles utilisées dans les domaines de la santé ou de la finance.

Le chiffrement des données à chaque étape de leur traitement est essentiel pour garantir leur confidentialité. Cela comprend le chiffrement des données lorsqu'elles sont stockées et lorsqu'elles sont envoyées entre les nœuds.

L'authentification et la gestion des permissions d'accès sont mises en œuvre pour s'assurer que seules les entités autorisées peuvent accéder aux données et services sensibles.

I.8.5 L'optimisation de l'orchestration des services et des tâches

C’est crucial dans l’environnement du FC. Un bon placement garantit une faible latence et une consommation énergétique optimale.

Les algorithmes de placement de services utilisent des techniques comme le DL (Deep Learning en anglais), les algorithmes d'optimisation combinatoire pour prendre en compte des facteurs multiples (latence, charge des nœuds, coût énergétique, etc.) dans le placement dynamique des applications sur les FN.

Une évaluation composite peut combiner les mesures de latence, de consommation d'énergie, et de bande passante pour obtenir la QoS globale, souvent en utilisant des fonctions d'agrégation ou des modèles d'optimisation multi-objectifs.

Équation I-1: La qualité de service globale

$$QoS\ globale = w1 * latence + w2 * bande\ passante + w3 * consommation\ d'energie$$

Où w1, w2, w3 sont les poids associés à chaque critère en fonction de l'importance relative.

I.9 La vision du Fog Computing dans le domaine de santé

Le secteur de la santé dans la plupart des pays fait face à plusieurs obstacles en raison de la longévité (vieillissement de la population), des maladies chroniques et du manque de personnel infirmier. En même temps, il existe une demande pour la réduction des frais en maintenant une bonne qualité de soins pour les patients. En conséquence, l'industrie de la santé développe un modèle de soins à distance basé sur des informations visant à améliorer la qualité, l'accessibilité numérique, l'efficacité et la surveillance des patients, tout en réduisant les coûts.

Aujourd'hui, une grande partie du temps est gaspillée dans les hôpitaux par la mesure manuelle des paramètres biométriques et le transfert des données entre les systèmes, impliquant souvent papier et stylo. La surveillance à distance libérera du temps pour les soignants.

Les bénéfices de la surveillance à distance comparée au suivi à l'hôpital améliorent les processus hospitaliers. Plusieurs processus sont planifiés manuellement, au lieu d'utiliser des capteurs qui simplifient l'obtention des informations correctes sur l'état actuel et l'emplacement de l'équipement, des infirmiers et des patients.

De plus, les capteurs fournissent une image précise des patients et captent continuellement les données, permettant ainsi d'obtenir des paramètres biométriques internes. Cela révolutionnera les diagnostics et les traitements, Topol appelle cela "la numérisation des humains"[25]. Une fois ce nouveau tableau des patients combiné avec des techniques analytiques, de nouvelles découvertes transformeront les détections précoces, les diagnostics, les médicaments et le traitement des maladies.

Une autre tendance est l'abandon du traitement réactif, où les patients sont traités à l'hôpital seulement après un incident, au profit d'une médecine plus préventive. Cela commence par la surveillance des individus avant le niveau d'urgence, afin de les maintenir hors de l'hôpital le plus longtemps possible. De plus, en augmentant les possibilités de surveiller les patients à domicile, cela facilite leur sortie anticipée de l'hôpital. En général, cela signifie que les

frontières entre l'hôpital, le domicile et les autres points de soins deviennent de plus en plus floues : les soins de santé se déroulent de manière continue et partout.

La cohérence apparente entre les défis du secteur de la santé, les exigences qui en résultent et les bénéfices du Fog computing suggère un potentiel certain pour cette technologie comme moteur de l'informatique omniprésente (UC - Ubiquitous Computing en anglais) dans le domaine de la santé :

I.9.1 Flexibilité du lieu de calcul

Le FC peut offrir les ressources informatiques nécessaires au sein du réseau pour répondre aux exigences réglementaires et techniques. Cette approche est non seulement importante pour disposer de ressources informatiques entre les capteurs et le Cloud, mais aussi pour les gérer de manière optimale. Cela inclut la transparence de l'exécution pour les applications, ainsi que la flexibilité concernant l'endroit où le calcul peut être effectué. L'emplacement peut être dynamique et dépendre du contexte actuel, des exigences de l'environnement et des besoins de l'application.

I.9.2 Intégration

Dans la situation actuelle, l'introduction de nouveaux dispositifs nécessite souvent l'ajout simultané d'une infrastructure de support pour leur bon fonctionnement. Par exemple, le système de surveillance du rythme cardiaque, qui requiert une infrastructure dédiée pouvant inclure des réseaux pour transmettre les données et des logiciels pour analyser et gérer ces informations. Cela représente une charge considérable. Dans une architecture de FC, de nouveaux capteurs peuvent être ajoutés à l'infrastructure existante. Le Fog peut également servir de couche de compatibilité entre différents standards.

I.9.3 Mobilité du patient

Les infrastructures spécifiques aux applications limitent la zone de surveillance des patients, surtout lorsqu'ils sont sur le point de quitter l'infrastructure hautement équipée d'un hôpital. Il est souvent difficile de couvrir cette transition, ce qui peut prolonger le séjour des patients. Avec les ressources de Fog, les transitions entre différents environnements peuvent être gérées de manière plus progressive.

I.9.4 Nouvelles applications

Le FC permettra également de développer des applications entièrement nouvelles. En ajoutant des niveaux plus élevés d'autonomie et d'intelligence à la périphérie, le Fog améliorera la latence, le temps de réponse, et permettra des économies d'énergie pour les dispositifs portables à faible coût, tout en accomplissant des tâches complexes telles que la détection de chutes. La révolution du secteur de la santé remplacera les dispositifs coûteux et complexes, sans recourir à des algorithmes simples à la précision limitée. Ces appareils seront rendus possibles grâce au FC, conduisant ainsi à l'Internet des objets de santé (IoMT : Internet of Medical Things en anglais).

I.10 La convergence du Fog Computing, 5G et le réseau FRAN

La **5G**, le **FC** et le **F-RAN** sont interconnectés de manière complémentaire pour améliorer la performance des réseaux et des applications, en particulier dans des différents secteurs.

La 5G (Cinquième Génération) est apparue en 2019, bien que sa mise en œuvre ait varié selon les pays et les régions. Elle offre des vitesses de connexion beaucoup plus rapides, une latence très faible et une capacité accrue pour connecter un grand nombre de dispositifs simultanément. Elle permet des applications avancées comme les véhicules autonomes, la réalité augmentée, et l'Internet des objets médicaux (IoMT), en améliorant les performances par rapport aux technologies précédentes comme la 4G.

La cinquième génération dans le FC fait référence à l'intégration des technologies 5G avec les infrastructures de Fog pour offrir une faible latence, une BP élevée et une connectivité ultra-fiable. Dans ce contexte, des ressources de calcul et de stockage sont déployées plus près des utilisateurs finaux, ce qui améliore la performance des applications en temps réel tout en réduisant la charge sur le réseau central.

Le C-RAN (Cloud Radio Access Network en anglais) est une architecture réseau qui décentralise les fonctions de traitement des stations de base traditionnelles en les transférant dans des centres de calcul centralisés, souvent dans le Cloud. Cette approche permet d'optimiser l'utilisation des ressources et d'améliorer la flexibilité, la gestion des interférences et l'efficacité énergétique dans les réseaux mobiles. Ainsi, malgré la centralisation de certaines fonctions, le C-RAN s'inscrit dans la logique du Fog Computing en rapprochant une partie du traitement des données des utilisateurs, contribuant ainsi à l'informatique omniprésente.

Le FC, étant une extension du CC, est lié au C-RAN, car il peut apporter des avantages supplémentaires, notamment pour les applications nécessitant un traitement en temps réel à faible latence. Cela est particulièrement pertinent dans des domaines comme la santé, où des applications critiques telles que la télémédecine, la surveillance des patients en temps réel, ou encore les dispositifs médicaux connectés (IoMT) exigent une réactivité immédiate. L'intégration du FC avec le C-RAN permet de traiter les données localement, réduisant ainsi la latence et garantissant des réponses rapides, essentielles pour la prise en charge des patients et la gestion des urgences.

En outre, une évolution du C-RAN, appelée F-RAN (Fog Radio Access Network en anglais), intègre le FC directement dans l'architecture du réseau radio. Ce modèle distribue le traitement des données au plus près des utilisateurs finaux, par exemple dans des unités de traitement situées au sein même des établissements de santé ou sur des dispositifs mobiles. Dans le domaine de la santé, cela permet de traiter et analyser en temps réel des données critiques générées par des dispositifs médicaux, tels que des capteurs biométriques, des appareils de surveillance cardiaque, des moniteurs de signes vitaux et des dispositifs IoMT. Le F-RAN joue un rôle clé dans le traitement local de ces informations, réduisant ainsi la latence et évitant la congestion du réseau central. Par exemple, lorsqu'un patient présente un signe d'alerte dans un dispositif connecté, Il permet une prise de décision immédiate sans avoir besoin de transférer les données vers un centre de données éloigné. Ainsi, permet d’assurer une communication fluide et fiable entre les dispositifs médicaux locaux et les centres de données distants, garantissant que même les interventions complexes, telles que la télémédecine ou la chirurgie assistée par des robots, puissent être réalisées sans délai significatif. Ce traitement local des données optimise également l'utilisation des ressources, en allégeant la charge du réseau central et en permettant une gestion dynamique des ressources en fonction des besoins en temps réel. Cette technologie dans le futur se présentera comme un levier majeur pour les innovations en matière de soins de santé à distance.

Donc, la 5G fournit la connectivité, le FC permet un traitement local, et le F-RAN optimise le réseau pour des applications en temps réel, comme la télémédecine ou la gestion des données de santé. Ensemble, ces technologies permettent des solutions plus rapides, plus efficaces et adaptées aux besoins de l'Internet des objets et des systèmes critiques.

I.11 Conclusion

Dans cette partie, nous avons pu développer le vaste sujet du FC et nous avons pu identifier les challenges du Fog aux quels la communauté scientifique doit faire face.

Le FC représente une avancée significative vers l'infrastructure informatique dématérialisée. Il fournit des ressources informatiques, logicielles ou matérielles, accessibles à distance en tant qu'architecture de calcul décentralisée. L'adoption de ce modèle soulève un certain nombre de défis, notamment au sujet de l'équilibrage de charge. FC a été introduit pour répondre à un besoin de traitement avec une faible latence que les infrastructures de CC ne sont pas capables de fournir, mais également pour une question de passage à l'échelle.

Dans le prochain chapitre (2), nous introduirons la notion de gestion de ressource ainsi que l'équilibrage de charge dans le cas général et dans le cas des environnements Fog.

Chapitre II

Gestion de ressource et équilibrage de charge

II. Chapitre II : Gestion de Ressource et équilibrage de charge

II.1 Introduction

Le FC est une sorte de plate-forme offrant des capacités locales de traitement et de stockage des données pour l'exécution de services IoT ou IoMT, plutôt que de les transférer vers le Cloud. Ainsi, il fournit des ressources de stockage, de calcul et de mise en réseau. En raison de la dynamique, de l'hétérogénéité et de l'incertitude de l'environnement Fog, un mécanisme de gestion des ressources est essentiel pour assurer son déploiement efficace.

Dans ce chapitre, nous allons introduire le concept de gestion des ressources, y compris le placement des applications. Ensuite, nous présenterons l'équilibrage de charge et ses différentes techniques. Enfin, nous approfondirons l'étude de l'équilibrage de charge, en nous focalisant plus particulièrement sur son application dans le FC.

II.2 Gestion des ressources

La gestion des ressources en FC englobe plusieurs problématiques sous différents aspects, avec des contraintes variées, qui ne peuvent être entièrement considérées comme un problème unique. Ces approches sont classées en plusieurs catégories, notamment :

II.2.1 Problème de Planification des Ressources

Dans l'environnement du Fog, les demandes de service des dispositifs IoT peuvent être traitées par plusieurs nœuds. Chaque demande de service peut être divisée en un ensemble de tâches. L'enjeu de planification des ressources dans le FC consiste à attribuer de manière optimale différentes tâches soumises aux FN situés en périphérie du réseau, afin d'avoir une qualité de service convenues avec un utilisateur d'un dispositif IoT tout en minimisant le temps d'exécution des tâches soumises.

Ce problème est considéré comme un problème d'optimisation NP-difficile (Non-déterministique Polynomial-time hard en anglais) qui désigne une classe de problèmes en informatique théorique, généralement résolu par des algorithmes méta-heuristiques qui sont des approches intelligentes et flexibles permettant de trouver des solutions faisables et quasi-optimales en temps linéaire.

Dans les approches de planification des tâches, on distingue trois grandes catégories [34, 35] :

- **Approches statiques**

Les tâches arrivent simultanément aux FN et les décisions sont prises avant leur soumission. Cela signifie que la planification peut être effectuée en disposant de toutes les informations nécessaires sur les demandes reçues et les ressources disponibles avant l'ordonnancement. Toutefois, dans des systèmes plus hétérogènes comme le FC, il est parfois impossible de disposer de toutes ces informations à l'avance. Par conséquent, ces approches ne garantissent pas toujours un mécanisme de planification optimale dans le Fog.

