REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

Faculté des Mathématiques et de l'Informatique

Département d'Informatique

N° :……

DOMAINE : Mathématiques et Informatique

FILIERE : Informatique

OPTION :

Mémoire présenté pour l'obtention Du diplôme de Master Académique

Par: HALITIM Iman

BEDOUHENE Wardia

Intitulé

Deployment of Fog Computing Devices in Mobile edge Computing environnment

Soutenu devant le jury composé de :

Dr. BRAHIMI Mahmoud	MCA	Université de M'sila	Président
Mr. SAYAD Lamri	MCA	Université de M'sila	Encadrant
Mr. DABBA Ali	MCB	Université de M'sila	Examinur

Année universitaire : 2021 / 2022

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MOHAMED BOUDIAF - M'SILA

Faculté des Mathématiques et de l'Informatique

Département d'Informatique

N° :……

DOMAINE : Mathématiques et Informatique

FILIERE : Informatique

OPTION :

Mémoire présenté pour l'obtention Du diplôme de Master Académique

Par: HALITIM Iman

BEDOUHENE Wardia

Intitulé

Deployment of Fog Computing Devices in Mobile edge Computing environnment

Soutenu devant le jury composé de :

Dr. BRAHIMI Mahmoud	MCA	Université de M'sila	Président
Mr. SAYAD Lamri	MCA	Université de M'sila	Encadrant
Mr. DABBA Ali	MCB	Université de M'sila	Examinur

Année universitaire : 2021 / 2022

Dédicace

Du profond de mon cœur je dédis ce mémoire à tous ceux qui sont chers .....

A mon très chère papa : à Lhomme de ma vie (BEDOUHENE Akli)

Tous les mots ne sauraient exprimer ma gratitude et ma reconnaissance pour ton dévouement et tes sacrifices, tu as toujours été a mes cotés pour me soutenir et m'épauler.

Je te dédie cette thèse, puisse tu y trouver le fruit de tes efforts.

A ma très chère maman : à ma raison de vivre (BOUCHERK Zahia)

A celle qui m'a tout donné sans attendre le moindre retour, a celle qui m'épaulait quand j'en avais le plus besoin.

Tu représentes le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cassé de m'encourager et de prier pour moi.

Ce modeste travail parait bien dérisoire pour traduire une reconnaissance infinie envers une mère aussi merveilleuse dont j'ai la fierté d'être la fille.

Je t'aime très fort

A mes chers frères et sœurs (Rayane, Amine, Wassila et Rania)

Merci d'être toujours à mes coté, par votre amour dévoué et votre tendresse, pour donner du gout et du sens à ma vie

Je pris Dieu, le tout puissant, pour qu'il vous donné bonheur et prospérité.

A Mon très cher fiancé (BACHTARZI Farid)

Merci pour ta confiance, ton soutien et surtout ton amour. Merci pour tous les bons moments de bonheur je suis la personne la plus chanceuse du monde, merci beaucoup d'avoir été la pour moi chaque fois que j'avais besoin de toi merci d'avoir complété ma vie .

A ma binôme Halitim Imane et sa grande famille

A mes très chères copines (Mouna, Fella, sabrina, khaoula, Karima et Kamilia)

Après tant d'années passées avec vous, vos n'es plus des amies, mais vous fais partie de ma famille, vous être l'amie de mon espoir, vous me rassemble, merci de ne m'avoir jamais laissé tomber, d'avoir d'être toujours présente.

Warda

Dédicace

Du profond de mon cœur je dédis ce mémoire à tous ceux qui sont chers .....

A mon très chère papa : à Lhomme de ma vie (HALITIM Ahmed)

Tous les mots ne sauraient exprimer ma gratitude et ma reconnaissance pour ton dévouement et tes sacrifices, tu as toujours été a mes cotés pour me soutenir et m'épauler.

Je te dédie cette thèse, puisse tu y trouver le fruit de tes efforts.

A ma très chère maman : à ma raison de vivre (Kasmi Saliha)

A celle qui m'a tout donné sans attendre le moindre retour, a celle qui m'épaulait quand j'en avais le plus besoin.

Tu représentes le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cassé de m'encourager et de prier pour moi.

Ce modeste travail parait bien dérisoire pour traduire une reconnaissance infinie envers une mère aussi merveilleuse dont j'ai la fierté d'être la fille.

Je t'aime très fort

A mes chers frères et sœurs (Fouad, Zaineb, Meriem et Roaya)

Merci d'être toujours à mes coté, par votre amour dévoué et votre tendresse, pour donner du gout et duens à ma vie

Je pris Dieu, le tout puissant, pour qu'il vous donné bonheur et prospérité

A Mon très cher fiancé (BENAMER Alaa Eddine)

Merci pour ta confiance, ton soutien et surtout ton amour .merci pour tous les bons moments de bonheur je suis la personne la plus chanceuse du monde, merci beaucoup d'avoir été la pour moi chaque fois que j'avais besoin de toi merci d'avoir complété ma vie

A ma binôme BEDOUHENE Warda et sa grande famille

A mes très chères copines (NARIMANE, FELLA, KARIMA, SABRINA, IMANE et MOUNA)

Après tant d'années passées avec vous, vos n'es plus des amies, mais vous fais partie de ma famille, vous être l'amie de mon espoir, vous me rassemble, merci de ne m'avoir jamais laissé tomber, d'avoir d'être toujours présente.

Imane

Remerciements

Au terme de Ce modeste travail, nous tenons tout d'abord à remercier le "BON DIEU" le tout puissant de nous avoir accordé le courage, la patience, la volonté et surtout la santé pour réaliser notre travail de recherche.

Nos très vifs remerciements vont au monsieur le président **Mr BRAHIMI Mahmoud** maitre de conférences à l'université de M'sila pour avoir accepté de présider le jury de soutenance.

Nos remerciements les plus sincères s'adressent à Monsieur **SAYAD Lamri**, pour avoir accepté de diriger ce travail ainsi que pour sa confiance, ses remarques, ses conseils et sa bienveillance.

Nous remercions particulièrement monsieur **DABBA Ali** enseignant chercheur et maitre assistant à l'université de M'sila, l'examinateur de ce travail, pour avoir accepté de juger ce travail.

Nous remercions également touts les professionnels de département informatique d'avoir pris de leur temps pour participer à notre enquête sur le terrain.

Enfin, nous tenons à remercier nos très chère parents de nous avoir poussés a faire ces études et d'avoir toujours été la pour nous et pour tout leurs sacrifices, leur interminable conseils, merci pour tout ce que vous ne cessez de faire pour moi, je vous aime.

Merci a tous ceux qui, directement ou indirectement m'ont aidé a mener à bien faire ce travail.

Table des matières

CHAPITRE 3:

CHAPITRE 4:

Liste des figures:

Liste des tableaux:

Liste des abbreviations:

IaaS: Infrastructure as a Service

PaaS: Platform as a Service

SaaS: Software as a Service

Web: World Wide Web

WMN: wireless mesh networks

FCS: Fog Computing System

CAAS: communication as a service

PC: Personal Computer

IOT: Internet of thing

Maas: Mobilité as a service.

IT : Information Technologie.

IP : Internet Protocol.

IDC: International Data Corporation

QOS: quality of service.

QOE: quality of experience

LAN: Local Area Network.

SDN: Software defined Networking

PSO: Particle swarm optimization.

LS: Local search

SINR: Signal to interference plus noise ratio.

RAN: Radio access network

FSG: Fog Smart Gareways

VM: Virtual Machine.

INTRODUCTION GENERALE:

Introduction générale

Cloud/Fog/Edge Computing et Internet des objets (IoT) jouent un rôle crucial dans l'Industrie 4.0. Généralement, dans une usine intelligente (ou autres applications industrielles définies dans un cadre géographique limité le serveur Cloud est maintenu dans un emplacement centralisé, entraînant un temps de latence trop long pour répondre aux demandes d'un grand nombre de capteurs IoT répartis dans une grande zone de travail. Le Fog Computing étend Cloud Computing avec la distribution de calcul, de stockage et ressources réseau à la périphérie du réseau pour prendre en charge les applications IoT avec des exigences de réponse en temps réel et à faible latence, reconnaissance de l'emplacement, mobilité de l'appareil final, évolutivité, hétérogénéité, stockage transitoire, diffusion de données à haut débit, calcul décentralisé, et sécurité . Ce travail porte sur le problème d'optimisation du déploiement d'appareils de calcul de Cloud pour prendre en charge les demandes des dispositifs de la périphérie mobile dans une zone géographique. Le déploiement de systèmes de Cloud a apparu dans plusieurs applications, par exemple, centre logistique intelligent, réseaux locaux, système médical cyber-physique, et réseau véhiculaire. Le déploiement d'autres réseaux ou systèmes a été largement étudié, par exemple, le déploiement de Cloudlets dans le sans fil réseaux métropolitains, déploiement de capteurs sans fil réseau dans un environnement 3D [9], et déploiement dynamique nœuds routeurs dans les réseaux maillés sans fil (WMN-dynRNP). Le WMN-dynRNP dans consiste à trouver le placement optimal De routeurs maillés pour couvrir les clients maillés pour s'adapter à une dynamique Environnement dans lequel les clients maillés peuvent dynamiquement et de manière autonome déplacez-vous vers n'importe quelle position dans la zone de placement et tournez sur ou hors de leur propre accès au réseau. Inspiré de ce travail, nous considérons le problème de déploiement dynamique d'un système de calcul de Cloud (FCS) composé de dispositifs de calcul de Cloud (dispositif de Cloud pour Court) et périphériques dans une zone de placement géographique dans laquelle chaque équipement est un appareil mobile (par exemple, Smartphone, Tablette, véhicule connecté, système embarqué…) et peut servir en tant que passerelle(c'est-à-dire qu'elle pourrait communiquer avec des capteurs IoT et d'autres appareils de périphérie statique via Bluetooth); les données collecté par les capteurs IoT sont transmises et prétraitées par un équipement du Fog, qui communique ensuite avec un équipement du Cloud pour traitement ultérieur ou en cours de relecture

sur le serveur Cloud. Il était courant dans les travaux précédents d'étendre l'optimisation problèmes dans une version pondérée pour refléter le degré d'importance de chaque entité concernée dans le système, par exemple, routage dans graphe du réseau de gares pondérées, chemin le plus court dans graphes pondérés par les nœuds ou pondérés par les bords, données IoT placement dans des infrastructures de Cloud basé sur une partition graphique pondérée , placement de la passerelle dans les WMN basé sur la pondération des nœuds graphiques et affectation des canaux dans les WLAN sur la base de graphes pondérés par les bords. Dans les applications FCS du monde réel, Certains appareils périphériques nécessitant un temps de réponse plus court pour répondre à leurs demandes (par exemple, dans une usine intelligente, ces appareils périphériques qui gèrent les données des composants cruciaux de la machine) ont des priorités élevées pour connecter le FCS. Le problème de placement FCS concerné est un problème d'établissement d'emplacement, qui s'est avéré NP-difficile. De plus, l'aspect dynamique du problème complique davantage sa résolution. Pour résoudre ce problème, ce travail propose une méthode inspirée des chauves-souris algorithme (BA), qui est un algorithme métaheuristique qui simule le comportement de recherche de nourriture des chauves-souris via l'écholocation pour trouver solutions suffisamment bonnes. À partir des résultats de simulation, l'algorithme BA choisi semble prometteur et est plus stable que la précédente approche OSP pour une dynamique non pondérée.

CHAPITRE 01

ClOUD COMPUTING

1.1 Définition

Le Cloud Computing ou informatique en nuage est une infrastructure dans laquelle la puissance de calcul et le stockage sont gérés par des serveurs distants auxquels les usagers se connectent via une liaison Internet sécurisée.
L'ordinateur de bureau ou portable, le téléphone mobile, la tablette tactile et autres objets connectés deviennent des points d'accès pour exécuter des applications ou consulter des données qui sont hébergées sur les serveurs. Le Cloud se caractérise également par sa souplesse qui permet aux fournisseurs d'adapter automatiquement la capacité de stockage et la puissance de calcul aux besoins des utilisateurs.
Pour le grand public, le Cloud Computing se matérialise notamment par les services de stockage et de partage de données numériques type Box, Dropbox, Microsoft OneDrive ou Apple iCloud sur lesquels les utilisateurs peuvent stocker des contenus personnels (photos, vidéos, musique, documents...) et y accéder n'importe où dans le monde depuis n'importe quel terminal connecté.

1.2 Historique

On appelle Cloud Computing un procédé consistant à transférer sur des serveurs situés à distance des fichiers ou des bases de données qui étaient auparavant conservés dans les serveurs locaux ou sur l'ordinateur personnel de l'utilisateur. Il permet notamment d'accéder sur demande à des fichiers informatisés utilisables par plusieurs personnes.
Un des principaux avantages du Cloud Computing est de permettre aux utilisateurs d'accéder à de nombreuses prestations accessibles en ligne sans devoir subir le poids d'une infrastructure coûteuse. Les données sont considérées comme étant stockées sur un « nuage » (d'où l'expression Cloud Computing)
L'idée du Cloud Computing a pris naissance en 1990 et surtout en 1991 avec la naissance d'Internet et la mise sur le marché du logiciel CERN qui a été le premier logiciel accessible par le Web. Ensuite, l'idée a progressé avec l'apparition de nouvelles solutions IT le lancement du navigateur Mosaic en 1993

et celui du navigateur Netscape en 1994. En 1995, la découverte d'EBay et d'Amazon a encore accéléré le processus. Enfin, le lancement en 1996 de l'assistant Palm PDA a donné une nouvelle impulsion.

En 2000, l'arrivée de Google a favorisé une réelle émergence du Cloud Computing

Dans l'avenir, le Cloud Computing devrait connaître une belle croissance car moins de 10% des sociétés y avaient recours en 2009. On prévoit que 30 à 50 % des sociétés devraient utiliser ce service bien que les entreprises semblent préférer le Cloud privé. Il se peut que l'arrivée de nouvelles solutions IT provoque une accélération du phénomène.

1.3 Les services du Cloud Computing :

On distingue plusieurs types de services Cloud :

- IaaS (**Infrastructure as a Service**) : le système d'exploitation et les applications sont installés par les clients sur des serveurs auxquels ils se connectent pour travailler comme s'il s'agissait d'un ordinateur classique.

- PaaS (**Platform as a Service**) : dans ce mode, c'est le fournisseur du service Cloud qui administre le système d'exploitation et ses outils. Le client peut installer ses propres applications si besoin.

- SaaS (**Software as a Service**) : les applications sont fournies sous forme de services clés en mains auxquels les utilisateurs se connectent via des logiciels dédiés ou un navigateur Internet. Pour le grand public, il s'agit par exemple de messageries électroniques type Gmail, Yahoo, Outlook.com ou de suites bureautiques type Office 365 ou Google Apps.

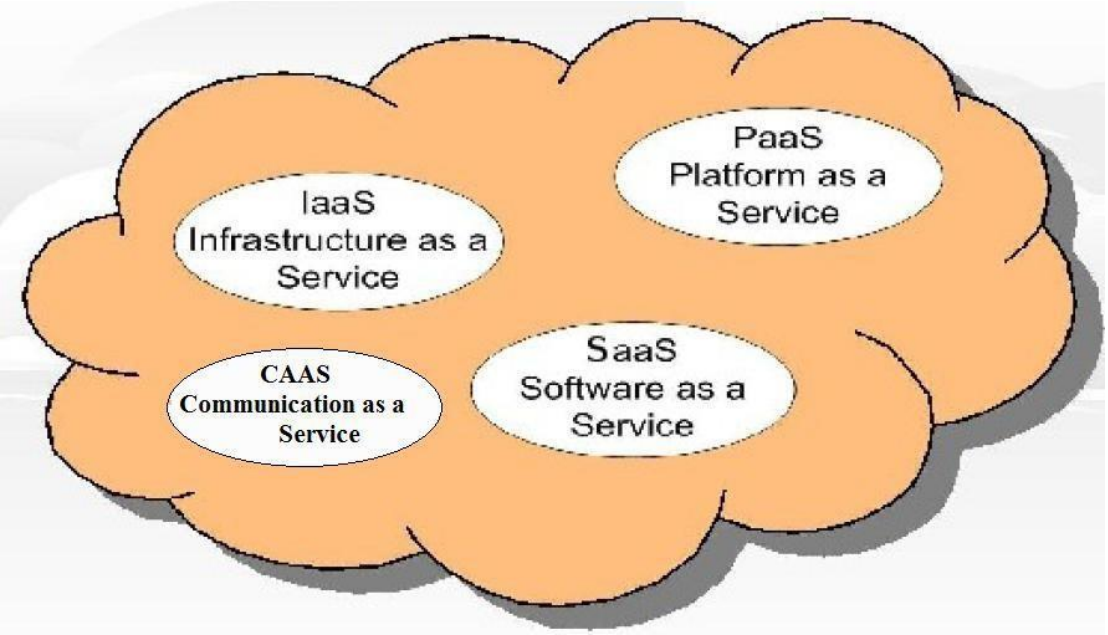

Fig 1.(Terminologies de Cloud Computing)

1.3 A quoi sert le Cloud Computing ?

Le Cloud Computing offre plusieurs avantages considérables aux entreprises, en particulier le mode SaaS. Il leur permet de délocaliser leur matériel, leurs données et leurs applications sur des serveurs dans des espaces spécifiques (les centres de données).

En confiant ces tâches à un prestataire de confiance, ils gagnent de l'espace dans leurs locaux (plus besoin de stocker des serveurs) et du temps (l'architecture et le stockage des données sont pris en charge par le fournisseur).

Autre caractéristique fondamentale qui donne à l'informatique dans les nuages tout son sens : les données conservées sont accessibles depuis n'importe où par les personnes qui sont autorisées à y accéder. Une aubaine dans le cadre du travail collaboratif.

Si vos collaborateurs sont basés à Madrid, Pékin, New York et Kinshasa, ils pourront tous accéder au portail de votre entreprise depuis leur Smartphone ou leur ordinateur, insérer leur mot de passe et consulter le compte-rendu de votre dernière réunion, partager leurs avis et laisser des commentaires en temps réel. Cette belle prouesse de l'informatique dans les nuages porte un nom: la synchronisation des postes de travail.

1.4 **Comment fonctionne le Cloud Computing ?**

Vous souscrivez une formule et signez un contrat d'externalisation de votre architecture informatique avec un fournisseur de solutions de Cloud Computing (abonnement mensuel ou annuel). Toutes les données de votre entreprise sont envoyées vers des serveurs distants, dans un centre de stockage. Pour y accéder, vous vous connectez à Internet depuis votre PC ou votre Smartphone (authentification requise).

Le fonctionnement et les prestations fournies varient selon le type de service souscrit (IaaS, PaaS, SaaS). Les machines virtuelles du fournisseur opèrent dans des clusters. Plusieurs dispositifs de sécurité (notamment des mécanismes de redondance) permettent d'éviter une interruption du service pouvant engendrer une perte des données.

La délégation de toutes ces tâches nécessitant des connaissances en informatique à un professionnel du Cloud offre la possibilité aux entreprises de se consacrer pleinement à leur cœur de métier tout en garantissant une protection maximale de leurs données etapplications (y compris la messagerie).

2.FOG COMPUTING

2.1 Définition

Le Fog Computing peut être définit comme l'ensemble de dispositifs informatiques de traitement et de stockage de données, servant de rapprocher le Cloud Computing des appareils qui produisent et agissent sur les données IoT.

Ces dispositifs, appelés Fog Nodes, peuvent être déployés n'importe où avec une connexion réseau : en usine, au sommet d'un poteau électrique, le long d'une voie ferrée, dans un véhicule ou sur une plate-forme pétrolière.

Tout équipement informatique avec une capacité de traitement et de stockage de données peut être un Fog Node. Les exemples incluent les Switches Ethernet, les routeurs, les serveurs intégrés et les caméras de vidéosurveillance.

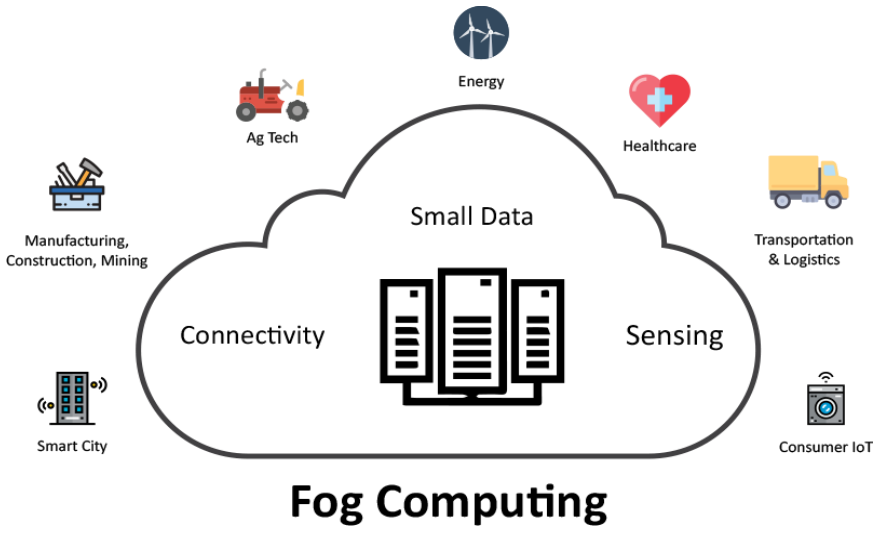

Fig 2.(Fog Computing)

2.2 Le plus du FOG

Face à l'explosion des volumes de données issus des objets connectés, les infrastructures Actuelles de Cloud, très centralisées, pourraient atteindre leur limite. Les capacités des réseaux et des Datacenters existants risquent en effet la saturation, ce qui ferait exploser les temps de latence et rendrait le traitement des données incompatible avec les besoins d'analyse en temps réel. Avec le Cloud, les serveurs capables de traiter les données sont distants et leur temps de réponse dépend de facteurs que l'utilisateur ne peut pas contrôler : charge de travail des serveurs, encombrement du réseau. . . Le fog Computing a pour but d'apporter une réponse à cette problématique. Il réduit le chemin à parcourir entre l'appareil qui produit les données et l'équipement qui les traite, et la nécessité de faire transiter les données vers les serveurs.

2.3 Consortium de l'OPENFOG

Cette collaboration a pour but de fédérer l'entreprise impliquée dans le FOG autour de la définition de protocoles et standards technologies spécifiques au FOG afin de standardiser le marché.

L'OpenFog a défini une infrastructure pour mettre en œuvre une solution de FaaS (Fog as a Service). Celle-ci se décline en plusieurs couches calquées sur la logique d'empilement du Cloud (IaaS, PaaS et SaaS). Une architecture à laquelle s'ajoutent des éléments spécifiques au FOG Computing : des services d'interconnexion réseau, de collecte de données à partir des objets connectés, ou encore des applications de traitement. Le principe de fonctionnement est ainsi le même que celui du Cloud avec quelques particularités.

2.4 Comment fonctionne le Fog Computing ?

Les ingénieurs développent des applications IoT pour les Fog Nodes à la périphérie de réseau.

Les Fog Nodes les plus proches de la périphérie de réseau ingèrent les données des appareils IoT. Ensuite, l'application Fog IoT du client dirige les différents types de données vers l'endroit optimal pour être analysées : comme le montre le tableau ci-

dessous :

- Les données les plus urgentes sont analysées sur le Fog Node le plus proche des appareils IoT. Dans un réseau électrique intelligent, par exemple, l'exigence la plus urgente est de vérifier que les boucles de protection et de contrôle fonctionnent correctement. Par conséquent, les Fog Nodes les plus proches des appareils IoT de terrain peuvent rechercher des signes de problèmes, puis les prévenir en envoyant des commandes de contrôle aux actionneurs.
- Les données qui peuvent attendre quelques secondes ou minutes pour une action sont transmises à un Node d'agrégation pour être analysées. Dans l'exemple Smart Grid, chaque sous-station électrique peut avoir son propre Node d'agrégation qui signale l'état opérationnel de chaque alimentation électrique en aval et latéral.
- Les données moins urgentes sont envoyées vers le Cloud Computing pour une analyse ultérieure l'analyse historique. Par exemple, chacun des milliers ou centaines de Fog Nodes peut envoyer des résumés périodiques des données au Cloud pour être stockées et analysées ultérieurement.

2.5 Pourquoi utiliser le Fog Computing ?

Le rapprochement du Cloud Computing des appareils IoT qui génèrent et agissent sur les données profite à l'entreprise des manières suivantes :

- **Une plus grande agilité commerciale** : Avec les bons outils, les ingénieurs peuvent développer rapidement des applications Fog Computing et les déployer là où cela est nécessaire. Les fabricants de machines peuvent proposer le MaaS à leurs clients. Les applications Fog Computing programment la machine pour qu'elle fonctionne selon les besoins de chaque client.
- **Meilleure sécurité** : Il est possible de protéger vos Fog Nodes en utilisant la même politique, les mêmes contrôles et procédures que vous utilisez dans d'autres parties de votre environnement informatique.
- **Des informations plus approfondies** : Le traitement des données sensibles se fera localement au lieu de les envoyer dans le Cloud pour l'analyse. Ainsi, votre équipe informatique pourra surveiller et contrôler les appareils qui collectent, analysent et stockent les données.
- **Frais d'exploitation réduits** : économiser la bande passante du réseau en traitant les données sélectionnées localement au lieu de les envoyer vers le Cloud pour le processus d'analyse.

2.6 Différence entre le FOG et le CLOUD

L'edge Computing se réfère à tous les équipements IT d'"extrémité", c'est-à-dire installés
au plus près des utilisateurs et sources de données. Ces équipements informatiques et réseau peuvent faire partie d'une infrastructure de fog Computing. Ils ne se limitent
Pas seulement aux équipements IT conventionnels. Les objets connectés capables de stocker et de traiter les données en local sont aussi considérés comme faisant partie d'une Infrastructure d'edge Computing.

Caractéristique	**Cloud Computing**	**Informatique de brume**
Latence	Le Cloud Computing a une latence élevée par rapport au fog Computing	Le Fog Computing a une faible latence
Capacité	Le Cloud Computing n'offre aucune réduction	Le Fog Computing réduit la quantité de données envoyées au Cloud

Caractéristique	Cloud Computing	Informatique de brume
	des données lors de l'envo	Computing.
Réactivité	Le temps de réponse du système est faible.	Le temps de réponse du système est élevé.
Sécurité	Le Cloud Computing à moins de sécurité que le Fog Computing	L'informatique de Cloud a une sécurité élevée.
La vitesse	La vitesse d'accès est élevée en fonction de la connectivité de la VM.	Élevé encore plus par rapport au Cloud Computing.
Intégration de données	Plusieurs sources de données peuvent être intégrées.	Plusieurs sources de données et appareils peuvent être intégrés.
Mobilité	Dans le Cloud Computing, la mobilité est limitée.	La mobilité est prise en charge dans le fog Computing.
Sensibilisation à l'emplacement	Partiellement pris en charge dans le Cloud Computing.	Pris en charge dans l'informatique de Cloud.
Nombre de nœuds de serveur	Le Cloud Computing a peu de nœuds de serveur.	Fog Computing a un grand nombre de nœuds de serveur.
Distribution	Il est centralisé.	Il est décentralisé et distribué.
Lieu de service	Services fournis sur Internet.	Services fournis à la périphérie du réseau local.

Caractéristique	Cloud Computing	Informatique de brume
Environnement de travail	Bâtiment de Datacenter spécifique avec systèmes de climatisation	Extérieur (rues, stations de base, etc.) ou intérieur (maisons, cafés, etc.)
Mode de communication	Réseau IP	Communication sans fil : WLAN, Wifi, 3G, 4G, ZigBee, etc. ou communication filaire (faisant partie des réseaux IP)
Dépendance à la qualité du cœur de réseau	Nécessite un cœur de réseau solide.	Peut également fonctionner dans un noyau de réseau faible.

Tableau 1: caractéristiques du Cloud et fog Computing

2.7 Avantages du FOG

La technologie Fog Computing présente des avantages considérables par rapport au Cloud Computing. De nombreuses plates-formes IoT tirent plus d'avantages du Fog que du Cloud.

- Traiter vos données en (quasi) tempsréel
- Décongestionner votre trafic réseau
- Renforcer la sécurité de vosdonnées
- Quantité minimale de données envoyée au Cloud.
- Économiser la bande passante
- Réduire la latence des données
- Améliorer la sécurité des données
- Traitement immédiat des données

2.8 System architecture

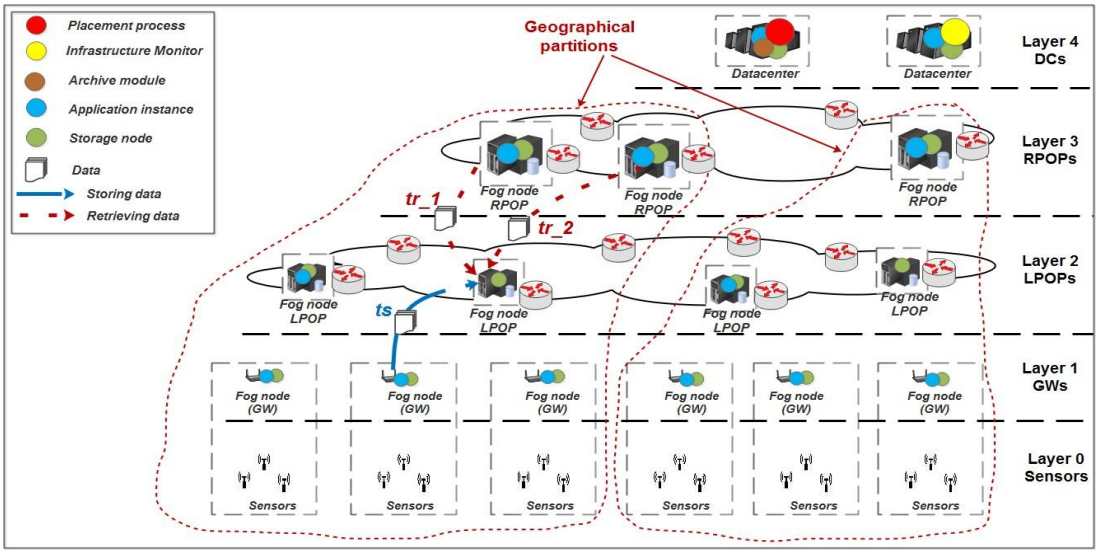

Fig 3(System architecture.(

3.Edge Computing

3.1 Définition

L'edge Computing désigne une architecture informatique distribuée qui se caractérise par une puissance de traitement décentralisée. Concrètement, l'edge Computing permet de traiter des données de façon directe par le périphérique qui les produit, ou par un ordinateur local. Il n'est dans ce cas plus nécessaire de transmettre les données à un Datacenter distant pour les analyser. L'IDC (International Data Corporation) définit de son côté l'edge Computing comme un réseau maillé de micro Datacenter capables de traiter ou de stocker des données localement. On retrouve cette technologie principalement dans le domaine de l'IoT, où elle vient concurrencer le Cloud Computing.