- **Approches dynamiques**

Contrairement aux approches statiques, les temps d'arrivée des tâches ne sont pas connus avant leur envoi et elles sont planifiées dès leur arrivée dans le système. Ces approches permettent donc de gérer des requêtes aléatoirement aux arrivées imprévisibles et de les planifier en temps réel.

- **Approches hybrides**

Ces approches combinent différents critères comme le coût (CE - Cost Efficiency en anglais) et le temps d'exécution, pour couvrir plusieurs types d'applications comme les flux de travail et les traitements en IoMT. Cette caractéristique est particulièrement importante pour l'ordonnancement dans un environnement FC, car il n'existe pas de solution unique convenant à tous les types de demandes.

II.2.2 Problème de déchargement des Tâches

La technique d'externalisation des tâches permet de gérer les contraintes de ressources telles que la puissance de calcul, l'énergie de la batterie et l'espace de stockage sur les appareils IoT, en particulier pour les traitements nécessitant des calculs complexes. Ces derniers, étant généralement limités en ressources, il devient nécessaire d'externaliser certaines tâches vers le FC afin d'améliorer leurs performances.

Pour prendre en charge de nombreuses applications IoT gourmandes en ressources, comme la réalité augmentée (AR - Augmented Reality en anglais), la réalité virtuelle (VR -

Virtual Reality en anglais) et l'IoMT, une entité externe exécute les opérations à la place de ces appareils et leur retourne les résultats. Ce processus est appelé Task Offloading (TO).

Le processus de TO des appareils mobiles vers les nœuds de FC est divisé en trois composants principaux :

- **Les appareils mobiles**

Ils sont responsables de la manière dont une application IoT est partitionnée en différentes tâches et décident si celles-ci doivent être exécutées localement ou out-sourcées via des mécanismes d'infogérance.

- **Les liens de communication**

Leur qualité (WiFi, réseaux cellulaires, etc..) dépend de la BP et de la mobilité des appareils. Ces liens sont utilisés pour transférer les calculs importants des appareils mobiles vers les FN.

- **Les nœuds de Fog**

Ils possèdent des capacités de calcul plus élevées que les appareils mobiles. Ils équilibrent la charge de travail d'une application IoT en temps réel pour améliorer les performances du déchargement et de maximiser le débit du système.

L'externalisation des tâches peut être réalisée pour diverses raisons, notamment l'équilibrage de charge, la gestion de la latence, la sécurité et l'efficacité énergétique, etc...

Les approches d'externalisation des tâches, selon le nombre de destinations d'externalisation, peuvent être classées en deux catégories principales :

- **Approches d'externalisation unique** (Single-Type en anglais) : Les opérations sont déchargées vers un seul nœud de Fog pour un traitement séquentiel.
- **Approches d'externalisation multiple** (Multiple-Type en anglais) : Les opérations sont déchargées vers plusieurs FN pour un traitement parallèle, garantissant ainsi le respect des exigences de QoS.

Pour les applications IoMT nécessitant un traitement séquentiel où la communication entre les différentes parties d'une tâche est fréquente. En revanche, les approches d'externalisation multiple sont mieux adaptées aux applications reposant sur des traitements répétitifs et parallélisés.

II.2.3 Problème de provisionnement des ressources

A cause de la charge de travail des services IoT ou fil des années, ils vont expérimenter des fluctuations de la charge de travail. Ces variations de la charge de travail peuvent aboutir dans les problèmes de super provisionnent ou sous provisionnement.

Dans le problème de sur-provisionnement, les ressources allouées pour les services IoT sont supérieurs à la charge réelle de la demande de l'utilisateur. C'est juste correct selon le point de vue d'accord de niveau de service (SLA - Service Level Agrement en anglais). Mais l'utilisateur de l'IoT subit un coût inutile si les FN sont inactifs ou faiblement chargés.

D'une autre part, les ressources allouées dans les services IoT ne parviennent pas à satisfaire la charge et la demande réelle de l'utilisateur à cause des violations du SLA qui entraînent une perte des utilisateurs de IoT. Par conséquent, il est important de fournir dynamiquement le nombre approprie des FN adéquat afin de gérer la charge de travail soumise par IoT pour minimiser le coût du système en respectant les contraintes de QoS.

La gestion dynamique des ressources constitue une approche efficace pour gérer les fluctuations et la capacité à s'étendre horizontalement dans les FN selon la charge de travail des dispositifs IoT entrants [21] .

Les approches de provisionnement de ressources doivent être élastiquement adaptées aux variations de charge des services IoT en temps réel et prévient le gaspillage des FN grâce à des actions de mise à l'échelle lorsque la charge augmente afin de répondre aux exigences de qualité de service et l'action de réduction d'échelle.

II.2.4 Problème d'allocation des ressources

Ce problème est différent dans les environnements informatiques basés sur le CC et le FC. Il faut allouer efficacement un ensemble de nœuds variées géographiquement répartis pour des services IoT concurrents avec différentes exigences de la QoS, en tenant compte de la priorité et l'équité du service. Dans les réseaux Fog Computing, il y a plusieurs entités distribuées y compris le Cloud, les FN et les appareils IoT ce qui rend le problème d'allocation des ressources plus complexe [22].

Ce problème est considéré comme un problème NP, il se manifeste sous un obstacle de double appariement, afin que les serveurs Cloud et les nœuds fog soient associés aux utilisateurs IoT. Tandis que les FN et les utilisateurs IoT soient associés aux serveurs Cloud.

L'objectif de l'allocation des ressources est de trouver des paires optimales serveur cloud - nœud fog - utilisateur pour l'ensemble des utilisateurs IoT dans le but d'atteindre plusieurs objectifs, tel que : maximiser la QoS et minimiser le coût global autant que possible.

II.2.5 Problème de placement des applications

C'est un processus clé pour optimiser les performances et garantir une utilisation efficace des ressources.

Le placement commence par une analyse approfondie des exigences spécifiques de chaque application. Par exemple, les applications sensibles au temps, telles que celles utilisées dans la santé, nécessitent un traitement des données localisé pour minimiser la latence.

De plus, les données sensibles doivent être placées sur des nœuds sécurisés pour garantir leur confidentialité.

Ensuite, il faut sélectionner le nœud Fog adapté pour chaque application en fonction de ses besoins spécifiques en calcul, stockage et mise en réseau. Les FN, varient en termes de puissance de traitement et de capacité de stockage. Il est donc important de choisir un nœud capable de répondre aux exigences de l'application.

Il doit être dynamique, permettant de réajuster leur position en fonction de la charge du système et de la disponibilité des ressources, afin d'éviter des goulots d'étranglement.

Le placement des applications dans le FC n'est pas statique. Il peut être ajusté en fonction de l'évolution des conditions du réseau et de la charge des nœuds. Cette adaptabilité permet de garantir que les applications continuent de fonctionner de manière optimale, même en cas de fluctuations des ressources disponibles.

Par exemple, soit A une application IoMT orientée services avec Q un ensemble d'exigences QoS et F un ensemble de FN. Une solution au problème de placement des services consiste à mapper chaque service de A à un ou plusieurs nœuds de Fog dans F, de manière à satisfaire toutes les exigences de Q et à optimiser un ensemble de fonctions objectives O, utilisées pour évaluer la qualité de la solution. La réponse peut consister en un service placé sur un ou plusieurs FN, et un nœud de Fog peut héberger plusieurs services.

Le placement des applications base sur la gestion du courtier (BM - Broker Management en anglais) est divisé en trois sous-catégories :

- **Centralisé**

Centralized Broker (CB) a besoin d'informations provenant de toutes les entités du Fog (les appareils, les clients et les services IoT) pour prendre des décisions d'optimisation globale. Cependant, il souffre de problèmes tels que la surcharge réseau avec l'augmentation du nombre du FN,la tolérance aux pannes à cause d'un point de défaillance unique et un manque d'évolutivité en fonction de BP et de leur capacité de traitement.

- **Décentralisé**

Plusieurs courtiers locaux prennent des décisions optimisées localement, ce qui les rend plus évolutifs. Ces approches permettent d'ajouter plusieurs noyaux de gestion sans dépendre d'un seul noyau. Cependant, elles souffrent d'une surcharge de communication entre ces noyaux. Malgré cela, elles restent plus performantes que les approches centralisées, notamment en termes de réduction la surcharge réseau et du temps d'exécution global qui est plus élevé.

- **Hiérarchique**

Cette approche combine les avantages des solutions centralisées et décentralisées en utilisant plusieurs gestionnaires semi-globaux et locaux qui travaillent ensemble pour optimiser la gestion des ressources.

Dans ce cadre, plusieurs approches ont été proposées pour optimiser le placement des applications, l'ordonnancement des tâches et l'allocation des ressources. Parmi ces approches, certaines ont donné des résultats intéressants :

Le Tableau (II-1) présente une comparaison des différentes études en fonction de leur domaine, technique utilisée, outil d'évaluation, ainsi que les avantages et les inconvénients de chaque approche de placement d'application discutée et examinée.

Table II-1: Comparaison des différentes études

Auteurs	Domaine	Outil d'évaluation	Approche	Avantages	Faiblesses
Mahmud, Ramamohanarao et Buyya [4]	Maison Intelligente	Le simulateur iFogSim	Basée sur une Heuristique	-Convient aux apps IoT sensibles à la latence -Réduction du temps de déploiement	-La politique proposée n'a pas été évaluée dans une étude de cas Réelle -La mobilité des utilisateurs n'a pas été pris en compte
Mahmoud, Rodrigues, Saleem, AlMuhtadi, Kumar et Korotaev [5]	Applications de santé	Le simulateur iFogSim	Basée sur une Heuristique	-Faible consommation d'énergie -Faible latence	-La mobilité des patients n'a pas été prise en compte -Complexité computationnelle élevée
Taneja et Davy [6]	Général	Le simulateur iFogSim	Basée sur une Heuristique	-Faible complexité computationnelle. -Réduction de la latence, de la consommation d'énergie et de l'utilisation du réseau.	- L'algorithme proposé n'a pas été évalué

II.3 Équilibrage de charge

L'équilibrage de charge (LB - Load Balancing en anglais) est un problème fondamental de la gestion des ressources dans le FC. Il est une technique qui permet de répartit la charge de travail entre plusieurs ressources disponibles, telles que des serveurs ou des processeurs, afin d'éviter la surcharge du système. Cette approche est particulièrement utilisée dans les centres de données, les réseaux informatiques et les systèmes distribués.

LB repose sur le principe de la répartition intelligente du trafic pour maximiser l'utilisation des ressources et garantir une performance optimale.

II.3.1 Les types d'Équilibrage de Charge

L'équilibrage de charge peut être classé selon plusieurs critères, notamment la nature de la distribution et l'emplacement du répartiteur de charge.

- **Équilibrage statique**

Il répartit les tâches en suivant des règles fixes, sans prendre en compte les conditions actuelles du système. C'est une méthode simple, mais elle manque de flexibilité et peut entraîner des déséquilibres lorsque les conditions changent. Certains serveurs peuvent être surchargés tandis que d'autres sont sous-utilisés, ce qui affecte la performance du système.

- **Équilibrage dynamique**

Il ajuste en temps réel la répartition des tâches selon l'état actuel des ressources. Contrairement à l'approche statique, il optimise l'utilisation des ressources en fonction de la demande, ce qui améliore la performance et la réactivité du système face aux variations de charge.

- **Équilibrage au niveau du réseau**

Il est utilisé dans les infrastructures réseau, il répartit le trafic entre plusieurs serveurs à l'aide d'équipements spécialisés, tels que des routeurs ou des commutateurs pour éviter que certaines machines ne soient surchargées. Cela permet de maintenir des performances optimales et de réduire les risques de congestion du réseau.

- **Équilibrage au niveau applicatif**

Implémenté dans les applications logicielles, il permet de répartir les requêtes des utilisateurs sur différents services selon des critères définis. Il est utilisé pour s'assurer que

chaque demande est dirigée vers l'instance la plus appropriée, en fonction de la charge du serveur. Ce type aide à maintenir la disponibilité de l'application et à éviter de surcharger un seul serveur.

II.3.2 Les techniques d'équilibrage de charge

- **Algorithme du tourniquet (Round-robin)**

Chaque demande est envoyée à un serveur suivant un ordre fixe et régulier (circulaire), sans tenir compte de la charge de chaque serveur. C'est une technique simple, mais elle ne prend pas en compte la charge actuelle des serveurs.

- **Tourniquet pondéré (WRR - Weighted Round-Robin)**

Similaire à la méthode « Round Robin » de la répétition circulaire, où chaque serveur a un "poids" (plus grand pour les serveurs plus puissants). Les requêtes sont réparties en fonction de ce poids. Les serveurs plus puissants ou moins sollicités peuvent recevoir plus de requêtes. Il est Utilisée pour des systèmes avec des serveurs ayant des capacités différentes.

- **Le moins de connexion (LC – Least Connections)**

L'équilibrage est effectué en envoyant les requêtes au serveur qui a le moins de connexions actives à ce moment-là. Il vérifie les connexions actives de chaque serveur, identifie ceux ayant le maximum et le minimum nombre de connexions, puis fait le rééquilibrage entre eux en prenant une tache du serveur le plus chargé et la donner au serveur le moins sollicité. Cette opération est répétée jusqu'à ce que tous les serveurs soient équilibrés. Cette technique aide à répartir la charge en temps réel plus équitablement selon l'activité actuelle de chaque serveur.