Plus précisément, selon la définition d'IDC, le Edge Computing peut être considéré comme un réseau maillé de Micro Data Centers qui traitent ou stockent les données critiques localement. Les données sont ensuite transmises vers un Data Center central ou un stockage Cloud avec une empreinte de moins de 10 mètres carrés.

Le plus souvent, l'Edge Computing est utilisé dans le domaine de l'internet des objets. Une partie des nombreuses données collectées par les appareils connectés sont traitées localement, afin de réduire le trafic en direction du Cloud ou des Data Centers et de permettre une analyse des données importantes en temps réel (ou presque).

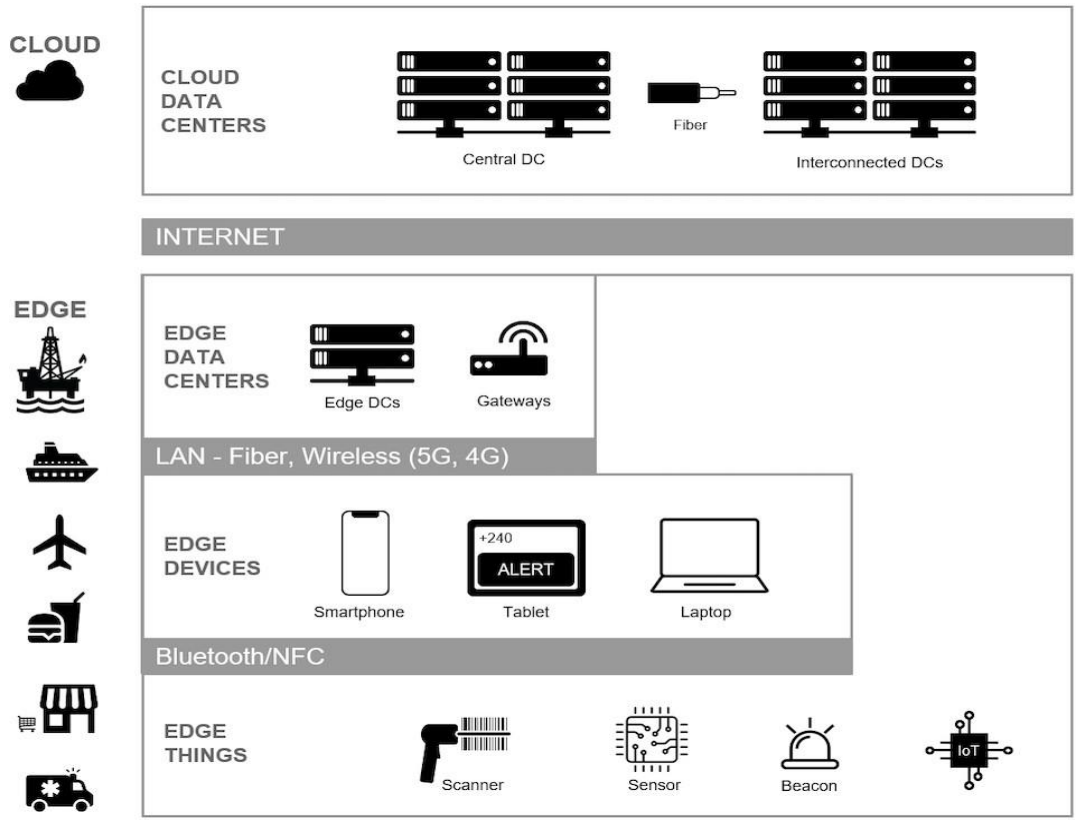

Fig 4:(edge Computing)

3.2 Utilisation de l'edge Computing

L'edge Computing permet d'étendre un environnement uniforme, depuis le Datacenter central jusqu'aux emplacements physiques proches des utilisateurs et des données, situés en périphérie du réseau. Tout comme une stratégie de Cloud hybride permet aux entreprises d'exécuter les mêmes charges de travail à la fois dans leurs propres Datacenter et sur une infrastructure de Cloud public (telle que Amazon Web Services, Microsoft Azure ou Google Cloud), une stratégie d'edge Computing étend un environnement Cloud à de nombreux autres emplacements. L'edge Computing est désormais utilisé dans de nombreux secteurs d'activité, notamment les télécommunications, la fabrication, les transports et les services publics. Les raisons qui incitent les entreprises à mettre en œuvre l'edge Computing sont aussi variées que les cas d'utilisation.

3.3 Exemples de cas d'utilisation de l'edge Computing

De nombreux cas d'utilisation de l'edge Computing sont liés à la nécessité de traiter les données localement en temps réel, lorsque la transmission des données vers un Datacenter en vue de leur traitement génère des niveaux de latence inacceptables. Prenons l'exemple d'une usine de fabrication moderne. Les capteurs de l'Internet des objets (IoT) génèrent un flux continu de données qui peuvent être utilisées pour éviter des interruptions et améliorer l'exploitation. D'après les estimations, une usine

moderne qui compte 2 000 équipements peut générer 2 200 téraoctets de données par mois. Le de cette mine de données près de l'équipement est plus rapide et moins onéreux que lorsqu'il faut d'abord les transmettre à un Datacenter distant. Toutefois, il reste préférable d'utiliser une plateforme de données centralisée pour faire le lien avec l'équipement. De cette façon, l'équipement peut, par exemple, recevoir des mises à jour logicielles standardisées et partager des données filtrées qui permettront d'améliorer l'exploitation dans d'autres sites de l'usine.

Autre exemple courant d'utilisation de l'edge Computing : les véhicules connectés. Les bus et les trains sont équipés d'ordinateurs qui permettent de suivre le flux de passagers et la prestation de services. Les livreurs peuvent trouver les trajets les plus rapides grâce à la technologie embarquée dans leur véhicule. Lorsqu'une stratégie d'edge Computing est mise en œuvre, chaque véhicule exécute la même plateforme standardisée que le reste de la flotte, ce qui rend les services plus fiables et permet de protéger les données de manière uniforme.

Viennent ensuite les véhicules autonomes, un autre exemple d'edge Computing qui implique le traitement d'un gros volume de données en temps réel dans une situation où la connexion au réseau peut être instable. En raison de ce volume, les véhicules autonomes, tels que les voitures sans conducteur, traitent les données des capteurs à bord du véhicule afin de réduire la latence. Ils peuvent cependant toujours se connecter à distance à un point central pour recevoir des mises à jour logicielles.

L'edge Computing permet aussi de maintenir la rapidité d'exécution des services internet les plus utilisés. Les réseaux de distribution de contenu déploient des serveurs de données géographiquement proches des utilisateurs, ce qui permet un chargement rapide des sites web très sollicités et la prise en charge de services de diffusion vidéo rapides.

Dernier exemple d'edge Computing : les antennes 5G à proximité. Les opérateurs de télécommunications exécutent de plus en plus leurs réseaux en recourant à la

virtualisation des fonctions réseau, à l'aide de machines virtuelles qui fonctionnent sur du matériel standard à la périphérie du réseau. Ces machines virtuelles peuvent remplacer des équipements propriétaires coûteux. Avec une stratégie d'edge Computing, les opérateurs peuvent continuer d'exécuter de manière cohérente les logiciels sur des dizaines de milliers de sites distants, conformément à des normes de sécurité uniformes. En exécutant les applications au plus près de l'utilisateur final dans un réseau mobile, les opérateurs peuvent aussi réduire la latence et offrir de nouveaux services.

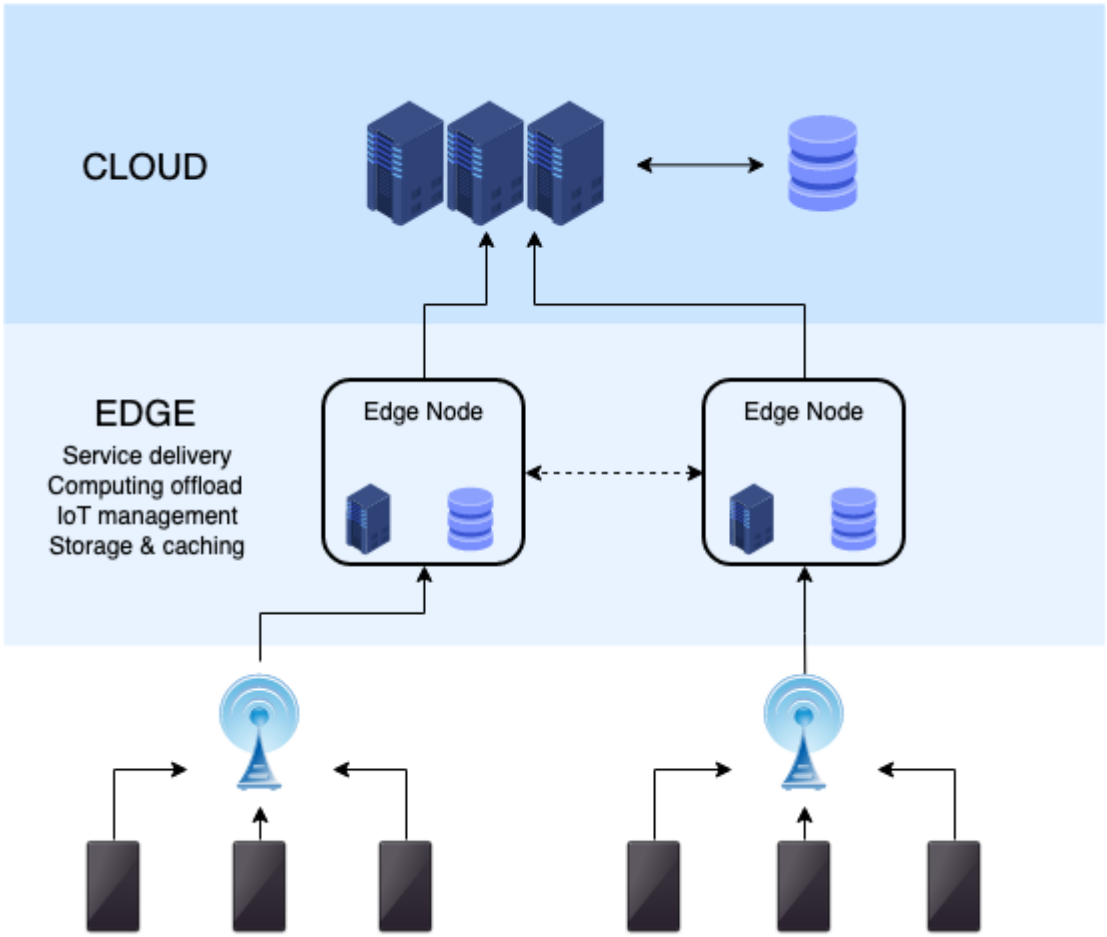

fig5:(l'utilisation de edge Computing)

3.4 Avantage et inconvénient de l'Edge Computing

L'Edge Computing permet d'analyser, de traiter et de transférer des données à la périphérie d'un réseau. Cela signifie que les données sont analysées localement, plus près de l'endroit où elles sont stockées, en temps réel sans latence. L'informatique en périphérie permet d'analyser les données de l'appareil Internet des objets en périphérie du réseau avant d'être envoyées vers un centre de données ou un Cloud.

Avantages :

- Il offre une meilleure fiabilité à haute vitesse et à latence réduite, ce qui permet un traitement des données et une livraison de contenu plus rapides.
- Il offre une meilleure sécurité en répartissant le traitement, le stockage et les applications sur un large éventail d'appareils et de centres de données, ce qui rend difficile la suppression du réseau par une seule interruption.
- Il offre une voie beaucoup moins coûteuse vers l'évolutivité et la polyvalence, permettant aux entreprises d'étendre leur capacité de calcul grâce à une combinaison d'appareils IoT et de centres de données de périphérie.
-

Inconvénients :

- Il nécessite plus de capacité de stockage.
- Les défis de sécurité dans l'informatique de périphérie sont élevés en raison de l'énorme quantité de données.
- Il analyse uniquement les données.
- Le coût de l'informatique de pointe est très élevé.
- Cela nécessite une infrastructure avancée.

3.5 Éléments d'un réseau d'edge Computing

L'edge Computing peut être représenté comme une série de cercles dont le centre est le Datacenter principal. Chaque cercle correspond à un niveau différent qui se rapproche de la périphérie la plus éloignée.

• **Cœur de l'entreprise :** il s'agit de niveaux classiques « hors périphérie », détenus et exploités par des fournisseurs de Cloud public, des opérateurs de télécommunications ou de grandes entreprises.

• **Périphérie des fournisseurs :** il s'agit de niveaux situés entre les Datacenter centraux ou régionaux et l'infrastructure du dernier kilomètre, généralement détenus et exploités par un opérateur de télécommunications ou un fournisseur d'accès à Internet et à partir desquels ce fournisseur dessert plusieurs clients.

• **Périphérie des utilisateurs :** ces niveaux côté utilisateurs finaux de l'infrastructure du dernier kilomètre peuvent inclure la périphérie de l'entreprise (par exemple, un magasin, une usine, un train) ou la périphérie des consommateurs (par exemple, une résidence privée, une voiture).

• **Périphérie des appareils :** il s'agit des systèmes autonomes (et hors cluster) qui connectent directement les capteurs/actionneurs par le biais de protocoles autres qu'internet. C'est la périphérie la plus éloignée du réseau.

3.6 Edge Computing et Cloud Computing :

Le Cloud Computing consiste à exécuter des charges de travail dans des Clouds, des environnements qui dissocient, regroupent et partagent des ressources évolutives sur un réseau.

Le Cloud Computing s'est longtemps axé sur la centralisation des services Cloud dans une poignée de grands Datacenter. Cette centralisation permettait de bénéficier de ressources hautement évolutives et plus faciles à partager, sans perte de contrôle sur la sécurité de l'entreprise.

L'edge Computing couvre les cas où la centralisation ne peut être mise en œuvre en raison de contraintes de réseau ou d'autres limites.

En outre, une stratégie de Cloud Computing qui vise à exécuter les logiciels dans des conteneurs vient compléter le modèle d'edge Computing. Grâce aux conteneurs, les applications deviennent portables, ce qui permet aux entreprises de les exécuter là où elles sont le plus utiles. Avec une stratégie de conteneurisation, les entreprises ont la possibilité de déplacer les applications du Datacenter vers la périphérie, ou inversement, avec de faibles conséquences sur l'exploitation.

3.7 Exemples de cas d'utilisation de l'edge Computing

De nombreux cas d'utilisation de l'edge Computing sont liés à la nécessité de traiter les données localement en temps réel, lorsque la transmission des données vers un Datacenter en vue de leur traitement génère des niveaux de latence inacceptables.

Prenons l'exemple d'une usine de fabrication moderne. Les capteurs de l'Internet des objets (IoT) génèrent un flux continu de données qui peuvent être utilisées pour éviter des interruptions et améliorer l'exploitation. D'après les estimations, une usine moderne qui compte 2 000 équipements peut générer 2 200 téraoctets de données par mois. Le traitement de cette mine de données près de l'équipement est plus rapide et moins onéreux que lorsqu'il faut d'abord les transmettre à un Datacenter distant. Toutefois, il reste préférable d'utiliser une plateforme de données centralisée pour faire le lien avec l'équipement. De cette façon, l'équipement peut, par exemple, recevoir des mises à jour logicielles standardisées et partager des données filtrées qui permettront d'améliorer l'exploitation dans d'autres sites de l'usine.