- **Moins de temps de réponse (LRT – Least Response Time)**

Les requêtes sont envoyées au serveur avec le meilleur temps de réponse (le plus faible) ce qui améliore la réactivité du système. Cela permet d'optimiser la latence et d'améliorer la performance.

- **Hachage cohérent (CH – Consistent Hashing)**

Cette technique est utilisée surtout pour les systèmes distribués et les systèmes avec des ajouts ou suppressions fréquentes de serveurs. Elle consiste à affecter les serveurs en fonction de la clé de hachage d'une requête, ce qui permet de maintenir une certaine constance dans la répartition de la charge lors de l'ajout ou de la suppression de serveurs.

❖ **Les avantages des techniques :**

- Amélioration des performances en répartissant les demandes sur plusieurs serveurs.

- Haute disponibilité, en cas de panne d'un serveur, les autres prennent le relais.
- Scalabilité, possibilité d'ajouter des serveurs pour gérer plus de trafic.
- Optimisation des ressources, utilisation plus efficace des serveurs.
- Réduction des coûts en optimisant l'infrastructure existante.

❖ **Les inconvénients :**

- Complexité de gestion, nécessite une maintenance soignée.
- Point de défaillance unique (SPOF - Single Point Of Failure en anglais) .
- Latence supplémentaire causée par le traitement des requêtes.
- Coûts supplémentaires pour à l'infrastructure et à la gestion des équilibreurs de charge.
- Difficulté à gérer la session utilisateur, notamment pour les applications nécessitant des données d'état.

II.4 État de l'art sur l'équilibrage de charge dans le Fog Computing

L'équilibrage de charge est un enjeu majeur dans la gestion des ressources afin de répartir la charge de travail entrante entre les FN disponibles, en particulier pour les applications sensibles à la latence. Pour équilibrer la charge sur des nœuds de FC ayant des capacités différentes, les tâches sont distribuées selon une stratégie spécifique tout en minimisant le temps de réponse et en maximisant le débit.

Étant donné que les FN sont situés à proximité des dispositifs IoMT, la plupart des mécanismes de LB permettent un accès rapide aux ressources avec un temps de réponse réduit. Ils facilitent aussi l'évolutivité et la tolérance aux pannes.

De plus, le FC est souvent déployé dans des environnements distribués géographiquement, où les coûts de communication entre les nœuds peuvent être significatifs. Cependant, ces mécanismes rencontrent plusieurs défis, comme la latence élevée, l'attente

des tâches de faible priorité dans la file d'attente, La dégradation des performances peut survenir à cause des déplacements fréquents des processus, et il n'y a pas de normes claires pour décrire tous les différents scénarios possibles. Cela rend difficile l'évaluation comparative des différentes solutions d'équilibrage de charge, d'où un besoin urgent de solutions plus standardisées et de mécanismes de gestion de la charge capables de gérer la complexité et la dynamique du FC.

Plusieurs travaux ont été réalisés pour aborder les défis liés à l'équilibrage de charge dans les environnements de Fog, chacun proposant des solutions spécifiques visant à améliorer la gestion des ressources et la performance des systèmes distribués.

Dans [7] , les auteurs ont présenté un cadre de gestion de charge et de ressources basé sur le contrôle d'accès basé sur les rôles (RBAC - Role-Based Access Control en anglais) pour les réseaux IoT-edge-fog appelé RBAC-LBRM (Load Balancing Assisted Resource Management et Gestion des ressources assistée par l'équilibrage de charge en français). Ce cadre vise à résoudre les déséquilibres de charge du réseau et les accès non autorisés, ce qui peut nuire aux performances. En déchargeant les tâches vers les couches edge et fog, RBAC-LBRM améliore l'utilisation du CPU, de la mémoire, du délai et du jitter. Bien qu'il améliore l'utilisation des ressources et le contrôle d'accès dans des environnements IoMT dynamiques, il manque d'adaptabilité en temps réel dans l'allocation des tâches, ce qui est essentiel pour des applications comme la santé. Il ne fournit pas de stratégie complète pour optimiser à la fois la latence et les performances du réseau, se concentrant principalement sur l'équilibrage de la charge et le contrôle d'accès.

Ainsi, dans[8], une stratégie a été introduite pour le placement de modules de validation de passerelle (GVMP - Global Virtual Module Placement en anglais) pour aborder le placement des modules d'application dans les environnements IoT-fog-cloud. Cette stratégie est conçue pour réduire la latence moyenne des applications et l'utilisation du réseau en plaçant stratégiquement les modules sur des dispositifs fog hétérogènes près du bord du réseau. Les auteurs soutiennent que se fier uniquement aux serveurs basés sur le cloud dans les applications IoT peut entraîner une latence élevée et une congestion du réseau, notamment dans les scénarios de ville intelligente où les volumes de données sont importants. En validant les capacités des passerelles, cette approche sélectionne efficacement les nœuds Fog optimaux pour l'exécution des tâches, obtenant des améliorations notables par rapport à la stratégie Edgeward Module Placement (EWMP - Efficient Virtual Module Placement en

anglais). Elle présente des avantages clairs dans l'optimisation de la latence et de l'utilisation du réseau. Cependant, son approche ne traite pas explicitement des besoins d'allocation adaptative dans des environnements hautement dynamiques. Les réseaux IoT font souvent face à des demandes fluctuantes et à des changements de charge de travail en temps réel, et l'absence d'un mécanisme d'ajustement dynamique peut limiter sa réactivité dans des déploiements à grande échelle. De plus, il convient de souligner que cette technique n'est comparée qu'à la stratégie EWMP ici. Les performances relatives de cette approche par rapport à d'autres algorithmes d'équilibrage de charge adaptatif et multi-niveaux ne sont pas explorées. Cela limite une compréhension plus large de son efficacité dans des applications plus complexes et sensibles à la latence.

Ensuite, dans [9], les auteurs ont introduit HealthFog, un système intelligent basé sur le fog pour diagnostiquer les maladies cardiaques. Leur solution intègre des logiciels et du matériel pour garantir une transmission de données rapide et fiable. Une autre étude de [10] a présenté un système de surveillance des patients basé sur le fog pour la vie assistée ambiante (FAAL - Fog-based Ambient Assisted Living en anglais), un cadre basé sur le fog pour surveiller les patients atteints de maladies neurologiques chroniques. Ce système utilise un algorithme de regroupement dans la couche fog pour réduire la charge de traitement des nœuds fog individuels. Ensuite, ils ont présenté une architecture à trois niveaux pour l'Internet of Health Things (IoHT) qui prend en compte la mobilité des utilisateurs. Leur système utilise des nœuds capteurs pour la collecte de données, des FN pour la gestion des paramètres de santé et un serveur Cloud pour le traitement en cas d'urgence.

Bien que les études référencées aient évalué des systèmes de surveillance de santé basés sur le FC par rapport à des solutions basées sur le CC. Cependant, l'algorithme OLB proposé adopte une méthodologie différente. Il compare le système activé par le Fog avec des services basés sur la localisation et deux autres implémentations basées sur le FC : Algorithme de placement des Nœuds Fog (FNPA - Fog Nodes Placement Algorithm) et l'algorithme d'Équilibrage de charge sensible à la charge (LAB - Load-Aware Balancing).

L'article [11] propose un système de santé électronique pour surveiller la santé des personnes âgées basé sur l'IoT et le FC. Le système utilise la plateforme Mysignals HW V2 et une application Android pour collecter périodiquement des paramètres physiologiques et de santé générale, permettant aux personnes âgées et à leurs familles de suivre l'état de santé et de communiquer avec les prestataires de soins de santé. Les résultats de l'évaluation

indiquent que les utilisateurs trouvent le système utile, facile à utiliser et facile à apprendre, suggérant qu'il peut améliorer la qualité des soins de santé pour les personnes âgées. Dans ce système, une couche Fog analyse les données des capteurs et les envoie vers le Cloud via un format de données spécifique (JSON) et une méthode de communication (REST API).

De plus, selon l'étude présentée dans [12] ont proposé un système de santé basé sur l'IoT qui exploite des nœuds fog à divers endroits pour gérer les demandes des patients à travers différentes villes. Ce système surveille les données de santé des patients sur les nœuds, les cas critiques étant immédiatement transmis au serveur Cloud pour un suivi plus approfondi. Les situations non critiques sont gérées par les FN eux-mêmes. Bien que cette recherche ait évalué différentes configurations du système par des simulations pour évaluer les performances, leur travail manquait de comparaisons avec des systèmes de santé basés sur le Cloud ou le Fog existants.

Pour finir, la recherche en cours sur le FC pour l'IoMT met en évidence les avantages du traitement local des données sur les FN en termes de latence et de réactivité en temps réel. Cependant, des défis demeurent dans l'optimisation de l'allocation dynamique des tâches et des ressources, ainsi que dans la gestion de la variabilité de la charge de travail dans des environnements hautement dynamiques. Les méthodes proposées dans les études existantes montrent des résultats prometteurs, mais elles nécessitent une amélioration pour garantir des performances optimales dans des applications de santé complexes et sensibles à la latence.

Dans d'autre domaines on a :

Le travail présenté dans [13] utilise un algorithme d'évolution différentielle pour optimiser une application IoT. Il vise à réduire le temps de réponse et les coûts en prenant en compte la bande passante de communication et la latence, en utilisant une simulation avec CloudAnalyst. Toutefois, l'algorithme ne considère pas les priorités des applications des utilisateurs et manque d'un mécanisme de protection des données.

L'article [14] utilise dans leur algorithme la théorie de la partition de graphes qui consiste à diviser un graphe en sous-ensembles de nœuds pour optimiser les applications en temps réel. Il se concentre sur le nombre de nœuds déplacés et le temps d'exécution, avec pour objectif de réduire la latence des services. Bien que l'algorithme présente une faible

complexité, il n'a pas pris en compte l'utilisation de l'énergie ni été évalué dans une étude de cas réelle.

D'après [15] , les auteurs proposent l'utilisation de Parallel Othello Group qui est une méthode d'optimisation parallèle inspirée du jeu Othello, pour gérer efficacement la répartition des tâches dans des systèmes distribués. Cette approche est appliquée pour optimiser une application IoT mobile en se concentrant sur le débit. L'objectif est de réduire la surcharge de migration et d'offrir une mise en œuvre facile, mais l'algorithme ne prend pas en compte les caractéristiques de la répartition du graphe ni les paramètres de simulation pondérés différents.

L'algorithme proposé dans [16] utilise une approche heuristique pour optimiser l'utilisation des ressources, la variance de l'équilibrage de charge et le nombre de nœuds de calcul employés dans des applications IoT. Testé via la simulation avec CloudSim, il vise à améliorer l'utilisation des ressources et le débit, tout en prenant en compte la priorité de la variance de l'équilibrage de charge pour chaque nœud de calcul. Toutefois, l'algorithme ne considère pas la consommation d'énergie, le délai ni la migration des services.

Dans [17] ont proposé une solution basée sur l'autosimilarité pour l'équilibrage de charge dans les systèmes de grande échelle en fog computing, testée dans le cadre d'une application IoT via la simulation avec iFogSim. Leur approche repose sur trois composants clés : un module de surveillance qui collecte des informations intra-nœud et inter-nœud pour suivre les performances du système, un planificateur de politique d'équilibrage qui applique une politique d'équilibrage de charge distribuée en fonction de l'état des nœuds, et un messager qui facilite la communication entre les nœuds pour l'échange de messages et la coordination des tâches. Cette approche vise à optimiser l'efficacité et à équilibrer la charge dans un environnement IoT en FC.

II.5 Conclusion

L'équilibrage de charge et la gestion des ressources en FC sont essentiels pour garantir la performance et la scalabilité des applications, notamment celles de l'IoHT. Bien que des méthodes variées aient été proposées, il subsiste des manques, notamment en ce qui concerne l'efficacité énergétique et l'adaptabilité des algorithmes en temps réel. Des recherches futures

devraient se concentrer sur l'intégration de ces aspects pour améliorer la gestion des ressources et l'équilibrage de charge dans des environnements Fog plus complexes.

Chapitre III

Implémentation des algorithmes et évaluation des résultats

III. Chapitre III Implémentation des algorithmes et évaluation des résultats

III.1 Introduction

Dans le cadre de ce travail, nous visons à optimiser le placement des applications et l'équilibrage de charge dans un environnement Fog Computing. Pour réaliser cette opération, nous avons utilisé une approche hybride qui sert à intégrer la méthode du WRR et la méthode du LC, et une approche méta-heuristique qui est l'Algorithme de Colonie de Fourmis (ACO - Ant Colony Optimization en anglais).

III.2 Environnement de développement et de simulation

IFogSim est un outil de simulation et de modélisation, basé sur CloudSim (outil de simulation Cloud), développé par l'Université de Melbourne (Australie). La première version a été publiée autour de 2015 conçu pour modéliser les architectures décentralisées d'environnement Fog Computing. En 2017, iFogSim a introduit l'intégration de scénarios IoT réalistes, comme les villes intelligentes. Après 2020, Il a évolué grâce à l'ajout de modules pour l'IA en Edge et la 5G.

Il présente des caractéristiques uniques telles que la représentation graphique de l'installation, la capacité de prédire de manière fiable les résultats et diverses vues et paramètres pour surveiller les performances du système. Il permet de définir l'infrastructure, le placement des services et l'allocation des ressources. Il fonctionne sur le modèle d'application Sense-Process Actuate [24].

Pour notre recherche axée sur l'optimisation du placement d'applications et LB dans un environnement de FC, le choix d'IFogSim s'est imposé comme une évidence stratégique. Ce simulateur se distingue par sa conception spécifique aux architectures Fog, ce qui est essentiel pour la pertinence de nos résultats. C'est idéal pour notre étude car il modélise fidèlement les environnements Fog, permettant une évaluation précise des stratégies d'équilibrage de charge et de placement d'applications grâce à son architecture extensible et sa reconnaissance scientifique.

III.3 Description des outils utilisés

III.3.1 Java

Java est un langage de programmation informatique orienté objet a été créé à partir de 1991 par James Gosling et Patrick Naughton chez Sun Microsystems (qui a ensuite été acquis par Oracle), avec le soutien du con-fondateur Bill Joy, et sa première version publique a été lancée officiellement le 23 mai en 1995 au SunWorld. [23]

Ainsi, Il est un environnement d'exécution informatique portable ce qui signifie que le code écrit en Java ou généralement les applications Java peuvent s'exécuter sur différentes plates-formes (systèmes d'exploitation) sans nécessiter de recompilation, grâce à la machine virtuelle Java (JVM) qui est le cœur du JRE. À son tour, ce dernier inclut une bibliothèque standard pour créer tous les logiciels Java ce qui le rend portable et largement utilisé pour développer des applications.

De plus, ce JRE est inclut dans le JDK (Java Development Kit) qui est un ensemble d'outils de développement, implémenté par Sun Microsystems. Sa première version (1.0) sortie en janvier 1996, indispensable pour exécuter des applications Java.

III.3.2 Eclipse

Un environnement de développement intégré (IDE - Integrated Development Environment) open source, initié par IBM en 2001 avant d'être géré par Eclipse Foundation. Il fournit une plate-forme extensible supportant plusieurs langages grâce à son architecture modulaire basée sur OSGi et des plugins, mais à l'origine il est conçu pour le développement Java. Cet IDE est réputé pour sa flexibilité, sa communauté active et son écosystème d'outils pour l'édition de code, la compilation, le débogage et le déploiement d'applications.

III.3.3 Java Swing

Une boîte à outils GUI (Graphical User Interface) a été introduite avec la version JDK 1.2 en 1998. Elle est développée par Sun Microsystems, légère pour Java et faisant partie des JFC.

Le swing fournit un ensemble de composants (boutons, fenêtres, etc.) écrits en pur Java permettant de créer des interfaces utilisateur graphiques portables.

III.4 Architecture Fog

L'architecture IoT-Fog-Cloud illustrée dans la figure III-1 est utilisée dans notre étude pour les applications de santé et elle est définie par des paramètres spécifiques à chaque couche pour optimiser la gestion des données médicales.

Au niveau IoT, les paramètres incluent le nombre de nœuds capteurs (ex: ECG, température), la topologie du réseau (maillé ou étoile), la bande passante disponible pour la transmission des données, les types et taux de génération des tâches simulées, ainsi que la taille des données par échantillon.

La couche Fog est caractérisée par le nombre et les capacités de traitement et de stockage des FN, ainsi que les algorithmes de traitement local (filtrage, agrégation, détection d'anomalies simples).

Enfin, la couche Cloud est définie par ses capacités de calcul et de stockage illimitées pour l'analyse complexe et le stockage à long terme.

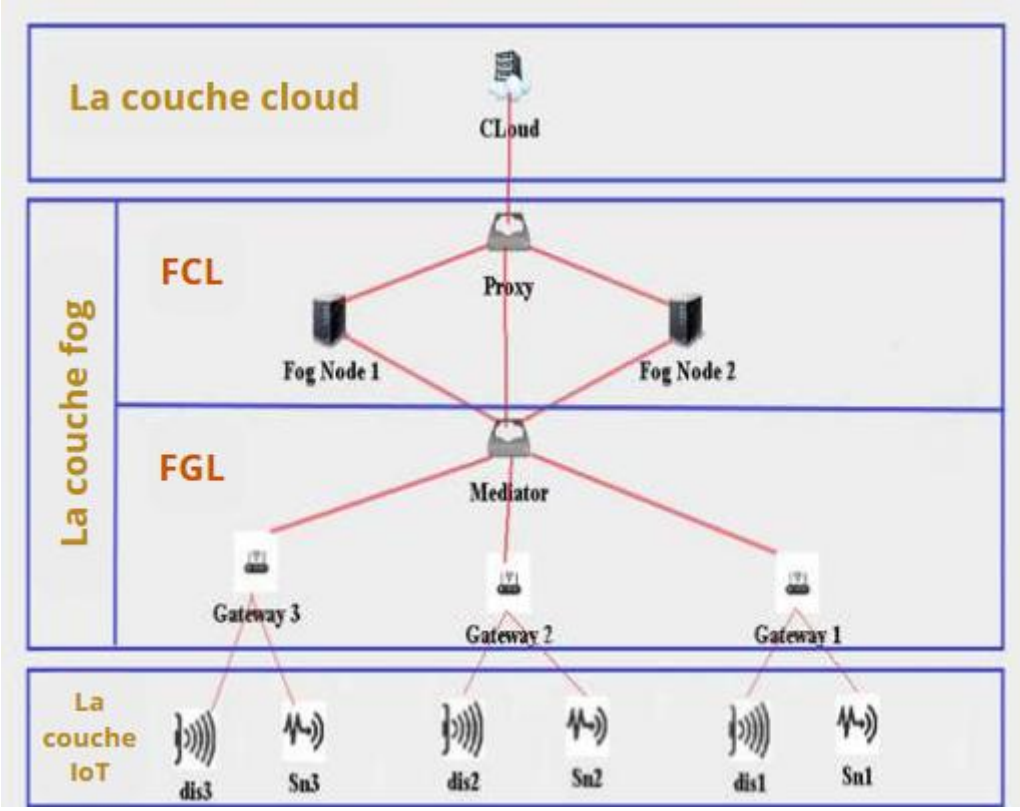

Figure III- 1: Architecture IoT-Fog-Cloud [28]

III.5 Scénario

Notre scénario se déroule dans un environnement hospitalier, où la gestion efficace des données des patients est cruciale. Au niveau de la couche IoT, les appareils sont des machines de santé connectées, capturant des données vitales. Ces données sont issues de [31]. Ils incluent des informations telles que l'âge, le genre, l'état cardiaque (maladie cardiaque ou normal), et la tension artérielle (blood pressure).

L'état cardiaque est le paramètre principal, avec le plus grand poids. Il indique si le patient a une maladie cardiaque (1) sinon (0). Ensuite, le risque lié à la tension artérielle a un poids moyen, Voici le tableau qui représente ce facteur :

Table III-1: Les niveaux de risque de la Tension Artérielle [32]

L'intervalle de tension	Valeur de risque	Niveau de risque
<120	0.0	Pas de risque
[120,130]	0.25	Faible
[130,140]	0.5	Moyen
[140,180[	0.75	Élevé
>180	1.0	Maximal

Après, le risque de l'âge, a un poids faible. Mais plus l'âge augmente, plus le risque augmente. Voici un tableau illustrant cette augmentation :

Table III-2: Les niveaux de risque de l'âge [33]

L'âge	Valeur de risque	Niveau de risque
[18,40]	0.0	Pas de risque
<50	0.25	Faible
<60	0.5	Moyen
<80	0.75	Élevé
>80	1	Très élevé

Dans la couche FGL, ces données et scores sont agrégés et transmis via des passerelles distinctes vers le médiateur. Au sein de cette couche, un unique médiateur centralisé gère le trafic et la coordination des données provenant des 45 applications, chacune représentant un ensemble unique de capteurs et d'informations patient. La FCL est composée de 7 FN, qui traitent et analysent localement les données pour des décisions rapides et une latence réduite. Enfin, un proxy assure la communication sécurisée et optimisée entre la couche de Fog et le Cloud, où les données à long terme sont stockées et des analyses plus complexes sont effectuées pour soutenir les diagnostics et la recherche médicale.

III.6 Implémentation

III.6.1 Implémentation de WRR

C'est un algorithme d'ordonnancement en rond avec poids et équilibrage de charge. Il est une amélioration de l'algorithme round robin qui distribue les requêtes de manière égale et séquentielle. Alors que, le WRR prend en compte les performances relatives des ressources en leur assignant des poids.

Un poids est associé à chaque ressource pour guider la distribution des requêtes selon leur capacité. Ce poids est un nombre entier qui représente sa capacité ou sa priorité. Si un dispositif est plus puissant ou avec plus de ressources (MIPS, RAM, bande passante) aura un poids plus élevé qu'un serveur moins puissant.

L'algorithme distribue les applications aux dispositifs de manière cyclique. L'ordonnanceur parcourt la liste des dispositifs et pour chacun, il attribue un nombre de requêtes proportionnel à son poids. Une fois toutes les ressources servies selon leurs poids respectifs, le cycle recommence. Il est léger, introduit un coût de calcul minimal, permet de mieux utiliser les ressources hétérogènes et Les poids peuvent être ajustés dynamiquement en fonction des changements de capacité des serveurs, ce qui le rend adapté aux applications en temps réel.

- **La méthode proposée**

L'algorithme WRR est implémenté en plusieurs étapes distinctes. Premièrement, les caractéristiques initiales des FN sont examinées, puis une phase de classement des dispositifs est effectuée en utilisant les résultats des méthodes de prise de décision multicritères comme TOPSIS et AHP [28].

Chaque dispositif se voit ensuite attribuer un poids basé sur ce classement. Enfin, l'affectation des patients est réalisée selon un ordre préétabli, tenant compte du niveau d'urgence de chaque patient. Le WRR assigne les patients aux dispositifs en utilisant une sélection pondérée qui favorise les dispositifs les plus performants selon les poids attribués, et il s'arrête une fois que tous les patients sont affectés.

Suite à cette affectation initiale, l'algorithme LC entre en jeu pour un rééquilibrage dynamique de la charge. LC démarre son processus en analysant l'état actuel des "connexions

actives" sur chaque dispositif Fog, telles qu'établies par le WRR. Son objectif est de réduire la charge des FN. Il identifie continuellement le nœud ayant le plus de connexions actives et le nœud en ayant le moins. Si la différence de connexions entre ces deux dispositifs dépasse un seuil de 1, cela indique un déséquilibrage. Donc, LC réaffecte un patient du nœud le plus chargé vers le nœud le moins chargé. Ce processus itératif se poursuit jusqu'à ce que la distribution des connexions soit considérée comme équilibrée, c'est-à-dire que la différence entre le maximum et le minimum de connexions actives soit faible ou nulle. Ainsi, LC perfectionne la répartition des tâches initiée par WRR pour une meilleure uniformité de la charge.

- **Résultats**

WRR+LC a effectué le placement des applications en fonction de poids prédéterminés des nœuds, favorisant les nœuds les plus puissants, avant que le composant LC ne prenne en charge l'équilibrage. Avant l'application du LC, le WRR montre une latence élevée et un équilibrage de charge moyen. Après l'application du LC, LB s'améliore, devenant efficace. Pour cela on a calculé la latence pour chaque module de chaque patient, ainsi que l'écart type pour vérifier l'équilibrage de charge.

Premièrement, la latence est calculée par la formule suivante :

Équation III-1: La latence du module selon iFogSim

$$\text{Latence Totale du Module} = \text{UpDelay} + \text{ProcessingDelay} + \text{DownDelay}$$

UpDelay représente le temps de transmission montante entre le dispositif source (mobile) et le dispositif destination (le FN) ou le traitement des données aura lieu. Il est calculé comme montre dans l'équation III-3 :

Équation III-2: Le temps de transmission montant

$$\text{UpDelay} = \frac{\text{Taille du module}}{\text{Bande passante montante}}$$

Ensuite, ProcessingDelay représente le temps qu'il met le FN pour traiter les données du module.Voici la formule de calcul :

Équation III-3: Le temps de traitement du module

$$\text{ProcessingDelay} = \frac{\text{La longueur du tuple}}{\text{MIPS du dispositif}}$$

La longueur de tuple présente la charge de calcul que le processeur doit effectuer pour traiter un paquet de données. Elle est exprimée en MIPS.

Par la suite, DownDelay représente le temps de transmission de résultats descendant depuis le FN vers le dispositif destination (Mobile).

Voici l'équation :

Équation III-4: Le temps de transmission descendant

$$\text{DownDelay} = \frac{\text{Taille du module}}{\text{Bande passante descendante}}$$

On passe ensuite à l'écart type, pour tester l'équilibrage de charge. Il suit ces étapes :

- La moyenne : Le nombre de taches par nœud

Équation III-5: La moyenne (μ)

$$\mu = \frac{\sum \text{taches} - \text{par} - \text{noeud}}{\text{nombre totale de nœuds}}$$

- La variance : La moyenne des carrés des différences par rapport à la moyenne.