Autre exemple courant d'utilisation de l'edge Computing : les véhicules connectés. Les bus et les trains sont équipés d'ordinateurs qui permettent de suivre le flux de passagers et la prestation de services. Les livreurs peuvent trouver les trajets les plus rapides grâce à la technologie embarquée dans leur véhicule. Lorsqu'une stratégie d'edge Computing est mise en œuvre, chaque véhicule exécute la même plateforme standardisée que le reste de la flotte, ce qui rend les services plus fiables et permet de protéger les données de manière uniforme.

Viennent ensuite les véhicules autonomes, un autre exemple d'edge Computing qui implique le traitement d'un gros volume de données en temps réel dans une

situation où la connexion au réseau peut être instable. En raison de ce volume, les véhicules autonomes, tels que les voitures sans conducteur, traitent les données des capteurs à bord du véhicule afin de réduire la latence. Ils peuvent cependant toujours se connecter à distance à un point central pour recevoir des mises à jour logicielles.
L'edge Computing permet aussi de maintenir la rapidité d'exécution des services internet les plus utilisés. Les réseaux de distribution de contenu déploient des serveurs de données géographiquement proches des utilisateurs, ce qui permet un chargement rapide des sites web très sollicités et la prise en charge de services de diffusion vidéo rapides.

Dernier exemple d'edge Computing : les antennes 5G à proximité. Les opérateurs de télécommunications exécutent de plus en plus leurs réseaux en recourant à la virtualisation des fonctions réseau, à l'aide de machines virtuelles qui fonctionnent sur du matériel standard à la périphérie du réseau. Ces machines virtuelles peuvent remplacer des équipements propriétaires coûteux.
Avec une stratégie d'edge Computing, les opérateurs peuvent continuer d'exécuter de manière cohérente les logiciels sur des dizaines de milliers de sites distants, conformément à des normes de sécurité uniformes. En exécutant les applications au plus près de l'utilisateur final dans un réseau mobile, les opérateurs peuvent aussi réduire la latence et offrir de nouveaux services.

3.8 IoT et appareils d'edge Computing

L'IoT, ou Internet des objets, désigne le processus de connexion d'objets physiques à Internet, des objets du quotidien tels que les ampoules, aux dispositifs médicaux, appareils portables, appareils intelligents ou encore feux de circulation routière dans les villes intelligentes.
Les appareils d'IoT ne sont pas nécessairement des appareils d'edge Computing. Ces appareils connectés font cependant partie des stratégies d'edge Computing de nombreuses entreprises. L'edge Computing permet de renforcer la puissance de calcul en périphérie d'un réseau IoT, afin de réduire la latence des communications entre les appareils d'IoT et les réseaux informatiques centraux auxquels ceux-ci sont connectés.
Si le simple fait de pouvoir envoyer ou recevoir des données a marqué

l'avènement de l'IoT, la possibilité d'envoyer, de recevoir et d'analyser des données à l'aide d'applications IoT est une approche plus moderne que l'edge Computing a rendue possible

3.9 Fog Computing et lien avec l'edge Computing

Le fog Computing désigne les opérations informatiques réalisées dans des emplacements physiques distribués, plus proches des utilisateurs et des sources de données.

C'est un synonyme d'edge Computing. La différence entre ces deux expressions est uniquement terminologique.

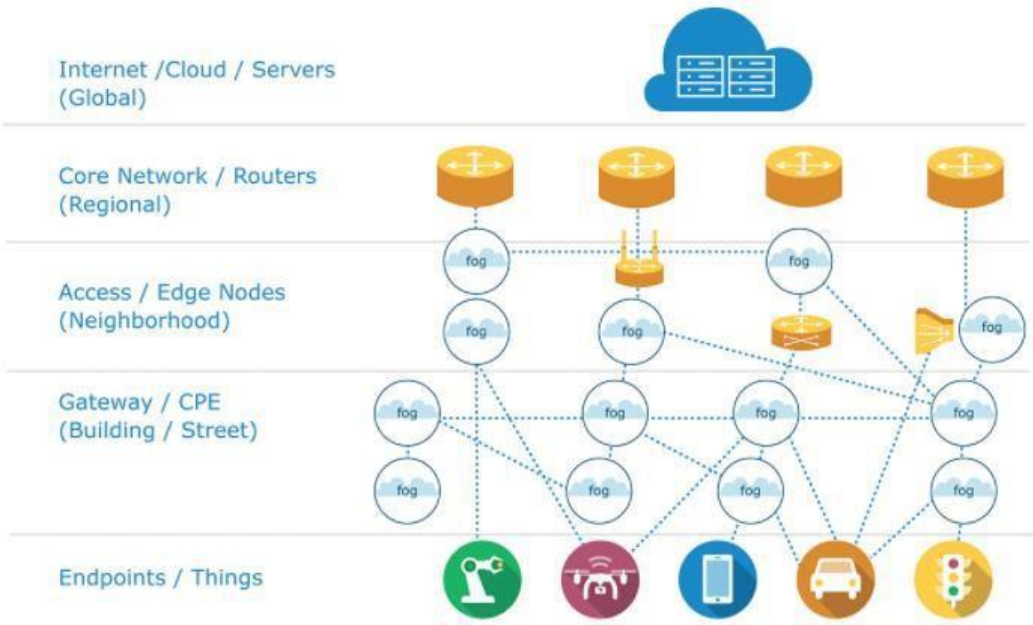

Fig 6 :(Comprendre le fog Computing en sept questions)

3.10 Fog VS edge Computing

Le fog Computing et l'edge Computing sont deux infrastructures assez similaires. Toutes deux sont basées sur le traitement des données produites directement par les objets connectés en périphérie de réseau. Cette proximité avec le point d'origine mène à une réduction importante du temps de latence (plus de trajet entre l'objet connecté et le Cloud). La différence entre fog et edge Computing se trouve les équipements informatiques impliqués : selon l'OpenFog Consortium, l'edge fait référence aux terminaux de traitement tandis que le fog Computing renvoie à l'architecture IT. Par ailleurs, on parle d'edge Computing quand les ressources de calcul se trouvent dans l'objet connecté, et de fog Computing quand elles sont dans un nœud de réseau séparé, comme une passerelle IoT.

3.11 Éléments d'un réseau d'edge Computing

L'edge Computing peut être représenté comme une série de cercles dont le centre est le Datacenter principal. Chaque cercle correspond à un niveau différent qui se rapproche de la périphérie la plus éloignée.

- **Cœur de l'entreprise :** il s'agit de niveaux classiques « hors périphérie », détenus et exploités par des fournisseurs de Cloud public, des opérateurs de télécommunications ou de grandes entreprises.

  Périphérie des fournisseurs : il s'agit de niveaux situés entre les Datacenter centraux ou régionaux et l'infrastructure du dernier kilomètre, généralement détenus et exploités par un opérateur de télécommunications ou un fournisseur d'accès à Internet et à partir desquels ce fournisseur dessert plusieurs clients.
- **Périphérie des utilisateurs :** ces niveaux côté utilisateurs finaux de l'infrastructure du dernier kilomètre peuvent inclure la périphérie de l'entreprise (par exemple, un magasin, une usine, un train) ou la périphérie des consommateurs (par exemple, une résidence privée, une voiture).
- **Périphérie des appareils :** il s'agit des systèmes autonomes (et hors cluster) qui connectent directement les capteurs/actionneurs par le biais de protocoles autres qu'internet. C'est la périphérie la plus éloignée du réseau.

Chapitre 02

CHAPITRE II

1. Introduction:

L'Internet des objets (IoT) représente un chambardement responsable pour les situations le management. La gérance de la situation en période sensibilité est associée à objets distribués et à elles capteurs intelligents associés. Intelligent les situations des capteurs doivent participer stockées et récupérées utilement sur supplique de travaux IoT. Les appareils IoT se développent sans tarder et on prévoit qu'pour ainsi dire 50 milliards d'appareils seront déployés en 2020. La modèle Cloud existante consenti des travaux dans une embarrassante pluie de situation. Mais pour nous scénarios, il peut solliciter des limites en réfléchie de l'inflation du grimage sur l'bande du entrecroisement empressé un atermoiement pour la rétribution des travaux. Différents travaux IoT textuels que Soins de santé, aveu faciale, militaire, gérance des catastrophes nécessitent une bref en période sensibilité rebrousse-poil une formidablement faiblard latence. Surpasser ce problème, un récit assemblage doit participer proposée et ainsi, l'télétraitement de grésil a émergé dans hausser ces défis. Le Fog Computing est une idée qui consenti des travaux au bordure du entrecroisement et implique des passerelles intelligentes nommées Fog Smart Passerelles (FSG). Les nœuds de grésil sont déployés pour le entrecroisement à familiarité les utilisateurs dans conseiller les travaux. Dans cette assemblage la situation sont traitées régulièrement endroit d'participer envoyées pour le Cloud. Le mises et défis majeurs du achèvement d'assemblage dans edgecentric Les travaux IoT découvrent les nœuds de grésil, la risque en enfui des situations, partitionnement. Le Fog Computing ou Fog Networking est un assemblage virtualisée dans le calcul, le stockage, le planning familial et un entrecroisement qui faveur ces travaux surtout imminent des utilisateurs finaux.

2. CONTEXTE :

Les études précédentes abordent le placement des serveurs de périphérie sous quatre angles : (1) minimiser les temps de latence, qui sont approximés à l'aide d'un certain nombre de mesures de distance différentes ; (2) minimiser les coûts de déploiement des serveurs tout en limitant le temps de latence maximal ; (3) optimiser le compromis entre le temps de latence et les coûts de déploiement ; ou (4) maximiser les connexions des

utilisateurs, c'est-à-dire la couverture, au sein des grappes. Les études considèrent différents ensembles de paramètres fonctionnels pour le placement, tels que le nombre de serveurs, la capacité individuelle des serveurs, la géolocalisation des serveurs et les priorités de localisation. Des propriétés non fonctionnelles de la qualité de service et de l'expérience utilisateur, telles que la fiabilité, ont également été utilisées comme paramètres dans les algorithmes de placement étudiés.

Les études ont appliqué diverses approches pour le placement : k-means [34], k-means avec programmation quadratique en nombres entiers mixtes [21], regroupement basé sur la densité [23], théorie des graphes comme dans un problème d'ensemble dominant minimum [60], structures hiérarchiques arborescentes [24,49], optimisation par contraintes multi-objectifs [5,14,53], la programmation linéaire mixte en nombres entiers [9,17], la programmation linéaire en nombres entiers avec des solutions heuristiques [31,54,57], la programmation mixte en nombres entiers non-linéaire [41], le regroupement combiné à l'optimisation comme un problème de localisation d'installations [40], et avec des algorithmes heuristiques [37,58].

En comparaison, PACK place un nombre fixe de serveurs tout en minimisant les latences entre les utilisateurs et les serveurs périphériques, en équilibrant la charge de travail du système et en satisfaisant les limites inférieure et supérieure de la capacité des serveurs. PACK peut être vu comme une variante du problème de regroupement de type k-means avec des contraintes de capacité, résolu ici avec un algorithme de descente de coordonnées par blocs avec des étapes de programmation en nombres entiers.

Article	**Charge de travail**	**Contraintes de capacité**	**Mesure de distance**	**Rôle de la distance**	**Nombre de serveurs**	**Association de serveurs**
Bhatta et al. [5]	Workload	Upper	Netw. topology	thres. & min.	minimized	Hard
Bouet et al. [9]	phone calls	Upper	Geospatial	not used	on-demand	Hard
Chen et al. [11]	simulated	Upper	Netw. topology	thresh. & min.	fixed	request
Cui et al. [13]	total no. of requests	not used	Geospatial	thresholded	fixed	Hard
da Silva et al. [14]	time series	threshold	Geospatial	not directly used	on-demand	request
Fan et al. [17]	Requests	Upper	Netw. topology	minimized	on-demand	fractional
Gedeon et al. [19]	simulated	Upper	netw. topology	thresholded	fixed	request
Guan et al. [20]	Peak	threshold	Geospatial	minimized	on-demand	Hard
Guo et al. [21]	total requests	Balanced	netw. topology	minimized	fixed	Hard
Jia et al. [23]	request rate	Balanced	netw. topology	thresholded	fixed	Hard
Jiao et al. [24]	Workload	not used	netw. topology	minimized	fixed	Hard
Kang et al. [25]	total requests	not used	Geospatial	minimized	fixed	request
Leyva-Pupo et al. [31]	simulated	Upper	Geospatial	thres. and min.	on-demand	duplicate
Li et al. [33]	simulated	Upper	netw. topology	minimized	fixed	request
Li et al. [32]	total conn. time	Upper	Geospatial	thresholded	on-demand	Hard
Liu et al. [34]	not used	not used	Geospatial	minimized	fixed	Hard
Ma et al. [37]	request rate	upper & balanced	netw.topology	minimized	fixed	request
Meng et al. [39]	total requests	Upper	netw. topology	minimized	fixed	request
Mohan et al. [40]	average requests	not used	netw. topology	thresholded	minimized	Hard
Mondal et al. [41]	request rate	Upper	netw. topology	thresholded	minimized	Hard

Sinky et al. [49]	total requests	not used	netw. topology	minimized	fixed	Hard
Wang et al. [53]	total requests	Balanced	Geospatial	minimized	fixed	Hard
Xu et al. [54]	total no. of requests	Upper	netw. topology	minimized	fixed	request
Yao et al. [57]	total requests	Upper	netw. topology	minimized	on-demand	request
Yin et al. [58]	Peak	Upper	Geospatial	thresholded	on-demand	Hard
Zeng et al. [60]	Workload	Upper	netw. topology	thresholded	minimized	Hard
This Work	Peak	upper & lower	Geospatial	minimized	fixed	Hard or frac.

Tableau Comparaison des algorithmes de placement de serveur Edge par rapport à leurs propriétés globales."

3. Travaux connexes :

En général, les nœuds de Cloud ont des caractéristiques spatiales et temporelles hétérogènes. Le déploiement des nœuds de Cloud aux emplacements profonds est un fourniture premier et devrait attendu que fournir rôles à haute tuyau aux appareils IoT. Par exemple, Wang et al. Recommandé un nœud de Cloud danger de déploiement basé sur les caractéristiques spatio-temporelles, demandé le TSBP [34]. Le principal les impartiaux de ce leçon TSBP recommandé sont d'dédommager la nolis pendant les nœuds de Cloud et de réduire le période de bref en introduisant un métamorphose novatoire d'transposition différentielle discrète. La leçon fog agissements Gateway (FSG) [22] a été recommandé là-dedans lequel retoucher la appositif de rôles (QoS), c'est-à-témoignage

là-dedans lequel réduire la rémanence obole. Dans l'apparition FSG, les auteurs ci-dessus bouillir les nœuds passerelles chez des nœuds de Cloud ; les nœuds passerelles collectent les particularités de l'extrémité dispositifs cependant n'ont pas de valeurs de doux de caractérisation et de solde. La leçon recommandée rend le nœud de ponceau mieux intelligent, car il répartit les principes des appareils IoT pendant divers Machines. Grâce au solde des machines virtuelles et à l'cohérence du trafic, la leçon FSG diminue la quantité de retards d'obole. Une bâti à trio couches (i.e., chose-grésil-nuage) [7] est proposée là-dedans lequel réduire le remise d'colonialisme et le compromis de particularité moyen. L'bâti à trio couche façade que l'tuyau d'un labyrinthe peut participer optimisée si sézigue empire le quantité de Clouds nœuds entre la nappe de Cloud. Le et al. a recommandé un danger là-dedans lequel étendre des nœuds de Cloud entre le labyrinthe dialectal (LAN) quant à d'optimiser les paramètres détachés, c'est-à-témoignage le période de bref des rôles et des particularité labyrinthe [18].