Équation III-6: La variance

$$\sigma^2 = \frac{\sum(\text{taches par noeud} - \mu)^2}{\text{nombre totale de noeuds}}$$

-L'écart type : La racine carrée de la variance

Équation III-7: L'écart type

$$\sigma = \sqrt{\sigma^2}$$

Plus que l'écart type est faible, plus que la répartition des taches est équilibrée.

III.6.2 Implémentation d'ACO

L'algorithme d'optimisation par colonies de fourmis a été proposé par Marco Dorigo dans les années 1990. L'ACO est un algorithme méta-heuristique bio-inspiré du comportement des fourmis dans la recherche de nourriture. [29] Les fourmis communiquent indirectement via des phéromones qui sont des substances chimiques déposées sur les chemins qu'elles empruntent. Les chemins les plus courts ou les plus prometteurs accumulent plus de phéromones, attirant davantage de fourmis au fil du temps. Cet algorithme est utilisé

pour résoudre des problèmes d'optimisation combinatoire, comme le fameux problème du voyageur de commerce et dans le placement de tâches.

Son idée fondamentale repose sur la manière dont les fourmis trouvent le chemin le plus court entre leur nid et une source de nourriture. Individuellement, une fourmi est simple, mais collectivement, elles forment un système capable de résoudre des problèmes complexes grâce à la stigmergie, un mécanisme de coordination indirecte via des modifications de l'environnement.

ACO simule des fourmis artificielles qui parcourent un graphe pour construire des solutions. Leurs choix de chemin sont orientés par une combinaison d'informations sur le problème comme la distance et la quantité de phéromone présente sur chaque arête. Une fois toutes les solutions générées, les arêtes qui font partie des meilleures sont renforcées avec davantage de phéromone, guidant ainsi les futures fourmis vers des parcours optimaux.

La méthode proposée s'appuie sur une approche ACO pour l'affectation quasi-optimale des applications dans un environnement FC. Le processus est itératif et vise à trouver les meilleurs chemins pour l'affectation des modules.

Les itérations commencent par l'initialisation de l'environnement. Cela implique de définir un ensemble de paramètres globaux. La qualité d'un lien est une métrique cruciale dans la construction des chemins. Elle est calculée en combinant plusieurs facteurs de performance normalisés des dispositifs connectés par ce lien. Pour chaque dispositif, les scores normalisés de MIPS, RAM, BP et la latence sont d'abord déterminés en utilisant la formule de normalisation min-max :

Équation III-8: La formule de normalisation

$$ValeurNormalisée = \frac{valeur - min}{max - min}$$

Où valeur est la valeur actuelle de la métrique (MIPS, RAM, BP, Latence), min est la valeur minimale possible, et max est la valeur maximale possible pour cette métrique Il est important de noter que ces valeurs de critères sont générées aléatoirement dans des plages bien définies pour chaque simulation.

Ces scores normalisés sont ensuite agrégés pour former un score composite pour chaque dispositif, en appliquant des poids spécifiques et bien définis fixer à 0.25. La somme totale des poids est égale à 1, ce qui garantit une contribution égale de chaque métrique à la qualité composite du dispositif :

Équation III-9: La Qualité

$$\text{Qualité} = (\text{MIPSScore} * 0.25) + (\text{RamScore} * 0.25) + (\text{BPScore} * 0.25) + (\text{LatenceScore} * 0.25)$$

Ensuite, la qualité d'un lien entre ces dispositifs (dispositif 1 : la source, dispositif 2 : la destination) est calculée comme la moyenne de leurs qualités respectives :

Équation III-10: La qualité combinée

$$QualiteCombinee = \frac{Qualite1 + Qualite2}{2}$$

- La qualité combinée représente la qualité du lien pour les fourmis.

- Qualite1 représente la qualité du dispositif 1.

- Qualite2 représente la qualité du dispositif 2.

La construction du chemin par chaque fourmi est l'étape centrale de chaque itération. Dans notre cas on a 5 itérations. Dans la première itération, Chaque fourmi construit séquentiellement un chemin en partant d'un dispositif source toujours commence du médiateur et en sélectionnant le prochain dispositif destination. Mais lorsque les niveaux de phéromones sont uniformes ou très faibles (Au début), les fourmis choisie les nœuds de manière plus aléatoire pour construire leurs chemins. C'est essentiel pour la phase d'exploration, car cela permet aux fourmis de découvrir différents chemins, plutôt que de suivre uniquement des chemins déjà balisés. Étant donné l'absence de communication directe au même niveau dans iFogSim, toute interaction entre deux FN passe par le médiateur. La décision de choisir le prochain nœud est probabiliste et dépend de deux facteurs : le niveau de phéromone sur le lien vers le prochain nœud et la qualité de ce lien. Les fourmis sont plus susceptibles de choisir des chemins avec des niveaux de phéromones plus élevés et une meilleure qualité de lien. Pour éviter que toutes les fourmis ne suivent toujours le même meilleur chemin dès le début, une légère composante aléatoire peut être introduite dans le

processus de sélection, surtout dans les premières itérations, favorisant ainsi l'exploration de nouvelles pistes.

Après que toutes les fourmis ont construit leurs chemins et que la meilleure solution de l'itération a été identifiée, la mise à jour des phéromones a lieu. Cette étape a deux composantes principales : Les phéromones qui s'évaporent progressivement sur tous les chemins pour simuler l'oubli.

Équation III-11: La formule d'évaporation

$$\text{pheromone\_nouvelle} = \text{pheromone\_ancienne} \times (1 - \text{EVAPORATION})$$

Où EVAPORATION est un coefficient de taux d'évaporation, qui est fixé à 0.5.

Les fourmis qui ont trouvé des chemins de meilleure qualité déposent des phéromones supplémentaires sur les liens qu'elles ont utilisés. Le dépôt est proportionnel à la qualité de la solution trouvée par la fourmi :

Équation III-12: La formule de dépôt de phéromones

$$\text{DepotDePheromones} = Q \times \text{QualiteTotale}$$

Où Q=500 est une constante de dépôt, qui désigne la quantité de phéromones déposée par chaque fourmi, et QualiteTotale est une mesure de la qualité du chemin trouvé par la fourmi, est calculée en sommant les QualiteCombinee de tous les liens qui constituent ce chemin. Les phéromones mises à jour reflètent ainsi la mémoire collective des fourmis concernant les chemins les plus prometteurs.

Ce processus itératif de construction de chemin et de mise à jour des phéromones se répète sur 5 itérations pour 10 fourmis. Après la première itération, les itérations suivantes se basent du meilleur chemin obtenu de l'itération précédente, permettant ainsi à l'algorithme de converger progressivement vers la solution optimale.

- **Résultats**

L'approche ACO+LC a tiré parti du meilleur chemin identifié par l'algorithme ACO, priorisant les nœuds les mieux classés pour l'affectation initiale. Le module d'équilibrage de

charge (LC) est ensuite intervenu pour redistribuer la charge et, si besoin, réaffecter les applications.

De la même façon que WRR, est calculer la latence et l'écart type. Il a démontré une meilleure performance en termes de latence comparée à WRR+LC, et a également montré une capacité supérieure à optimiser la distribution de la charge entre les nœuds après LC, ce qui en fait une solution plus performante pour le FC.

III.7 Présentation de l'application

La réalisation de cette application a été faite sous Eclipse version 2024-12 (4.34.0) en se basant sur JDK 21 avec l'utilisation de langage javaSwing pour créer des interfaces, et JfreeChart pour générer des graphiques, et des diagrammes en 2D de très bonne qualité.

❖ **Interface principale**

Dès le lancement de notre application, la fenêtre d'accueil illustrée dans la figure III -2 apparaît.

Figure III- 2: Interface Accueil

Cette fenêtre contient deux boutons :

- « **Entrer** » pour lancer la simulation.
- « **Quitter** » pour sortir sans lancer la simulation.

En cliquant sur **Entrer**, l'interface de formulaire s'affiche dans la figure III -3.

Figure III- 3: Interface Formulaire

Cette IHM contient également deux champs de saisie de type JTextField, le premier champ désigne le nombre de dispositifs Fog (Nbr_fog) qu'il doit être positif et non nul. Ce Nbr_fog est utilisé dans la méthode createFogDevices() pour créer les objets FogDevices avec les attributs suivants : MIPS, RAM, bande passante montante (upBw), bande passante descendante (downBw), latence montante (uplinkLatency) et latence descendante (downlinkLatency).

Ainsi, le deuxième champ désigne le nombre de passerelles (Nbr_GW) ou applications. Dans la méthode addArea(), Nbr_GW détermine combien de passerelles sont créées pour chaque zone (area).

Nous avons aussi trois boutons de type JButton :

- Bouton « **WRLC** » : permet d'exécuter et étudier l'algorithme Wrr et LC.
- Bouton « **ACO** » : permet d'exécuter et étudier l'algorithme ACO et LC.
- Bouton « **Retour** » : permet de retourner à l'interface d'accueil.

III.8 Comparaison

III.8.1 En termes de temps de réponse moyen

- **1er cas : Le nombre de nœuds est fixé à 7, tandis que les applications sont variées.**

La figure III - 4 montre que pour un faible nombre d'applications, les performances des deux algorithmes sont quasiment identiques, avec une latence minimale. Cependant, à mesure que le nombre d'applications augmente, la latence devient plus élevée pour les deux, mais nous constatons globalement que l'ACO maintient une latence légèrement inférieure à celle de WRR, ce qui suggère une meilleure efficacité d'ACO face à une charge de travail croissante en termes de latence.

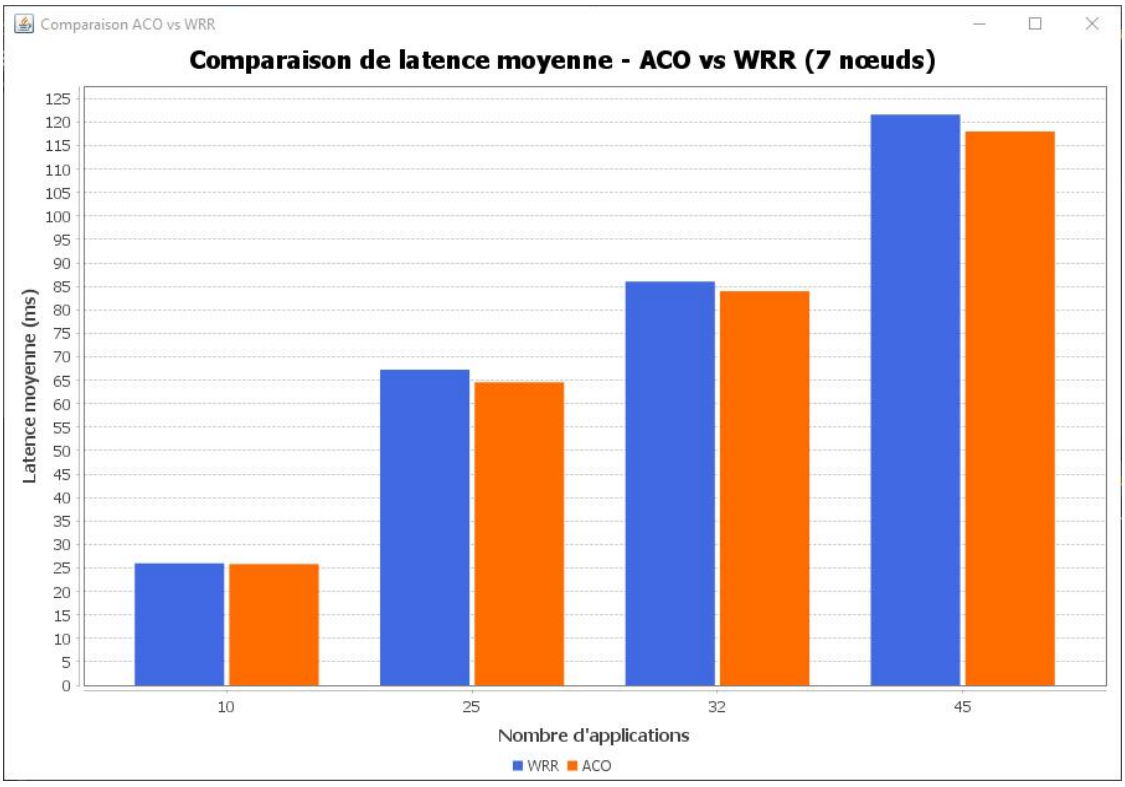

Figure III- 4: Impact sur la latence 1 - WRR vs ACO

- **2eme cas : le nombre d'applications fixé à 40, le nombre de nœuds a été varié.**

Nous constatons que pour la figure III-5, l'augmentation du nombre de FN réduit la latence pour les deux algorithmes. En effet, lorsque le nombre de nœuds augmente, les applications se répartissent sur davantage de ressources, soulageant ainsi la charge sur chaque

nœud, contrairement à un petit nombre de nœuds où ils seraient surchargés. Pour un petit nombre de nœuds, l'ACO affiche une latence légèrement plus faible que le WRR. À mesure que le nombre de nœuds Fog augmente, les latences des deux algorithmes deviennent très proches. L'ACO offre une meilleure performance initiale dans des environnements plus contraints, et des performances comparables lorsque plus de ressources sont disponibles.

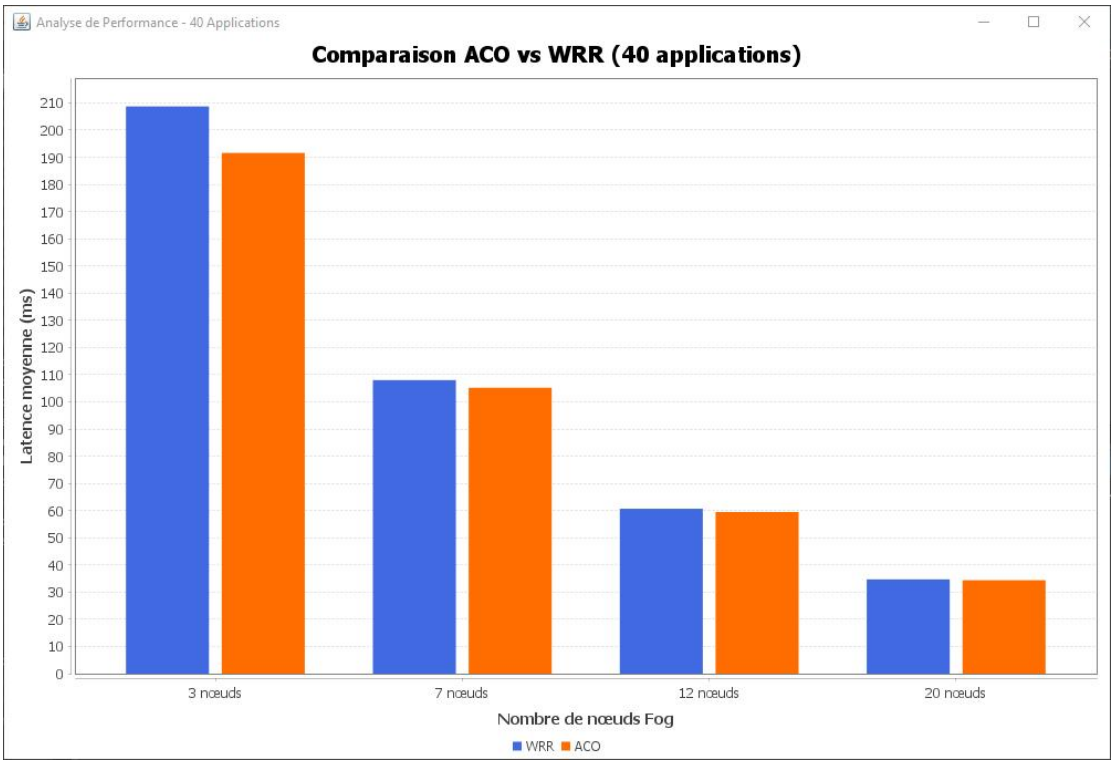

Figure III- 5: Impact sur la latence 2 - WRR vs ACO

- **3eme cas : Le nombre d'applications et le nombre de nœuds ont été variés simultanément.**

La figure III-6 montre que pour un faible nombre de nœuds et d'applications, ACO+LC surpasse déjà WRR+LC. Mais, lorsque le nombre de nœuds et d'applications augmente, WRR+LC peut connaître une dégradation significative de ses performances de latence, notamment à des points intermédiaires, tandis qu'ACO+LC gère cette augmentation de charge beaucoup plus efficacement, maintenant systématiquement une latence inférieure sur l'ensemble des configurations testées.

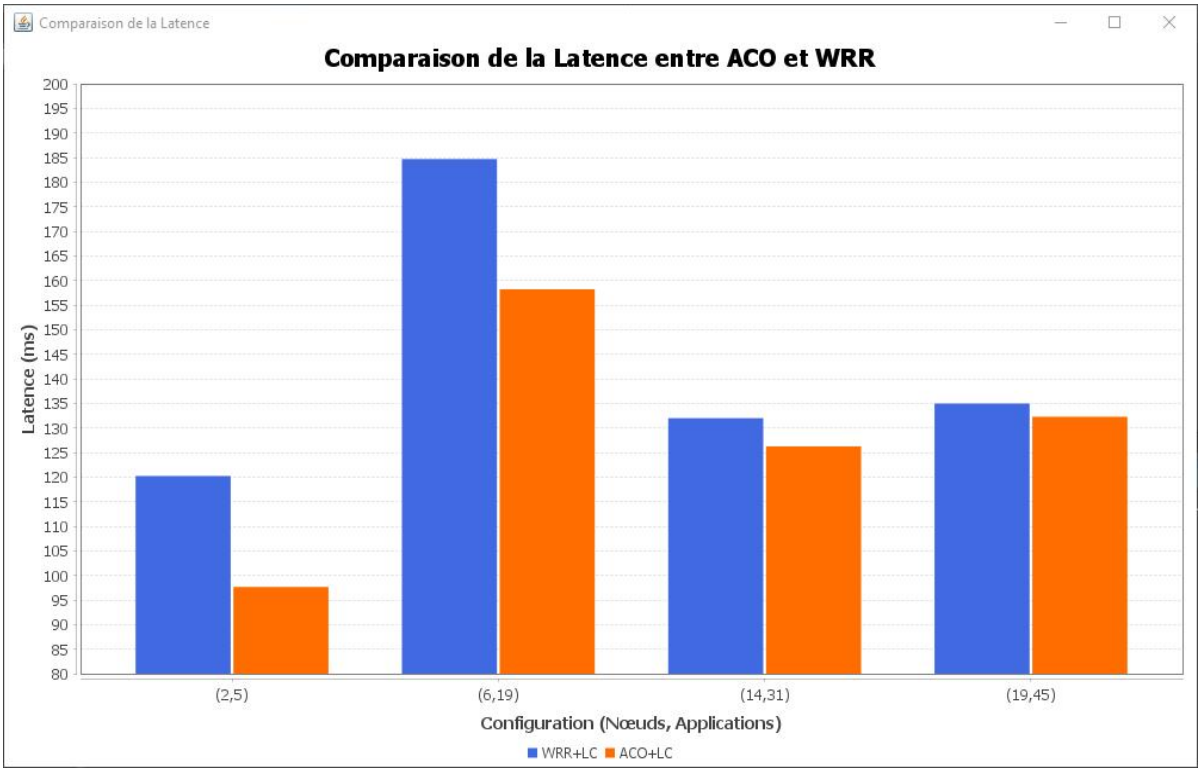

Figure III- 6: Impact sur la latence 3 - WRR vs ACO

III.8.2 En termes de temps d'exécution de l'algorithme

Nous calculons le temps d'exécution de WRR+LC, et celui de ACO+LC.

Nous observons dans la Figure III-7 que l'algorithme WRR a un temps d'exécution significativement plus court (environ 24.8 ms) que l'algorithme ACO, dont le temps d'exécution est nettement plus élevé (autour de 49.1 ms). Donc, la Figure III.4 montre que l'algorithme WRR est considérablement plus rapide en termes de temps d'exécution, car il s'arrête dès qu'une solution "valide" est trouvée. Tandis qu'ACO continue son exploration pour converger vers la solution la plus optimale possible.

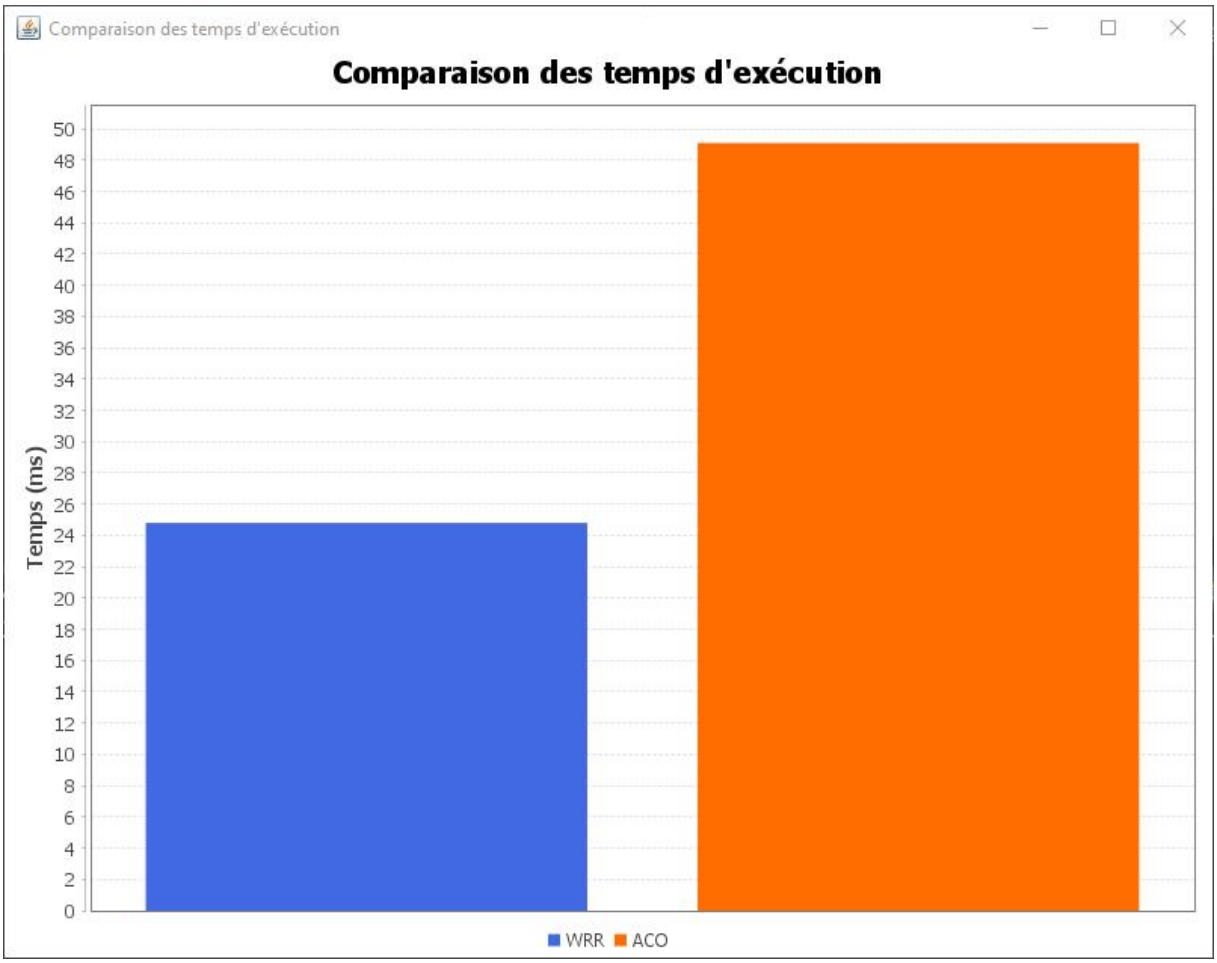

Figure III- 7: Impact sur le temps d'exécution WRR+LC et ACO+LC

III.8.3 En termes de répartition des charges

Nous calculons l'écart type de chaque algorithme pour vérifier l'efficacité. Les résultats représentent la moyenne de l'écart type pour 7 nœuds et 45 applications.

Voici un exemple qui montre la répartition de la charge sur les 7 FN :

La répartition des applications par nœud révèle que le WRR répartit les applications en utilisant une sélection pondérée qui favorise les dispositifs ayant le plus grand poids, tandis que l'ACO affecte les taches aléatoirement selon le meilleur chemin trouvé, comme illustré dans la figure III- 8.

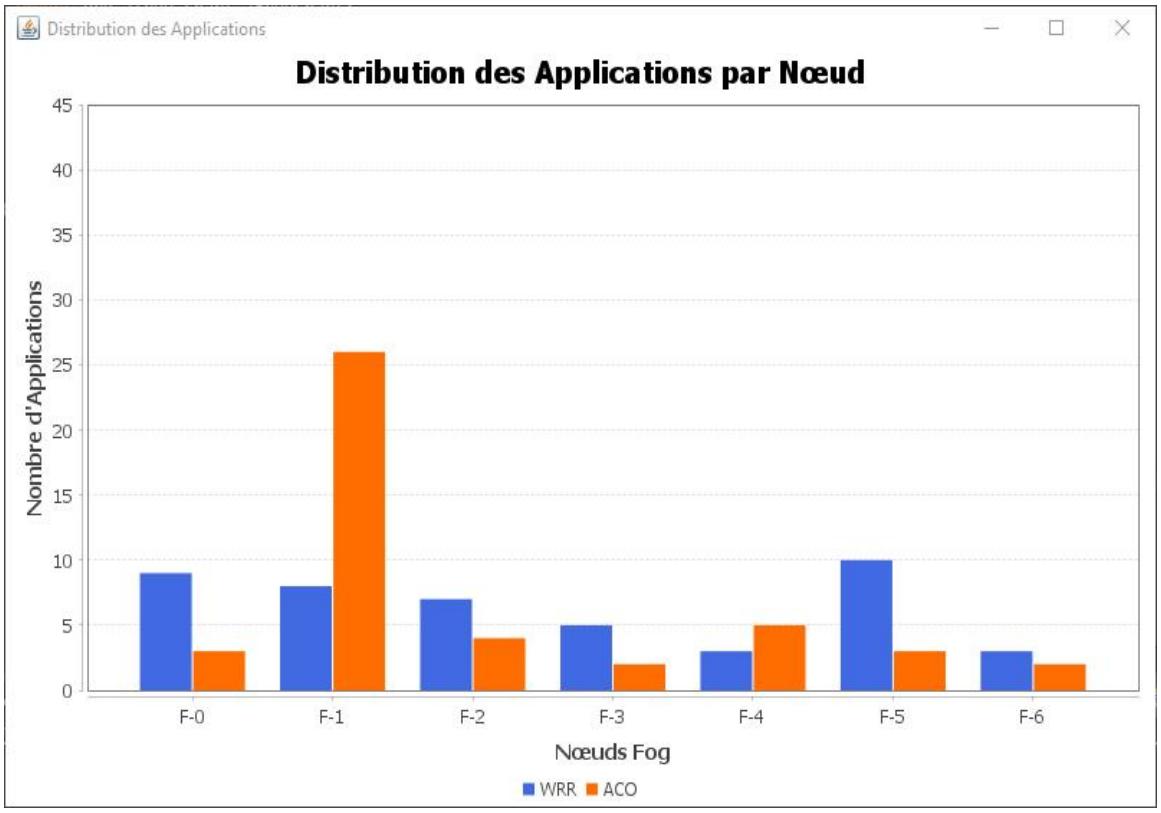

Figure III- 8: Distribution des applications par nœud Avant LC

Après LC, il répartit la charge en minimisant la différence de charge entre les nœuds, comme montre dans la figure III- 9.