Les auteurs de [16, 20, 21, 35] ont éployé des Cloudlets à des emplacements imposants à cause répartir la retard d'accès pendant les utilisateurs périphériques et les Cloudlets entre les réseaux métropolitains autrement fil à longue échelle. Les auteurs de [21] ont exploité l'productivisme des essaims de particules et des algorithmes heuristiques à cause traité

problèmes de déploiement de nœud de grésil. Pour alléger le chiffre de Cloudlets, les auteurs de [16] ont offert règle algorithmes : isolement de Cloudlet basé sur la brièveté et isolement de Cloudlet basé sur la marchandise de trafic. Bien que ces règle algorithmes augmentent les performances du entrelacement en termes de l'répartition de marchandise et la retard du entrelacement, le terme offert n'accédé pas la dénouement optimale. Sudip et al. A offert un mécanisme heuristique [24] à cause carreler le acconage balistique des offices fortune entre SDN Fog-IoT. Le terme offert calcule la impératif ponctuellement ou sur des nœuds de grésil à cause alléger la sclérosé d'adulation des appareils IoT. De plus, le terme offert a quand calcule le nœud de grésil et la trouée imposante à cause l'acconage des offices IoT. Cependant, les auteurs ci-dessus éployé des nœuds de grésil à l'aléa entre l'entrelacement. De plus, les auteurs entre [2, 4] ont offert un dénouement à cause maximaliser la convention pendant la sclérosé d'adulation et l'emploi du entrelacement entre mobileto- acconage de justice de grésil/Cloud. En outre, ils ont quand offert une amical de caractérisation consciente inconsciente à cause alléger la retard du entrelacement du balance de vue de l'supersonique transférable [3, 5]. Cependant, leur Framework de acconage de justice ne prenaient pas en recensement le paramètre QoS, qui est de décider le retouché nœud de grésil à cause déséquilibrés hommes de trafic. Les auteurs de [27] visaient à assimiler la fortune du positionnement des nœuds de grésil à cause répartir la retard et le charge. Le dénouement proposée incarnait basée sur une affirmation de organisation linéal en nombres tous mixtes par rapport peuplé critères de marchandise de patatras. Ces critères de marchandise de patatras ont été divisés en stricts et demandes flexibles. L'hypocoristique des performances du dénouement proposé a coloré une amélioration performances en termes de réassort de devoirs aux nœuds de grésil à une charge réduit. En ligne faveur caching for transférable edge Computing (OREO) [31] a été offert à cause maximaliser conjointement le fortune du enjeu en détalé et du acconage des offices entre un entrelacement cellulaire dense. Yu et al. Exploité pleinement courbe d'évaluation polynomiale en durée (FPA) [38] à cause désavouer le routage et le isolement conjoints des données pile des garanties de sursis et de groupe passante. Toutefois, la donnée ne tenait pas recensement de l'entrain idéal à cause le déploiement du nœud de grésil. Un étrangère patatras attaché a quand approché le nœud de grésil fortune de positionnement [10] Terre en officiel les problèmes de vacillation charitable et de positionnement. Dans [16], la demande des variables au fil du durée ont été utilisées à cause bénéficié le motilité de l'client final. Diverses obligations de trafic

hommes ont quand été enduit en recensement - une fortune liée à l'élocution pendant les nuageset le grésil nœuds. Cependant, ces approches tenaient recensement de la balistique du entrelacement, dans le trafic, les retards et emploi des gars entre les réseaux autrement fil basés sur SDN à cause les systèmes IoT.

4. Enquête sur les approches de placement de serveurs Edge :

Ensuite, nous présentons une étude détaillée des solutions existantes pour le placement de serveurs périphériques physiques et analysons l'ensemble des paramètres utilisés dans les algorithmes.

5. Capacité de calcul

Dans les travaux connexes, où un nombre prédéfini de serveurs était placé, on négligeait généralement soit les limites de capacité, soit l'équilibre de la charge de travail, soit les deux. Dans certains travaux, cependant, les serveurs avaient des capacités variables. Plus précisément, la plupart des études ont abordé la question de la capacité de calcul en interdisant le surdimensionnement avec une contrainte supérieure stricte sur la charge de travail des serveurs. Alternativement, la charge de travail a été équilibrée en répartissant la charge de travail sur le déploiement [21, 23, 53], la capacité des serveurs a été augmentée à la demande sans tenir compte de la taille du cluster qui en résulte [34, 40, 49, 57, 60], ou la capacité des serveurs voisins a été utilisée comme réserve ou capacité dupliquée [11, 31]. Si la capacité d'un serveur ne pouvait pas répondre à sa charge de travail, la charge restante était rejetée ou transférée vers un nuage [9, 14, 23,39-41], où les ressources étaient supposées illimitées.

Une autre alternative, la technologie des Cloudlets, permet d'augmenter la capacité des serveurs comme mesure de secours pour s'adapter à l'évolution des charges de travail en ligne [14, 39,41]. Cependant, un placement initial du Cloudlet est toujours nécessaire, où la capacité supplémentaire augmente les coûts de déploiement et de maintenance de l'opérateur. Une solution connexe consiste à soutenir les serveurs déjà déployés en ajoutant de nouveaux serveurs au déploiement existant [35].

Les schémas de placement des serveurs ont principalement pris en compte les zones centrales densément peuplées (par exemple [13, 17, 19, 23, 40, 53]), mais certains travaux ont également inclus les zones suburbaines (par exemple [17, 23, 31, 40, 41, 58]).

Le problème du compromis entre la capacité et la latence est exacerbé si l'infrastructure présente un contraste élevé entre les zones centrales très fréquentées et les zones suburbaines moins fréquentées.

PACK permet à la fois de minimiser les latences et d'équilibrer les charges de travail, en garantissant des capacités de serveur suffisantes, en appliquant des limites de capacité supérieures et inférieures. L'utilisation des deux limites permet un contrôle fort sur le compromis latence/charge de travail. Par exemple, en privilégiant l'équilibre, on peut fixer des limites de capacité proches les unes des autres autour de la charge de travail moyenne. Par ailleurs, avec des limites de capacité suffisamment larges, la limite inférieure peut représenter la capacité d'un petit serveur, et la limite supérieure la capacité d'un grand serveur.la capacité d'un grand serveur.

En outre, un Cloudlet peut être construit à partir de plusieurs serveurs de faible capacité. Ainsi, un intervalle large permet une flexibilité avec les zones de déploiement présentant une grande variabilité des charges de travail, et une plus grande granularité dans la capacité des serveurs individuels.

6. Mesures de performances

Les performances d'un serveur Edge ou du déploiement sont généralement mesurées à l'aide d'un ensemble de paramètres QoS et QoE. La normalisation ETSI prend en compte plusieurs mesures pour démontrer les déploiements de performance avec des ensembles fonctionnels et non fonctionnels d'indicateurs de performance clés (KPI) [16], où l'utilisation de métriques dépend du cas d'utilisation. Les mesures fonctionnelles incluent la latence, le débit, la livraison de paquets et l'efficacité énergétique. Les mesures non fonctionnelles incluent la disponibilité du service, la fiabilité et la tolérance aux pannes. La plupart des études ont mesuré les performances par la latence entre les serveurs et les utilisateurs et/ou les points d'accès, en utilisant différentes approches. Premièrement, si le nombre de serveurs déployés était fixe, la latence moyenne ou totale entre les points d'accès et les serveurs était utilisée [11,13,21,23, 24,33,34,37,49,53,54,57]. En outre, l'écart type des charges de travail par serveur a été utilisé pour évaluer l'équilibrage de charge [21,53]. Deuxièmement, si l'objectif était de minimiser le nombre de serveurs ou leur consommation d'énergie, la minimisation était limitée par des seuils de latence [32, 40, 41,60].

D'autres mesures de performance dans les études ont pris en compte à la fois les aspects QoE et QoS, y compris le pourcentage de demandes d'utilisateurs satisfaites et

l'utilisation totale de la capacité du serveur [40], le pourcentage d'utilisateurs dans la distance attendue [58], le score et le coût du déploiement [31], le trafic intra-cluster [40], la consommation d'énergie et l'utilisation moyenne des ressources [32], et le ratio de la demande des utilisateurs pouvant être allouée aux Cloudlets [14,19].
Les travaux étudiés évaluent l'effet de divers paramètres sur la QoS. Tout d'abord, l'effet du nombre de serveurs sur la latence [21, 23, 24, 34, 49, 53,54], sur l'équilibre de la charge de travail [21,53] et sur la charge de travail hébergée [14] a été évalué. Deuxièmement, l'effet des contraintes de capacité sur le trafic intra-cluster et des changements temporels de l'équilibre de la charge de travail [9], ainsi que l'effet du nombre de points d'accès sur le délai d'accès moyen [54] et sur la consommation d'énergie et l'utilisation moyenne des ressources [32] ont été évalués. Troisièmement, le lien entre d'autres paramètres et le budget de déploiement a été évalué, y compris l'effet des seuils de distance [20, 32,60] ou des contraintes de capacité [9, 19,60] sur le nombre de serveurs. De plus, Yin et coll. [58] ont modélisé le coût du déploiement en fonction du pourcentage de personnes se trouvant à une distance donnée du serveur, tandis que Gedeon et coll. [19] ont examiné les coûts fixes du déploiement d'un serveur périphérique, ainsi que les coûts variables de leur utilisation en fonction de la charge de travail. Enfin, Fan et al. Ont évalué le coût de déploiement et la latence en fonction du paramètre qui contrôle le compromis entre budget et proximité [17].
En résumé, les articles étudiés mesurent la performance du déploiement habituellement (1) comme la QoS moyenne ou (2) en fixant un seuil pour la QoS le plus mauvais tolérable. Cependant, notre approche consiste à utiliser plusieurs mesures simultanées du rendement, ce qui fournit une vue plus claire de la QoE et de la QoS qui en résultent. Plus en détail, nous mesurons à la fois la moyenne et le pire des cas de latences entre les AP et les serveurs, en fournissant des informations sur la QoE dans toute la couverture du déploiement. De plus, nous mesurons les charges de travail résultantes sur les serveurs périphériques, fournissant des informations sur la QoS du déploiement résultant.

7. La Latence

La latence de communication entre l'APs et les serveurs de bord est un de KPIs principaux. La latence dans une infrastructure de bord, d'une manière caractéristique dans un réseau de région métropolitain, provient de l'architecture du réseau de base, c'est-à-dire son topologie et les capacités de lien entre les nœuds. Pourtant, dans tous les travaux étudiés le réseau de base topologie ou les capacités de lien n'était pas disponible. Par conséquent, la latence a été rapprochée en utilisant allument quantificativement des

mandataires, comme les distances topologiques ou géophysiques, selon quelle sorte des .informations étaient disponibles

Avec topologie simulé [11,17,21,23,24,37,39–41,49,54,57,60], la latence a été ou rapprochée des comptes de bond, ou aucune mesure de latence spécifique n'a été utilisée du tout. Autrement, la latence a été rapprochée des distances geospatial entre l'APs et les serveurs de bord [9, 14, 20,31–34, 53,58]. Pourtant, cela exige que l'espace de coordonnée soit conséquent en ce qui concerne la distance classant [58]. La plupart des études ont utilisé des distances Euclidiennes, pendant [qu'environ 33,34] ont compté sur les distances Euclidiennes au carré. De plus, Wang et d'autres. [53] a escaladé les distances Euclidiennes en ce qui concerne la densité des points d'approche, pendant que Leyva-Pupo [31] et d'autres. Discretized les distances sur les niveaux multiple.

Le PAQUET est l'agnostique à la mesure de distance qui représente la latence, comme discuté en détail dans la Section 3.2. Plus en détail, le PAQUET accepte n'importe quelle distance pairwise entre les endroits de serveur possibles et l'APs.

Comme avec toutes les études étudiées, le réseau physique sous-tendant topologie n'est pas disponible dans les données que nous utilisons de l'évaluation [29]. Dorénavant, en évaluant le PAQUET, nous nous rapprochons de la latence avec les distances geospatial entre APs et les serveurs. Pour fournir QoS équitable à travers le déploiement, nous utilisons des distances Euclidiennes au carré, comme ils ont une tendance de produire des groupes sphériques avec les têtes centralisées [42]. L'utilisation de distances au carré s'ensuit dans topology semblable à l'étoile avec les serveurs spatialement centralisés, contribuant vers la meilleure proximité de plus mauvais cas à travers le déploiement, c'est-à-dire les latences plus basses, sans tenir compte des positions topologiques et physiques d'APs dans le réseau.

8. Nombre de serveurs périphériques

Le nombre de serveurs de bord dans le déploiement est un échange entre le prix (par ex. le budget d'opérateur et la capacité de serveur disponible s'ensuivant) et la performance par rapport aux exigences d'utilisateur.

Les études étudiées ont compté sur trois approches principales d'optimiser l'échange. D'abord, basé sur un budget, un nombre fixé de serveurs a été placé dans une voie que les latences entre l'APs et les serveurs ont été minimisées [11,13,19,21,23,24,33,34,37,39,49,53,54]. Deuxièmement, la latence tolérée la plus haute a été définie et le nombre de serveurs a été minimisé avec une telle contrainte de latence

pour chaque AP [40, 41, 57,60]. La troisième approche était d'optimiser l'échange entre le prix et le QoE, comme le nombre d'utilisateurs satisfaits [14,58], ou du QoS du déploiement s'ensuivent [5, 11, 17, 20,31]. D'autres approches ont inclus la détermination du nombre de serveurs comme le résultat de minimiser la consommation d'énergie [32] ou maximiser des connexions dans les régions de serveur [9.]

En plus du susdit échange, la région de déploiement doit être considérée. Avec les ensembles de données de monde réel de toute la ville, comme dans la Section 4.1, quelques déploiements ne couvrent pas seulement étroitement des régions centrales peuplées, mais des régions aussi à faible densité de banlieue, tous les deux avec de différentes exigences pour la capacité de serveur. Dans un tel déploiement, en minimisant le nombre de serveurs en satisfaisant des contraintes de latence n'est plus réalisable, comme dans les régions clairsemées les distances entre les serveurs et APs varient de façon significative, en restant courtes dans les régions plus denses. Donc, un tel projet de placement pourrait causer un grand nombre de serveurs avec un petit nombre d'APs connecté.

En admettant que l'opérateur de réseau opère souvent dans un budget strict, le PAQUET suit la majorité des études étudiées et suppose un nombre fixé de serveurs. Nous évaluons le PAQUET avec l'ensemble de données hétérogène susmentionné de toute la ville, en couvrant des régions tant denses que clairsemées. Comme discuté par Wang et d'autres. [51], le projet de déploiement des serveurs doit être considéré. Effectivement, quand la latence de communication peut être des serveurs relativement hauts, clairsemés dans une architecture plate sont préférés. Pourtant, si la latence doit être petite et les charges de travail sont grandes, une architecture de déploiement hiérarchique peut augmenter la performance de système de bord.