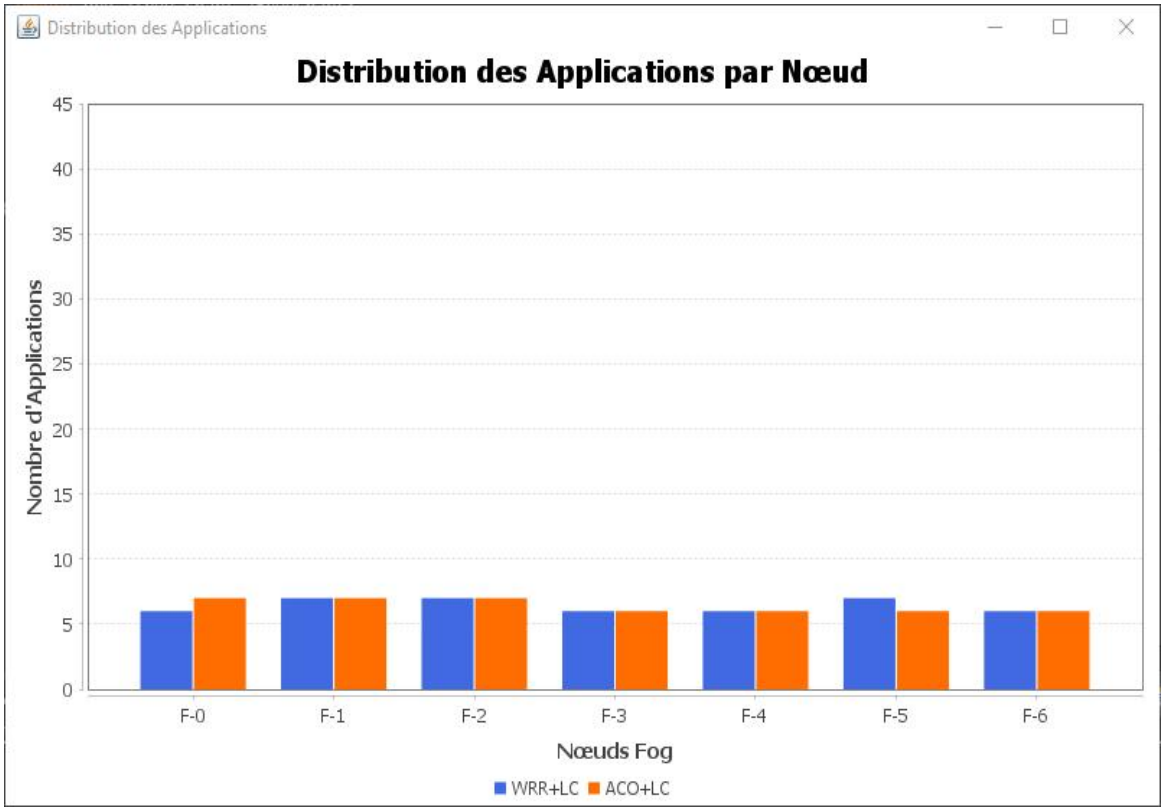

Figure III- 9: Distribution des applications par nœud Après LC

- **1er Cas : Avant LC**

Nous constatons dans la figure III-10, WRR présente un écart type plus faible que l'ACO. Cela revient à la stratégie de WRR qui se base sur les poids des dispositifs pour distribuer les requêtes de manière cyclique. Cette approche déterministe et régulière conduisant à un faible écart type des temps de réponse.

Tandis qu'ACO se base sur le meilleur chemin obtenu, il favorise les nœuds les premiers classés ; il explore différentes voies, parfois en envoyant des requêtes sur des nœuds qui ne seraient pas le choix évident à un instant précis, afin de découvrir des chemins plus efficaces à long terme ou d'éviter des congestions futures. Cette nature exploratoire et adaptative résultant en un écart type plus élevé.

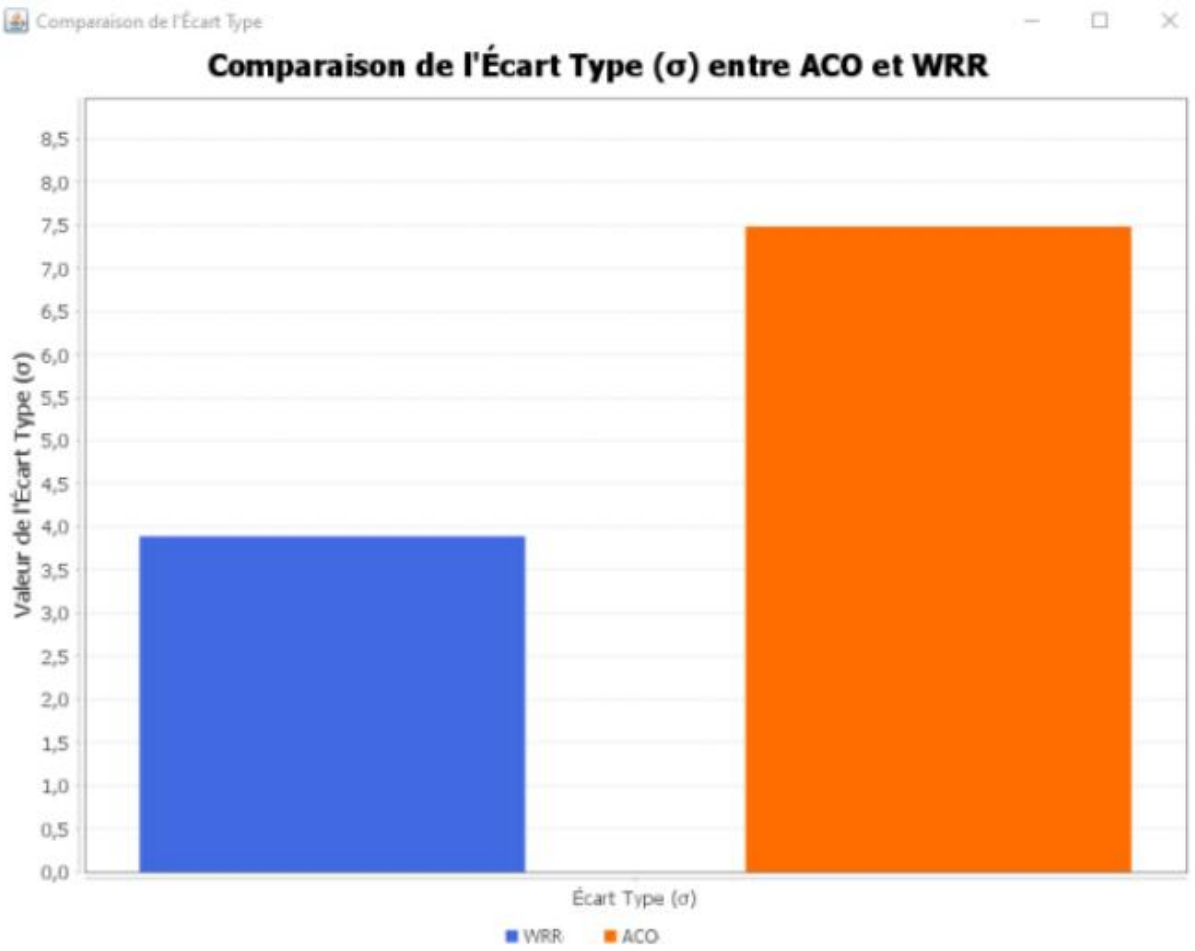

Figure III- 10: Impact sur la charge avant LC - WRR vs ACO

- **2éme Cas : Après LC**

La Figure III-11, montre que la valeur de l'écart type est pratiquement identique pour les algorithmes WRR et ACO. Cette similarité est une observation importante qui peut s'expliquer par le fait que les deux méthodes, bien qu'ayant des approches différentes, s'appuient sur l'équilibrage de charge via la métrique LC. Par conséquent, elles tendent à distribuer la charge de manière similaire en termes de variabilité, aboutissant à une stabilité comparable de la charge entre les nœuds.

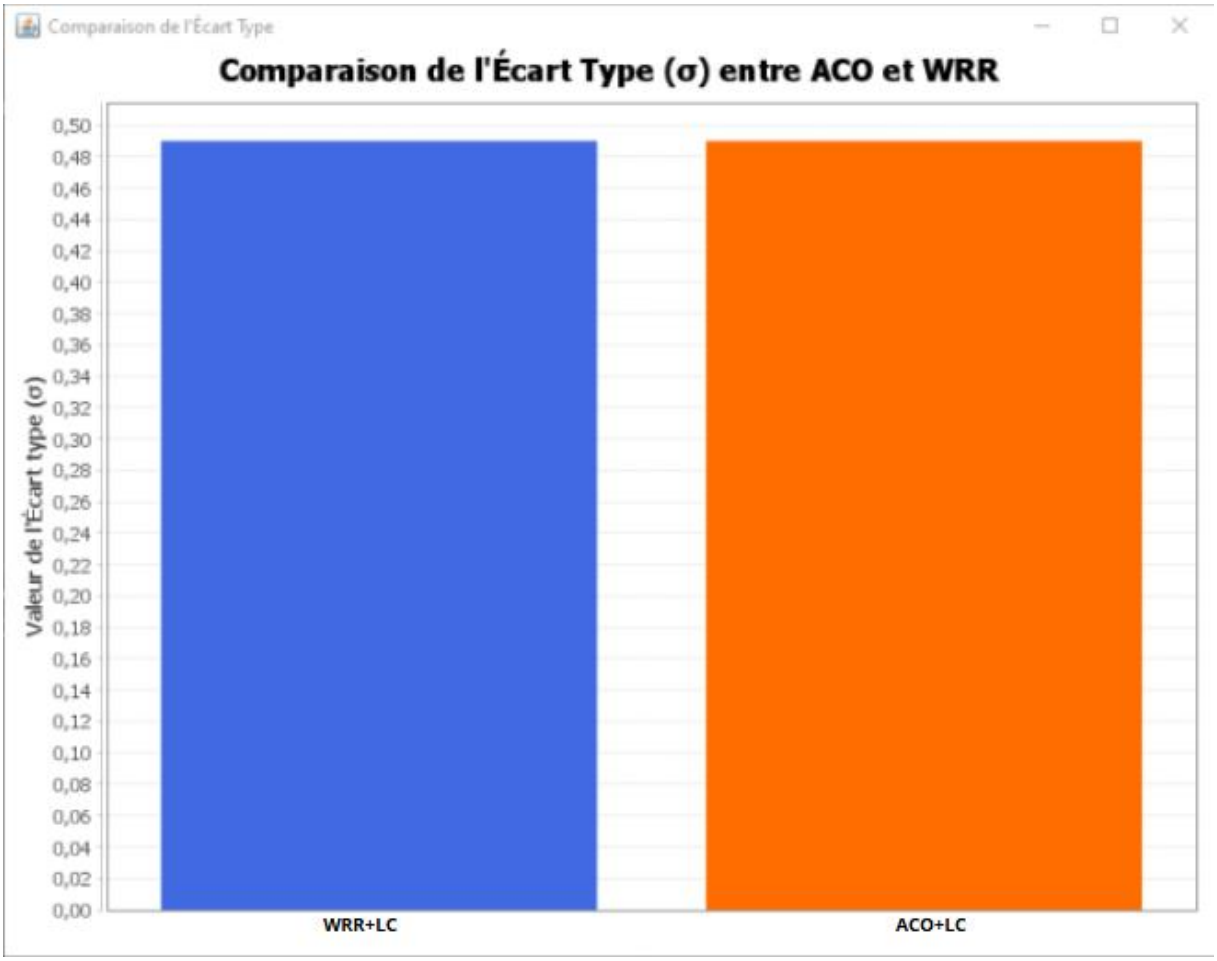

Figure III- 11 : Impact sur la charge après LC - WRR vs ACO

III.9 Conclusion

Nous avons présenté dans ce chapitre notre contribution dans le cadre de ce PFE. Notre travail est concentré sur l'équilibrage de charge, visant à optimiser le placement des tâches de manière efficace et à améliorer les performances globales du système. Pour atteindre cet objectif, nous avons exploré et mis en œuvre des méthodes statiques et dynamiques, spécifiquement WRR, LC et ACO. Notre démarche visait à développer une solution optimale pour but de réduire la latence.