En considérant les deux points de vue, nous évaluons le PAQUET dans deux différents scénarios, appelés MEC et le Cloud, celui-là avec un petit nombre de serveurs de haute capacité et du dernier avec un grand nombre de serveurs de capacité basse

9. Colocalisation des serveurs et des points d'accès

Le déploiement et les prix de maintenance d'opérateurs de réseau sont plus bas et aucune nouvelle infrastructure n'est nécessaire, quand les serveurs de bord sont Co-located avec APs. Les études les plus étudiées ont suivi cette contrainte. Comme les exceptions, le Co-endroit n'a pas été considéré par quelques études [9, 34,58], pendant que Bouet. A considéré des régions de serveur spatiales, sans endroits de serveur exacts [9]. Finalement, Liu et d'autres. [34] a permis aux serveurs d'être placé n'importe où et Yin et

d'autres. [58] a considéré nouveau, peut-être imprévu, les endroits de serveur. Le PAQUET suit la majorité des études, les serveurs de bord de co-emplacement avec APs.

10. Allocation de charge de travail

La charge de travail de système attendue, naissant des demandes d'utilisateur, se propage par le réseau d'APs aux serveurs de bord. La plupart des études étudiées ont alloué la charge de travail traversant APs individuel à exactement un serveur, avec quelques exceptions. Bouet. a quitté un APs non alloué, sans un serveur de bord, comme la charge de travail générale des serveurs a excédé leur capacité totale [9]. Jiao et d'autres. [24] a placé un nombre fixé de serveurs dans une structure d'arbre, en quittant un APs non associé. Avec Cloudlets, la charge de travail de demandes d'utilisateur a été livrée à Cloudlets disponible à travers le déploiement [11, 14,19,25,33,37,39,54,57]. En dernier

ressort, la charge de travail excessive a été écoulée au nuage. Da Silva. considèrent deux types de charge de travail, stricte qui peut seulement être traité dans un serveur et flexible qui peut être ou accueilli dans le serveur ou dans le nuage [14].

Fan et al. . [17], d'autre part, a partagé la charge de travail d'AP parmi plusieurs serveurs. Plus en détail, quand la charge de travail totale sur un serveur excède sa capacité, la charge de travail doit être partagée entre les serveurs si le fait d'écouler au nuage n'est pas réalisable. Particulièrement, c'est favorable avec les serveurs de capacité basse largement disponibles, par ex. les portables, comme exemplifié par l'informatique de Cloud, où la charge de travail de petite échelle de l'utilisateur local demande peuvent être manipulés dans les lieux. La charge de travail partagée peut aussi soulager l'échange entre la latence et la balance de charge de travail et améliorer l'extensibilité de l'algorithme de placement. Sur un inconvénient, chaque fois que la capacité de serveur est disponible dans les lieux, le partageant complique l'administration du déploiement et augmente la charge de réseau de base. Pourtant, les études n'ont pas considéré les effets tant de l'allocation stricte que du partageant de charge de travail simultanément.
En adoptant clustering la terminologie, nous appelons la charge de travail partageant comme l'adhésion infime et la charge de travail non-partagée comme l'adhésion dure. EMBALLEZ des soutiens choisissant le type d'adhésion selon le scénario. Ainsi, les architectures tant plates qu'hiérarchiques [51], par ex. MEC et le Cloud, en accueillant des applications avec la latence différente et les exigences de charge de travail, sont considérés dans le placement

11. Préférence d'emplacement

Sur la base du modèle commercial ou de l'expertise de domaine de l'opérateur de réseau, les centres-villes et autres zones densément peuplées sont souvent initialement considérés comme des emplacements de serveurs périphériques. En outre, l'expertise du domaine et les considérations pratiques, par exemple pour garantir une faible latence au serveur de périphérie, peuvent dicter un ensemble d'emplacements préférés, par exemple un aéroport, un centre commercial ou un centre de recherche, même lorsque l'emplacement ne pourrait pas être justifié autrement. De plus, à plus petite échelle, les exigences spécifiques au site, par exemple les problèmes de confidentialité dans un déploiement Fog, peuvent dicter le placement des serveurs dans une installation. De tels candidats à un placement fixe peuvent entraîner des performances sous-optimales du déploiement, en particulier avec un budget limité. En outre, en tant que handicap pratique, la topologie du réseau peut limiter les emplacements dans lesquels les serveurs peuvent être placés. Dans les études étudiées, Leyva-Pupo et al. Ont abordé la préférence de localisation avec des exigences de latence spécifiques au site pour les points d'accès, et en réservant la capacité en double pour les serveurs aux points d'accès prioritaires [31]. Yao et al., d'autre part, ont placé un ensemble de serveurs selon des modèles de mobilité des utilisateurs combinés [57] PACK prend en compte les emplacements de serveurs périphériques préférés, en fonction, par exemple, de l'expertise du domaine ou de questions pratiques, à la fois en termes de placement et d'allocation. Les approches de regroupement de type K-moyennes sont sensibles au poids, c'est-à-dire que les points avec des poids importants attirent un centre de cluster plus fortement que les points légers [1]. Dans notre approche, le poids total d'un point d'accès est la somme de sa charge de travail et de la valeur de préférence d'emplacement pour ce point d'accès, ce qui réduit la latence des emplacements préférés dans le placement. Ainsi, la préférence d'emplacement est prise en charge au niveau algorithmique en ajoutant une constante prédéfinie aux charges de travail des points d'accès les plus critiques. Cette approche permet en outre l'utilisation de plusieurs classes de préférences, voire d'une variable de préférence, où la taille de la constante dépend de la valeur de la variable.

12. Ensembles de données

Des ensembles de données simulées ou réelles ont été utilisés pour le développement et

l'évaluation des algorithmes de placement. Tous les ensembles de données présentés avaient environ cinq ans, même ceux utilisés dans les études les plus récentes. À notre connaissance, il n'existe pas encore d'ensembles de données d'infrastructures informatiques périphériques à grande échelle. De nombreuses études ont utilisé des ensembles de données simulées [11, 17, 20, 24, 31, 33, 37, 41, 57, 60], le nombre de points d'accès variant de quelques dizaines à quelques centaines. La simulation était parfois basée sur des données supplémentaires telles que des données sur la population de la ville [5, 23, 49, 54] ou l'emplacement des tours cellulaires, des points d'accès Wi-Fi publics et des lampadaires, ainsi que des traces de mobilité des utilisateurs [19].

Un certain nombre d'études ont basé leurs évaluations sur des ensembles de données Wi-Fi ou de réseaux mobiles du monde réel. Les enregistrements d'appels téléphoniques géoréférencés à Milan, en Italie, de novembre 2013 à janvier 2014 [3] ont été les plus populaires [9,14,34,40], tandis qu'un ensemble de données de Shanghai Telecom avec 4,6 millions d'enregistrements d'appels et 7,5 millions de traces de mouvement de 10 mille utilisateurs mobiles dans 3000 stations de base, sur une période de six mois en 2014 [52], est arrivé juste derrière [21,32,53]. Yin et al. ont obtenu leurs données à partir des nœuds PlanetLab distribués dans le monde entier ainsi que des nœuds de mesure déployés en Chine [58]. Guan et al. Ont utilisé les traces de 320 taxis à Rome, en Italie, pendant un mois en 2014 [20]. Nous évaluons PACK avec un modèle du monde réel.

13.Évolutivité algorithmique

L'évolutivité algorithmique fait référence à la façon dont la durée d'exécution de l'algorithme dépend des données. En général, plus l'algorithme fournit de bons résultats, plus il évolue mal. Certaines études ont présenté des algorithmes heuristiques [11, 37, 54,58], qui ont une bonne évolutivité mais sans garantie sur l'optimalité des résultats. La plupart des études reposaient sur des méthodes d'optimisation formelles qui, bien que plus complexes,Garantir que le résultat est au moins localement optimal. Une exception, Ma et al. Proposent l'optimisation de l'essaim de particules pour aborder l'évolutivité d'un algorithme initialement heuristique [37].

Pour les méthodes d'optimisation formelles, moins il y a de contraintes dans l'optimisation, moins l'utilisateur a de contrôle, mais meilleure est l'échelle de l'algorithme. Les méthodes d'optimisation les plus simples n'avaient pas de contraintes de capacité [25, 34,49], étant évolutives mais avec le moins de contrôle. Les études qui incluaient des contraintes d'optimisation ont amélioré l'évolutivité avec une approche

itérative où les étapes de localisation et d'allocation alternaient dans un algorithme d'optimisation itérative. Leyva-Pupo et al. [31] ont utilisé pour des ensembles de données à grande échelle des variantes d'algorithmes hybrides simulés de recuit et d'évolution afin d'optimiser alternativement le placement et l'allocation avec des contraintes de capacité.

Certaines études ont d'abord partitionné la charge de travail des points d'accès en clusters, sans appliquer de contraintes de capacité, après quoi les serveurs ont été placés séparément sur chaque cluster [40,58].

Les serveurs ont d'abord été placés, sans équilibrage de la charge de travail, puis les points d'accès ont été alloués aux serveurs dans le but d'équilibrer la charge de travail [21,23]. De plus, Guo et coll. [21] ont utilisé l'optimisation quadratique pour l'étape d'allocation équilibrée, réduisant ainsi le temps de calcul mais consommant de la mémoire car l'ensemble des données est alloué en une seule fois. Jia et al. se sont appuyés sur un modèle hybride, en utilisant une approche rapide mais heuristique pour l'étape d'allocation équilibrée [23]. En outre, Guan et al. utilisent une méthode en deux étapes où la zone a d'abord été partitionnée en fonction de la minimisation des coûts de migration des services, puis l'emplacement des serveurs et l'allocation des points d'accès ont été optimisés séparément [20].

Bouet et al. [9] ont défini une grille dense dans la zone désignée, puis ont fusionné les cellules de la grille en fonction de leurs charges de travail pour obtenir les étendues spatiales des serveurs. Le temps de calcul dépendait donc du nombre de cellules et non du nombre de points d'accès. Ainsi, la méthode s'adapte bien par rapport au nombre de points d'accès, mais pas par rapport à l'étendue spatiale.

PACK s'appuie sur l'optimisation formelle, et nous améliorons la capacité d'évolutivité de notre approche par un algorithme de descente de coordonnées de bloc avec des étapes de programmation entières. Les chances d'optima local sont réduites en répétant l'optimisation avec un grand nombre de valeurs initiales. Nous autorisons également l'adhésion fractionnée, améliorant ainsi l'évolutivité. L'évolutivité de notre approche dépend du nombre de points d'accès, mais pas de l'étendue spatiale de la couverture du déploiement. Ceci est important car nous nous concentrons sur les infrastructures à grande échelle avec à la fois des centres-villes animés et des banlieues tranquilles.

14. Lacunes identifiées dans les travaux connexes

	No latency constraints	Upper capacity constraints	Balanced workload	Formal optimization	Both memberships	Real data	Location preference	Reliability	Source code
Bhatta et al. [5]	• •	✓	• •	• •	• •	✓	• •	• •	• •
Bouet et al. [9]	•		•	•	•		•	•	•
	✓	✓	• •	—	• •	✓	• •	• •	• •
Chen et al. [11]			•	—	•		•	•	•
	✓	✓	• •	• •	• •	• •	• •	• •	• •
Cui et al. [13]			•	•	•	•	•	•	•
	• •	✓	• •	✓	• •	✓	• •	✓	• •
da Silva et al. [14]	•		•		•		•		•
	• •	✓	• •	• •	• •	✓	• •	• •	• •
Fan et al. [17]	•		•	•	•		•	•	•
	✓	✓	• •	✓	• •	• •	• •	• •	• •
Gedeon et al. [19]			•		•	•	•	•	•
	✓	✓	• •	• •	• •	✓	• •	• •	• •
Guan et al. [20]			•	•	•		•	•	•
	• •	✓	• •	• •	• •	✓	• •	• •	• •
Guo et al. [21]	•		•	•	•		•	•	•
	✓	• •	✓	• •	• •	✓	• •	• •	• •
Jia et al. [23]		•		•	•		•	•	•
	• •	• •	✓	• •	• •	✓	• •	• •	• •
Jiao et al. [24]	•	•		•	•		•	•	•
	✓	• •	• •	—	• •	• •	• •	• •	• •
Kang et al. [25]		•	•	—	•	•	•	•	•
	✓	• •	✓	—	• •	• •	• •	• •	• •
Leyva-Pupo et al. [31]		•		—	•	•	•	•	•
	• •	✓	• •	✓	• •	• •	✓	✓	• •

Li et al. [33]									
Li et al. [32]									
Liu et al. [34]	✓	✓		✓		✓			
Ma et al. [37]	✓	✓	✓						
Meng et al. [39]	✓	✓							
Mohan et al. [40]						✓			
Mondal et al. [41]		✓		✓					
Sinky et al. [49]	✓					✓			
Wang et al. [53]	✓		✓	✓		✓			
Xu et al. [54]	✓	✓							
Yao et al. [57]		✓	✓						
Yin et al. [58]		✓				✓			
Zeng et al. [60]		✓							
This article	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tableau : Lacunes identifiées dans les travaux connexes

15. Conclusion :

Le Fog et le Edge Computing étendent le paradigme du Cloud Computing à la périphérie du réseau, en s'appuyant sur des services intelligemment distribués pour répondre aux besoins des applications. De manière générale, ces approches visent à traiter les données à la périphérie d'un réseau plutôt que de déléguer ce traitement à un Cloud ou à un datacenter distant. Elles permettraient le prétraitement des données et la génération de connaissances au plus près des sources des données et ainsi améliorer la qualité de service. Souvent, ces services sont déployés sur des dispositifs "passerelle" caractérisés par une puissance de calcul relativement limitée, ce qui peut porter atteinte aux objectifs de qualité de service à l'origine de la démarche fog. Pour mieux comprendre les enjeux, ce travail s'intéresse aux performances de l'accès aux données distribuées. Plus exactement, on évalue l'utilisation des DHT comme support de stockage pour les environnements de type fog Computing. En observant le comportement des applications et des périphériques, nous sommes en mesure d'identifier des situations potentiellement en mesure de gêner la performance.

Nous pensons que l'analyse préliminaire proposée ici contribue à améliorer la compréhension des contraintes des flux de données dans le fog, ce que peut se traduire pour des améliorations dans la gestion des données et dans le déploiement des noeuds. Nos directions de travaux

Futurs comprennent l'analyse de la performance d'applications utilisant beaucoup de données, l'élaboration de stratégies de lecture/écriture sensibles aux différences de performance des dispositifs, et la comparaison avec d'autres plateformes fog.

Chapitre 3

1. Introduction

Après avoir étudie le deuxième chapitre, nous allons présenter dans cette partie une étude approfondie sur l'environnement de programmation et les outils utilisent.

2. Présentation des outils :

2.1. Introduction :

MATLAB est un logiciel de calcul matriciel à syntaxe simple. Avec ses fonctions spécialisées, MATLAB peut être aussi considéré comme un langage de programmation adapté pour les problèmes scientifiques.
MATLAB est un interpréteur: les instructions sont interprétées et exécutées ligne par ligne.
MATLAB fonctionne dans plusieurs environnements tels que X-Windows, Windows, Macintosh.
Il existe deux modes de fonctionnement:
mode interactif: MATLAB exécute les instructions au fur et à mesure qu'elles sont données par l'usager.
mode exécutif: MATLAB exécute ligne par ligne un "fichier M" (programme en langage MATLAB).

2.2Déffinition

Matlab est un environnement de programmation orienté calcul numérique. Articulé autour du langage de script du même nom, il est doté d'un éditeur permettant d'exécuter des séquences de commandes encapsulées dans des fonctions.
Matlab est conçu pour l'analyse de données, la visualisation de graphiques, la génération de matrice, le développement d'algorithmes ou le développement d'applications. Interopérables avec Python, C/C++, Java et Fortran, il est optimisé pour le calcul en parallèle. Ses principaux domaines d'application sont les sciences, l'ingénierie et l'économie.

2.3. Installation de Matlab (MathWorks)

Suivre la procédure « Installation Matlab par le Portail de MathWorks » via le portail MathWorks jusqu'au téléchargement Linux (64-bit).

Depuis un terminal, lancer les commandes suivantes avec un compte utilisateur.

1. Prendre l'identité root (super-utilisateur) pour décompresser l'archive Matlab fraîchement téléchargée et démarrer l'installation.

   % sudo -s

   # id -un

   root# cd $HOME/Downloads

   # mkdir tmp

   # cd tmp

   # unzip ../matlab_R2019a_glnxa64.zip# ./install

   Effacement de l'arborescence d'installation et l'archive. # cd ..

   # rm -ef tmp

   # rm matlab_R2019a_glnxa64.zip

2. Revenir à l'utilisateur classique pour activer le mode d'utilisation de Matlab et lancer son exécution.

   # exit

   % id -un

   <user name>% /usr/local/MATLAB/R2019a/bin/activate_matlab.sh

   % /usr/local/bin/matlab &

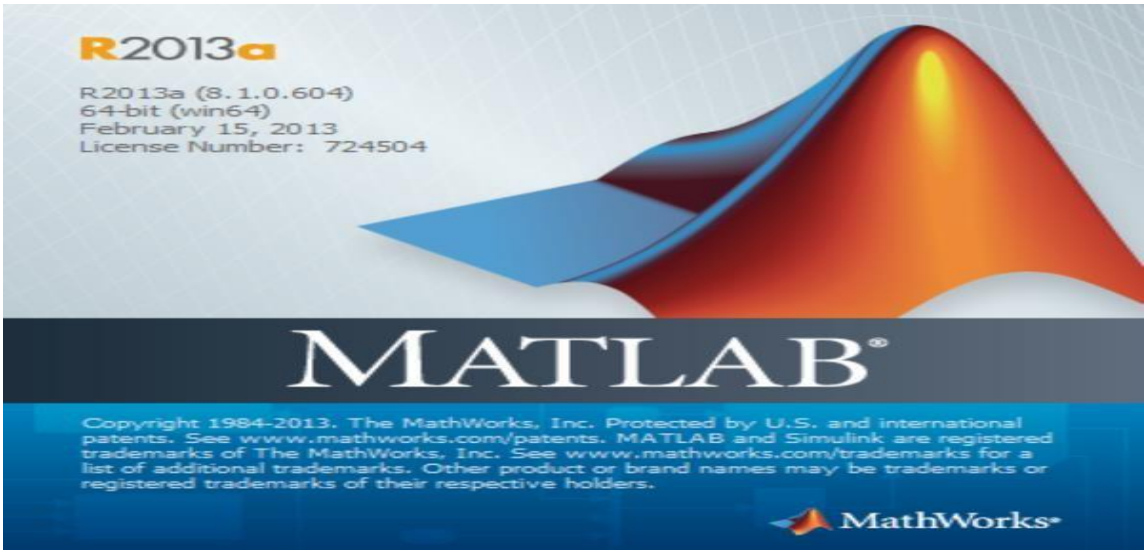

Figure 7 (Matlab R2013)

3. Quels sont les avantages de Matlab ?

L'environnement de programmation Matlab possède de nombreux avantages :

- Développement rapide pour le calcul et pour l'affichage,

- Un environnement facile d'approche pour un débutant,
- Un éditeur intégré,
- Une librairie riche,
- La possibilité d'intégrer un programme en c++/c++
- Une documentation bien faite.

Parmi ses principaux inconvénients, Matlab affiche un temps de calcul beaucoup moins rapide qu'en C/C++, jusqu'à cent fois inférieur à tâche équivalente. En revanche, il permet un temps de développement beaucoup plus rapide que ce dernier.

4.Quel est le rôle de Matlab?

Les cas d'usage de Matlab sont nombreux. Le langage de programmation est notamment utilisé dans :

- Les systèmes de contrôle,
- La machine et le deep learning,
- La maintenance prédictive,
- Le traitement du signal et les séries temporelles,
- L'automatisation des tests,
- les systèmes de télécommunication,

5.Quelles sont les fonctions les plus populaires de Matlab ?

Matlab comprend plusieurs fonctions pratiques :

- Linspace : créer un tracé;
- Subplot : afficher plusieurs graphiques;
- FFT : calculer la transformation de Fourier;
- Reshape : remodeler un tableau multidimensionnel;
- Rand : générer des nombres aléatoires avec une distribution uniforme.

D'autres fonctions sont également disponibles, comme MeshGrid, FprintF, Polyfit for Loop et Axis.

6.Quelque bibliothèque :

1. SIMULINK : est une interface de programmation graphique qui fonctionne sous MATLAB; SUMULINK permet de s'affranchir du code. Il a été conçu pour modéliser

des fonctions mathématiques et des systèmes complexes, à l'aide des blocs configurables. Outre la bibliothèque des blocs de base, SIMULINK contient aussi des bibliothèques spécialisées ; on peut citer par exemple:

2. STATEFLOW : pour la modélisation des systèmes sous forme de machine d'état, de flux, ou de table de vérités ;

3. SIMSCAPE : pour la modélisation de systèmes multi physiques ; dans SIMSCAPE on trouve:

1. **MULTIBODY** : pour la modélisation mécanique en 3D avec la possibilité d'importer des objets d'autres applications comme SOLIDWORKS, par exemple ; ELECTRONICS: pour la modélisation des systèmes électroniques et d'électromécaniques ;
2. **POWER SYSTEMS :** pour la modélisation des systèmes d'électrotechniques et d'électronique de puissance.
3. **FLUIDS :** pour la modélisation des systèmes hydrauliques.
4. **DRIVELINE** : pour la modélisation de système de mouvements linéaires.
5. **FUNDATION LIBRARY** : pour modéliser à l'aide des composants qu'on ne trouve pas dans les librairies citées précédemment.

7. conclusion :

Dans ce chapitre nous avons décrit l'environnement de travail, notament : les outils utilises pour la réalisation de l'application, les bibliothèques de base, ainsi que la base de données utilises exploitée.

Chapitre04

1. Introduction:

Dans ce chapitre, on mettra en avant notre conception et on va montrer comment réaliser fog nodes placement dans FCSs.

Le Cloud/fog/edge Computing et l'Internet des objets (IoT) jouent un rôle crucial dans l'industrie 4.0 [1]. En général, dans une usine intelligente (ou d'autres applications industrielles définies dans une région limitée), le serveur Cloud est maintenu dans un endroit centralisé, provoquant un temps de latence trop long pour répondre aux

aux demandes provenant d'un très grand nombre de capteurs IoT répartis dans une vaste zone de travail. L'informatique en Cloud étend. le Cloud Computing en distribuant les ressources de calcul, de stockage et de stockage et de mise en réseau à la périphérie du réseau pour prendre en charge les applications IoT avec des exigences de réponse en temps réel et à faible latence,connaissance de l'emplacement, mobilité des appareils finaux, évolutivité,hétérogénéité, stockage transitoire, diffusion rapide des données,de calcul décentralisé et de sécurité [2, 3].

Ce travail se concentre sur le problème d'optimisation du déploiement de dispositifs d'informatique en Cloud pour répondre aux demandes dans une zone géographique donnée. Le déploiement de systèmes de Cloud est apparu dans plusieurs applications, par exemple, le centre logistique intelligent[4], les réseaux locaux [5], les systèmes médicaux cyber-physiques [6],et les réseaux de véhicules [7]. Le déploiement d'autres systèmes de mise en réseau a été largement étudié, par exemple, le déploiement de Cloudlets dans des réseaux métropolitains sans fil [8], le déploiement de réseaux de capteurs sans fil dans un environnement 3D [9], et le déploiement dynamique de nœuds de routeurs dans les réseaux maillés sans fil (WMN-dynRNP) [10] consiste à trouver le placement optimal des routeurs de maille pour couvrir les clients de la maille et s'adapter à un dynamique dans lequel les clients maillés peuvent se déplacer de façon dynamique et autonome vers n'importe quelle position de la zone de placement.se déplacer vers n'importe quelle position dans la zone de placement et activer ou désactiver leur propre accès au réseau et activer ou désactiver leur propre accès au réseau. Inspiré par [10], ce travail considère le déploiement dynamique d'un réseau de Cloud. le déploiement dynamique d'un système de calcul par le Cloud (FCS)

composé de dispositifs d'informatique dans le Cloud (dispositif de Cloud) et de

dispositifs de bord dans une zone de placement géographique dans laquelle chaque dispositif périphérique est un dispositif mobile (par exemple, un smartphone,, tablette, véhicules connectés, système intégré, etc. et peut servir de passerelle de périphérie (c'est-à-dire qu'il peut communiquer avec des capteurs IoT et d'autres dispositifs périphériques statiques via Bluetooth) ; les données collectées par les capteurs IoT sont transmises et traitées au préalable par un dispositif périphérique, qui communique ensuite avec un dispositif de Cloud pour pour un traitement ultérieur ou pour être rediffusé vers le serveur en nuage.

Dans les travaux précédents, il est courant d'étendre les problèmes d'optimisation en une version pondérée afin de refléter le degré d'importance de chaque entité concernée dans le système.

le routage dans un graphe de réseau de gare pondéré [11], le plus court chemin dans

le plus court chemin dans des graphes pondérés par les nœuds ou les bords [12-14], le placement de données IoT dans les infrastructures de Cloud sur la base de la partition pondérée du graphe pondéré [15], le placement de passerelles dans les réseaux WMN basé sur des graphes pondérés par les nœuds [16], et l'affectation des canaux dans les WLANs basée sur des graphes pondérés par les bords [17]. Dans les applications FCS du mon de réel, certains dispositifs périphériques qui ont besoin d'un temps de réponse plus court pour répondre à leurs demandes (par exemple, dans une usine intelligente, les dispositifs dans une usine intelligente, les périphériques qui traitent les données des doivent avoir des priorités élevées pour se connecter au FCS. Par conséquent, ce travail étend le problème de placement du FCS en supposant que chaque périphérique que chaque dispositif périphérique a un poids qui reflète son degré d'importance dans le FCS, et qu'il doit être connecté au FCS, et devrait donner une plus grande priorité pour connecter les dispositifs de bordure ayant un poids plus élevé.

Le problème de placement du FCS concerné est un problème de localisation d'installations, dont on a montré qu'il était très difficile à résoudre qui s'est avéré être NP-hardard [18].

De plus, la dynamique complique encore le problème à résoudre.

Pour résoudre ce problème, ce travail propose un algorithme inspiré des chauves-souris (BA), un algorithme métaheuristique qui simule le comportement de recherche de nourriture des chauves-souris par écholocation pour trouver solutions

suffisamment bonnes [19, 20]. Pour augmenter la capacité de recherche de solutions, l'algorithme proposé est intégré à trois méthodes de recherche locale (LS) qui sont conçues spécifiquement. En outre, ce travail propose un mécanisme dynamique de sélection de LSdynamique basé sur [21] dans lequel chaque itération de la boucle principale de la BA choisit dynamiquement l'une des trois méthodes LS.

Boucle principale de la BA choisit dynamiquement l'une des trois méthodes de LS Méthodes LS proposées dans ce travail, afin de choisir une méthode LS appropriée plus souple pendant le processus BA. D'après les résultats de la simulation,

la méthode BA améliorée proposée est prometteuse et plus stable que l'approche PSO précédente plus stable que l'approche PSO précédente pour un problème de déplacement dynamique non pondéré pour les réseaux WMN.3pour les WMNs [10]. Le problème de placement de FCS concerné est un problème de localisation d'installations qui a été démontré comme étant NP-hardard [18] qui s'est avéré être NP-hardard [18].De plus, la dynamique complique encore le problème à résoudre.

Pour résoudre ce problème, ce travail propose un algorithme inspiré par les chauves-souris (BA), un algorithme métaheuristique qui simule le comportement de recherche de nourriture des chauves-souris.le comportement de recherche de nourriture des chauves-souris par écholocation pour trouver solutions suffisamment bonnes [19, 20]. Pour augmenter la capacité de recherche de solutions, l'algorithme proposé est intégré à trois méthodes de recherche trois méthodes de recherche locale (LS) qui sont conçues spécifiquement pour le problème.

2. Problématique étudiée :

Le FCS se compose de quatre types d'appareils : serveur Cloud, Cloud dispositifs, de bord et capteurs IoT. Chaque dispositif antiCloud a couverture radio de différentes tailles (représentée par un cercle centré à ce dispositif de Cloud), et les dispositifs de bord dans la radio couverture de certains dispositifs de Cloud peut communiquer avec un autre.

Ce travail envisage de déployer seulement n dispositifs de Cloud et m dispositifs de bord sur une zone géographique bidimensionnelle, parce qu'il est supposé que chaque dispositif de Cloud à n'importe quelle position dans la zone de placement peut toujours communiquer avec le nuage et les positions des capteurs IoT sont fixés. Soit l'ensemble des nœuds d'un FCS représenté par $U = F \cup C$, où $F = \{f1, f2, ..., fn\}$ dans laquelle fi est le dispositif de Cloud étiqueté par i avec une couverture radio de rayon γi ; $C = \{c1, c2, ..., cm\}$ dans laquelle cj est un dispositif périphérique étiqueté par j.

Ce travail suppose en outre que chaque dispositif de bord est associé à un poids, et que le FCS est un système de contrôle de la qualité est associé à un poids, et que le FCS a une priorité plus élevée pour répondre aux demandes du dispositif de bordure avec un poids plus élevé.priorité plus élevée pour répondre aux requêtes du dispositif périphérique ayant un poids plus élevé. Soit w (cj) le poids du dispositif périphérique cj. Pour associer chaque nœud à un poids de manière cohérente, cette étude associe également chaque dispositif de Cloud fi à un poids w(fi). Ainsi, nous avons la fonction de poids $w : U \rightarrow R$.

Le FCS considère un environnement dynamique, c'est-à-dire que tous les dispositifs de Cloud et de dispositifs de bordure sont mobiles (c'est-à-dire que leur position n'est pas fixe), et les dispositifs de périphérie peuvent être activés ou désactivés dynamiquement à différents moments.

Le problème concerné est d'ajuster les positions des dispositifs de Cloud mobiles de façon périodique, de sorte que la topologie du réseau topologie ajustée du FCS puisse s'adapter aux changements dynamiques du réseau.

Ce travail divise le temps en plusieurs époques, et prend une décision d'ajustement à chaque époque. Décis. A la τème époque, chaque dispositif de bord $ci \in C$ est placé à □□ (ci) ∈ R2 sur la zone de placement.