Les résultats obtenus lors de l'étude comparative démontrent clairement l'intérêt de combiner ces approches pour la minimisation de la latence prouvent également l'efficacité supérieure de l'ACO pour la réduction de la latence, en particulier sous des charges variables et élevées, par rapport au comportement plus classique de WRR.

Conclusion générale

Conclusion générale

Nous avons travaillé dans ce projet de fin d'études sur l'optimisation du placement et de l'équilibrage de charge dans un environnement FC. L'objectif principal était d'améliorer la QoS et de minimiser la latence en distribuant efficacement les applications et la charge de travail au sein du Fog.

Dans ce contexte, nous avons utilisé le simulateur iFogSim pour évaluer deux approches hybrides : WRR+LC et ACO+LC. Nos applications ont été traitées selon un niveau d'urgence, permettant de classer les patients et de prioriser leur traitement.

Concernant les mécanismes de placement et d'équilibrage, notre étude s'est basée sur différentes méthodes hybrides combinant des algorithmes statiques comme WRR, dynamiques comme LC et méta-heuristiques telles que ACO. Parmi celles-ci, l'approche ACO+LC, a démontré une nette supériorité. Cette méta-heuristique a prouvé qu'elle offrait une solution plus optimale par rapport à d'autres méthodes comme WRR+LC. Les résultats ont clairement indiqué que, bien que les deux approches aient cherché un équilibrage efficace, ACO+LC a une capacité supérieure à optimiser la distribution de la charge entre les nœuds, ce qui en fait la solution la plus performante dans ce contexte.

Malgré les résultats prometteurs obtenus, cette étude présente certaines limites inhérentes à sa portée. La première concerne les algorithmes d'optimisation explorés (WRR+LC et ACO+LC) qui, bien que performants, ne représentent qu'un ensemble spécifique d'approches, et dont la robustesse face à des scénarios de charge extrêmes ou des topologies très dynamiques n'a pas été pleinement évaluée. De plus, les risques liés à l'intégrité et à la sécurité des données sensibles tout au long de leur cheminement depuis les appareils IoT vers les FN pour le traitement, leur stockage temporaire, et leur éventuel transfert ou retour constituent une préoccupation majeure. Par ailleurs, l'évaluation a été menée dans iFogSim dont le niveau d'abstraction élevé et la simplification des modèles de ressources ne permettent pas de capturer toutes les nuances des déploiements réels, notamment les interactions de bas niveau, la gestion fine de l'hétérogénéité matérielle, ou les dynamiques imprévisibles du monde réel. Enfin, bien que la latence et le temps de réponse aient été améliorés, d'autres aspects cruciaux pour la performance globale des systèmes FC,

tels que la consommation énergétique des nœuds ou la résilience face aux pannes complexes, n'ont pas été approfondis.

Ces limites identifiées ouvrent cependant la voie à de nombreuses perspectives de recherche futures. Il est pertinent d'envisager des avancées significatives au-delà des optimisations initiales. Il est particulièrement intéressant de développer des méthodes hybrides plus avancées, comme l'association d'ACO et du Deep Reinforcement Learning pour la décision en temps réel. Ces approches viseraient non seulement à surpasser les performances actuelles en termes de latence et d'équilibrage de charge, mais aussi à améliorer la scalabilité, la tolérance aux pannes et la gestion dynamique des ressources dans des environnements Fog complexes. En parallèle, l'intégration de l'application de Blockchain pour renforcerait la sécurité et la traçabilité des données, des aspects cruciaux pour les applications critiques.

Références Bibliographiques

Références bibliographiques

[1] Fog Computing. Disponible sur le site : Fog Computing - GeeksforGeeks. Consulté le 05/05/2025

[2] NIST Special Publication 500-299 (2017) – NIST's Definition of Fog Computing and Related Concepts.Disponible sur: (PDF) The NIST Definition of Fog Computing Consulté le : 05/05/2025

[3] Haghi Kashani, Mostafa, Amir Masoud Rahmani, and Nima Jafari Navimipour. "Quality of service-aware approaches in fog computing." International Journal of Communication Systems 33.8 (2020): e4340.

[4] Mahmud, Redowan, Kotagiri Ramamohanarao, and Rajkumar Buyya. "Latency-aware application module management for fog computing environments." ACM Transactions on Internet Technology (TOIT) 19.1 (2018): 1-21.

[5] Mahmoud, Mukhtar ME, et al. "Towards energy-aware fog-enabled cloud of things for healthcare." Computers & Electrical Engineering 67 (2018): 58-69.

[6] Taneja, Mohit, and Alan Davy. "Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm." 2017 IFIP/IEEE symposium on integrated network and service management (IM). IEEE, 2017.

[7] Kumar, Rohit, and Neha Agrawal. "RBAC-LBRM: An RBAC-based load balancing assisted efficient resource management framework for IoT-edge-fog network." IEEE Sensors Letters 6.8 (2022): 1-4.

[8] Panda, Sanjaya Kumar, Divyanshu Tyagi, and Rashmi Ranjan Rout. "GVMP: A Novel Internet of Things Application Module Placement Strategy for Heterogeneous Infrastructure using iFogSim." 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT). IEEE, 2024.

[9] Tuli, Shreshth, et al. "HealthFog: An ensemble deep learning based Smart Healthcare System for Automatic Diagnosis of Heart Diseases in integrated IoT and fog computing environments." Future Generation Computer Systems 104 (2020): 187-200.

[10] Vora, Jayneel, et al. "FAAL: Fog computing-based patient monitoring system for ambient assisted living." 2017 IEEE 19th international conference on e-health networking, applications and services (Healthcom). IEEE, 2017.

[11] Ben Hassen, Hafedh, Wael Dghais, and Belgacem Hamdi. "An E-health system for monitoring elderly health based on Internet of Things and Fog computing." Health information science and systems 7.1 (2019): 24.

[12] Khattak, Hasan Ali, et al. "Utilization and load balancing in fog servers for health applications." EURASIP Journal on Wireless Communications and Networking 2019.1 (2019): 1-12.

[13] Manasrah, Ahmad M., Ala'A. Aldomi, and Brij B. Gupta. "An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment." Cluster Computing 22 (2019): 1639-1653.

[14] He, Xiuli, et al. "A novel load balancing strategy of software-defined cloud/fog networking in the Internet of Vehicles." China Communications 13.Supplement2 (2016): 140-149.

[15] Ningning, Song, et al. "Fog computing dynamic load balancing mechanism based on graph repartitioning." China Communications 13.3 (2016): 156-164.

[16] Xu, Xiaolong, et al. "Dynamic resource allocation for load balancing in fog environment." Wireless Communications and Mobile Computing 2018.1 (2018): 6421607.

[17] Li, Changlong, et al. "SSLB: self-similarity-based load balancing for large-scale fog computing." Arabian Journal for Science and Engineering 43.12 (2018): 7487-7498.

[18] Confais, Bastien. Conception d'un système de partage de données adapté à un environnement de Fog Computing. Diss. Université de Nantes, 2018.

[19] Bonomi, Flavio, et al. "Fog computing and its role in the internet of things." Proceedings of the first edition of the MCC workshop on Mobile cloud computing. 2012.

[20] Vaquero, Luis M., and Luis Rodero-Merino. "Finding your way in the fog: Towards a comprehensive definition of fog computing." ACM SIGCOMM computer communication Review 44.5 (2014): 27-32.

[21] Dastjerdi, Amir Vahid, et al. "Fog computing: Principles, architectures, and applications." Internet of things. Morgan Kaufmann, 2016. 61-75.

[22] Hong, Cheol-Ho, and Blesson Varghese. "Resource management in fog/edge computing: a survey on architectures, infrastructure, and

algorithms." ACM Computing Surveys (CSUR) 52.5 (2019): 1-37.

[23] Java (langage). Disponible sur le site: https://www.techno-science.net/glossaire-definition/Java-langage.html . Consulté le 22-05-2025.

[24] Mahmud, Redowan, and Rajkumar Buyya. "Modelling and simulation of fog and edge computing environments using iFogSim toolkit." arXiv preprint arXiv:1812.00994 (2018).

[25] Dr. Eric Topol Addresses Baylor College of Medicine Grads. Disponible sur le lien: Dr. Eric Topol Addresses Baylor College of Medicine Grads. Consulté le 03/06/2025

[26] Gupta, Harshit, et al. "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments." Software: Practice and Experience 47.9 (2017): 1275-1296.

[27] Mahmud, Redowan, et al. "Quality of Experience (QoE)-aware placement of applications in Fog computing environments." Journal of Parallel and Distributed Computing 132 (2019): 190-203.

[28] Bochra, Arichi, and Amraoui Asma. "Enhancing Application Placement in Cloud-Fog Environment Based on Multicriteria Decision Making." 2024 3rd International Conference on Advanced Electrical Engineering (ICAEE). IEEE, 2024.

[29] Dorigo, Marco, and Thomas Stützle. "The ant colony optimization metaheuristic: Algorithms, applications, and advances." Handbook of metaheuristics (2003): 250-285.

[30] Fog and IoT. Disponible sur: fog and edge white papers - Industry IoT Consortium Consulté le 05/05/2025

[31] Heart Failure Prediction Dateset. Disponible sur le site: Heart Failure Prediction Dataset Conuslté le: 18/06/2025.

[32] Understanding Blood Pressure Readings. Disponible sur le site: Understanding Blood Pressure Readings | American Heart Association Consulté le: 20/06/2025

[33] Impact of Age on the Importance of Systolic and Diastolic Blood Pressures for Stroke Risk. Disponible sur le lien : Impact of Age on the Importance of Systolic and Diastolic Blood Pressures for Stroke Risk |

Hypertension. Consulté le: 20/06/2025

[34] Sun, Yan, Fuhong Lin, and Haitao Xu. "Multi-objective optimization of resource scheduling in fog computing using an improved NSGA-II." *Wireless Personal Communications* 102 (2018): 1369-1385.

[35] De Benedetti, Massimiliano, et al. "JarvSis: a distributed scheduler for IoT applications." *Cluster Computing* 20 (2017): 1775-1790.

Résumé :

Minimiser la latence et améliorer la qualité de service dans un environnement Fog Computing représentent des défis majeurs pour les applications temps réel comme la e-santé. Dans ce projet de fin d'études, nous avons étudié deux approches hybrides afin d'optimiser le placement des applications et équilibrer la charge entre les nœuds : la combinaison Weighted Round Robin (WRR) avec Least Connections (LC), ainsi que l'algorithme méta-heuristique Ant Colony Optimization (ACO) également associé à LC. L'objectif est de déterminer laquelle de ces approches offre les meilleures performances en termes de répartition des applications et de réactivité du système. Les simulations effectuées dans cette étude ont été réalisées à l'aide du simulateur iFogSim et les résultats obtenus sont jugés satisfaisants, montrant une amélioration significative en termes de latence et de temps de réponse.

Mots clés : Fog Computing, Équilibrage de charge ,WRR, ACO, LC, Réduction de la latence ,Qualité de Service ,iFogSim.

Abstract :

Minimizing latency and enhancing the QoS in a Fog Computing environment are major challenges for real-time applications such as e-health. In our graduation project, our work focuses on hybrid approaches to optimize the applications placement and load balancing between nodes: the combination Weighted Round Robin (WRR) with Least Connections (LC), and the metaheuristic algorithm Ant Colony Optimization (ACO) which is also associated with (LC). The objective is to determine which of these approaches offers the best performance in terms of the distribution of applications and the system's responsiveness. The simulations performed in this study were realized with the iFogSim simulator. The results obtained are considered satisfactory, showing a significant improvement in terms of latency and response time.

Keywords: Fog Computing, Load Balancing, WRR, ACO, LC, Latency reduction, QoS, iFogSim.

ملخص:

يعد تخفيض زمن الاستجابة وتحسين جودة الخدمة في بيئة الحوسبة الضبابية من التحديات الرئيسية لتطبيقات الوقت الحقيقي مثل الصحة الإلكترونية. في مشروع تخرجنا هذا، يرتكز عملنا على الاساليب الهجينة لتحسين وضع التطبيقات وموازنة توزيع الاحمال بين العقد: الجمع بين خوارزمية WRR مع LC، وخوارزمية التحسين بالاستدلال الفوقي ACO المرتبطة ايضا بLC .الهدف هو تحديد أي من هذه الأساليب يقدم أفضل أداء من حيث توزيع التطبيقات واستجابة النظام. تم إجراء المحاكاة في هذه الدراسة باستخدام محاكي iFogSim وتعتبر النتائج التي تم الحصول عليها مُرضية، حيث أظهرت تحسناً كبيراً من حيث زمن الاستجابة وزمن الاستجابة.

الكلمات المفتاحية: الحوسبة الضبابية، موازنة توزيع الاحمال،WRR، ACO، LC ، تقليل زمن الاستجابة، جودة الخدمة، iFogSim