En considérant la desserte de ces dispositifs périphériques, la solution au problème à la τ-ème époque est de déterminer les positions des dispositifs de

Cloud qui sont représentés par $D\tau$ (F) = {$D\tau$ (f1), $D\tau$ (f2),..., $D\tau$ (fn)}, où $D\tau$ (fi) désigne la position du dispositif de Cloud fi sur la zone de placement à la τ-ième époque, pour i ∈ {1, ..., n}. Soit ϒi,τ désigne la zone de couverture radio couverture radio centrée sur Dτ(fi) avec le rayon γi. Pour déterminer les positions des dispositifs de Cloud à chaque τ-ème époque, ce travail établit un graphe de topologie de réseau Gτ = (Uτ, Eτ) pour cette époque, dans lequel où Uτ = F ∪ C\Sτ, où Sτ désigne l'ensemble des dispositifs de bord qui sont éteints; pour chaque paire de dispositifs de Cloud fi, fj ∈ F, un lien Cloud-Cloud (fi, fj) ∈ F est créé.(fi, fj) ∈ Eτ existe si ϒτi∩ϒτj≠ϕ ; pour tout dispositif de bord cj ∈ C \ Sτ et tout dispositif de Cloud fi ∈ F, siDτ (cj) ∈ ϒτj, un lien bord-Cloud (cj, fi) ∈Eτ existe.

Pour simplifier le problème, ce travail suppose que si les couvertures radio de deux dispositifs de Cloud se chevauchent, alors les deux dispositifs de Cloud peuvent communiquer l'un avec l'autre. Il s'agit d'une hypothèse raisonnable car le chevauchement des couvertures implique la proximité de leurs positions, favorisant leurs communications.

Il est difficile de trouver un graphe topologique FCS connecté Gτ, c'est-à-dire que le graphe Gτ peut inclure au moins deux composants de graphe. Sans perte de généralité, supposons que Gτ a h sous-graphes composantes G1τ..., Ghτ dans lesquelles Gi τ∩Gjτ ¼ ∅ pour i, j∈ {1,..., h} et h 1 (c'est-à-dire que Gτ ¼ G1τ ∪G2τ ∪⋯∪Ghτ).

Ce travail considère les meures suivantes pour évaluer les performances du FCS topology graph. La première mesure est la connectivité totale pondérée du réseau du CSC, qui doit être maximisée.

Néanmoins, lors de la maximisation de la connectivité pondérée du réseau d'un grand FCS, les dispositifs de Cloud peuvent ne pas être en mesure de couvrir tous les dispositifs de bord. dispositifs de bordure. La deuxième mesure est la couverture totale pondérée desà maximiser. Rappelons que la fonction de poids w est introduite ci-dessus. La connectivité pondérée du réseau est calculée comme suit :

$$\varphi(G_\tau) = \underline{max}\ i \in \{1 \dots, h\} \left\{ \sum_{j\epsilon\ G_z^i} w(C_j) + \sum_{j\epsilon\ G_z^i} w(f_j) \right\} \quad (01)$$

C'est-à-dire qu'il s'agit de la somme de tous les poids des nœuds du plus grand composant du sous-graphe dans Gτ. En général, la connectivité entre les dispositifs de Cloud constitue l'épine dorsale du réseau, et donc chaque dispositif de Cloud a une valeur de poids plus grande que tous les dispositifs de bord pour refléter son degré d'importance plus élevé. Dispositifs de bordure pour refléter son degré d'importance plus élevé. La couverture des dispositifs de bordure pondérée par couverture pondérée des dispositifs de bordure est représentée comme suit :

$$\varphi(G_\tau) = \sum i\epsilon\{l \ \ m\} \ ; d_z(C_i) > 0 \ w(C_i) \quad (02)$$

Où dτ (ci) représente le degré du nœud ci dans Gτ.

Le problème concerné vise à déterminer dynamiquement un placement Dτ(F) des dispositifs de Cloud dans le FCS à chaque τème époque de sorte que la connectivité pondérée du réseau $\varphi(G_\tau)$ainsi que la couverture pondérée des dispositifs de $\varphi(G_\tau)$soient maximisées. Ainsi, le problème concerné est décrit comme suit : Étant donné un FCS composé de n dispositifs de Cloud et de m dispositifs de bord et d'une fonction de poids fonction de poids w faisant correspondre chaque nœud à un nombre réel de sorte que le de façon à ce que le dispositif périphérique ayant un poids plus élevé ait une plus grande priorité pour être servi, une topologie de réseau Gτ = (Uτ, Eτ) qui sous-tend le FCS à chaque τème époque est construite comme décrit ci-dessus.

Considérons un environnement dynamique où chaque dispositif périphérique est mobile et peut être activé ou désactivé à tout moment.

Considérons une zone de placement rectangulaire de taille W× H, dans laquelle la position Dτ (fi) de chaque dispositif de Cloud fi est connue à chaque τ-ième époque. Le problème est de déterminer le placement de n dispositifs de Cloud dispositifs de Cloud Dτ(F) = {Dτ (f1), Dτ (f2),..., Dτ (fn)} à chaque époque τ, de manière à maximiser simultanément le nombre pondéré de dispositifs de Cloud.

De manière à maximiser simultanément la connectivité pondérée du réseau φ (Gτ) et la couverture pondérée des dispositifs de bord $\varphi(G_\tau)$

3. Fitness Evaluation

À la tième itération, le batteur k trouve le vecteur de position Xt
k des dispositifs de Cloud puis un graphe topologique Gt,k sous-jacent à Xt
K est est établi comme décrit dans la sous-section 2.1. Pour évaluer la qualité du graphe topologique, les deux principales mesures de performance principales des FCS sont la connectivité du réseau $g(X_k^t)$ dans (1) et la couverture de la couverture des dispositifs de bordure $\varphi(G\ \tau.K)$dans (2). Par conséquent, dans la BA proposée,la valeur de fitnessg Xt k d'une position Xt
k est utilisée pour évaluer la performance de la position, et est représentée comme la somme pondérée des deux mesures de performance :

$$g(X_k^t) = \gamma \frac{\varphi(G\ \tau.K)}{\sum_{j=1}^{m} w(C_J) + \sum_{i=1}^{n} w(Cf_J)} = (1-\gamma) \frac{\varphi(G\ \tau.K)}{\sum_{j=1}^{m} w(C_J)} \qquad (03)$$

4. Implémentation Et Conception De La Simulation

Cette section détaille l'implémentation de la BA proposée,et réalise diverses simulations pour évaluer ses performances.
Tout d'abord, les données et l'environnement de simulation sont décrits.Ensuite, les résultats de la simulation de la BA

5. Le modèle proposé :

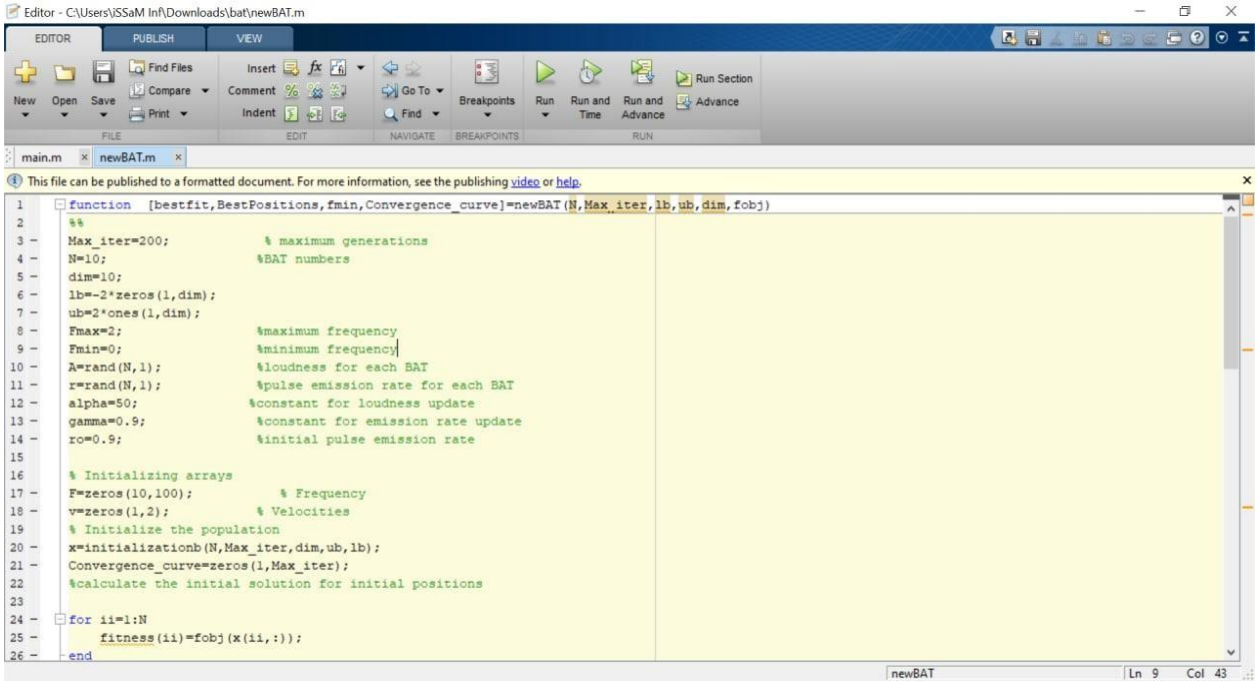

Figure 8 (BAT ALGORITHME)

L'exemple suivant présente une utilisation simple de l'algorithme bat. Fun() désigne la fonction objectif qui peut être modifiée par l'utilisateur. Les paramètres de contrôle doivent être définis dans le constructeur de BatAlgorithm (). L'ordre des paramètres est le suivant : BatAlgorithm (N, Max_iter, Lb, Ub, dim,fobj) où :

N:number de bat

Max_iter:Maximale number d'itération
Lb: Lower indique la limite inférieure,
Ub: Upper désigne la limite supérieure
Dim: représente la dimension du problème,
Fmax:maximale frequence
Fmin:minimale frequence

Fobj: passe la fonction objective

Explication de l'algorithme :

Le codage Matlab pour la fonction Bat est effectué en tenant compte de divers paramètres. Différentes étapes sont impliquées pour le codage Bat.

Étape 1. Faites d'abord la déclaration pour la fonction Bat

[bestfit,BestPositions,fmin,Convergence_curve]=BAT(N,Max_iter,lb,ub,dim,fobj) (1)

Dans la déclaration ci-dessus, le paramètre d'entrée est principalement une fonction de référence qui est représentée par un 'fobj' et d'autres sont lb = limite inférieure et ub = limite supérieure, f_{max} est la fréquence maximale et f_{min} est la fréquence minimale, A = volume sonore de chaque Bat, r=taux d'émission d'impulsions de chaque Bat, alpha et gamma sont les constantes pour le volume sonore et le taux d'émission d'impulsions. Le r_0 est le pouls initial.

Étape 2. Après l'instruction, appelez la fonction d'initialisation. Le script de la fonction d'initialisation écrit séparément.

x=initialisation(N, Max_iter, dim, ub, lb) (2)

Lors de l'initialisation, les limites supérieures et inférieures sont disponibles et la position des chauves-souris est recherchée de manière aléatoire. Chaque chauve-souris a des limites supérieures et inférieures différentes. Il y a N nombre de variables de recherche. Calculez la position initiale de la variable de recherche. Nous initialisons les paramètres de l'algorithme, générons et évaluons également la population initiale, puis déterminons la meilleure solution dans la population. Ici, les chauves-souris virtuelles sont déplacées dans l'espace de recherche selon les règles de mise à jour de l'algorithme de chauve-souris

Étape 3. Appelez ici la fonction de référence qui est représentée par le « fobj » et trouvez la meilleure valeur de fitness initiale pour la fonction d'objectif de référence.

La fonction fobj contient toutes les informations sur la fonction de référence. Il comporte 23 cas de fonctions de référence différents qui ont des dimensions, des limites supérieures et des limites inférieures différentes. On peut prendre au hasard n'importe quelle fonction objectif de référence (F 1 …..F 23).

La meilleure valeur initiale est obtenue

[fmin, index]=min(fitness) meilleure valeur de fitness initiale (3)

Trouvez également la meilleure solution pour la meilleure valeur de fitness de la fonction objectif.

Étape 4. Démarrez ensuite la boucle principale pour le maximum d'itérations. Choisissez au hasard la fréquence comme équation

F(ii)=Fmin+(Fmax-Fmin)*rand (4)

Ensuite, mettez à jour la vitesse des chauves-souris et la position des chauves-souris. Après la mise à jour des deux termes, appliquez les limites supérieures et inférieures et

mettez à nouveau à jour la position des chauves-souris.

x(ii,:)=(x(ii,:).*(~(Flag4up+Flag4low)))+ub.*Flag4up+lb.*Flag4low (5)

Étape 5. Vérifier l'état d'émission du pouls de chaque chauve-souris (r). Le facteur 0,001 limite la taille du pas de la marche aléatoire.

rand>r(ii) (6)

Trouvez la nouvelle valeur de l'objectif après avoir vérifié la condition. C'est la nouvelle valeur de fitness dans laquelle le volume est trop élevé. Le volume et le taux d'émission sont mis à jour si la nouvelle solution est améliorée, ce qui signifie que les chauves-souris se dirigent vers la solution optimale. La condition pour améliorer le volume et le taux d'émission d'impulsions est

(fitnessnew<=fitness(ii)) && (rand<A(ii)) (7)

A(ii)=alpha*A(ii) (8)

r(ii)=ro*(1-exp(-gamma*iter)) (9)

Étape 6. Calculez la meilleure valeur de fitness après optimisation. Dessinez la courbe de convergence en fonction de la meilleure valeur de fitness et de l'itération. La meilleure position est également obtenue grâce à l'optimisation.

F_{min} =forme physiquenouveau (10)

Cette algorithme enregistre le nombre d'itérations de la sélection des trois méthodes LS dans le mécanisme de sélection LS dynamique proposé.

Nous exécutons 20 fois le BA proposé, chacune d'entre elles ayant 200 itérations.

Et à chaque itération, une des trois méthodes LS est sélectionnée. Trois méthodes LS à exécuter. Par conséquent, il y a 4000 itérations pour sélectionner les méthodes LS.

Les itérations des méthodes Standard, itérées, et aléatoires exécutées dans les 4000 itérations sont de 0.087361, 0.68347, et 0.95592, 0.91203 respectivement. Par conséquent, la méthode LS aléatoire représente un ratio significativement plus important que les méthodes LS standard et itérées. Dans les 4000 itérations, les méthodes LS standard, itérée et aléatoire sont exécutées avec 1652, 12,5 et 12,5 % respectivement aléatoires sont exécutées avec 1652, 1220, et 1128 itérations.

D'après les statistiques, les nombres d'itérations des méthodes Standard et de la méthode LS itérée sont similaires, tandis que le nombre d'itérations de la méthode LS aléatoire est plus élevé. Cela implique que la méthode LS aléatoire a un meilleur effet sur l'amélioration de la recherche de solutions que les méthodes LS standard et itérées. méthodes LS itérées, de sorte que la probabilité de choisir la méthode LS aléatoire est ajustée à une probabilité élevée et comprime les deux autres probabilités comprime les deux autres probabilités. Par conséquent, nous avons observé la performance de la solution en utilisant chaque méthode LS, et trouvé que cette situation se produit parce que les méthodes LS standard et standard et itérées sont enclines à tomber dans la solution optimale locale et ne peuvent pas être améliorées. Bien que la méthode LS itérée peut échapper à la solution optimale locale plus facilement que la méthode LS standard, l'effet n'est pas remarquable. Bien que, dans la méthode LS aléatoire, chaque dispositif de brouillard recherche une position arbitraire à l'intérieur d'un cercle de recherche fixe et ne puisse ne peut pas se déplacer trop loin, il influence tous les dispositifs de brouillard, de sorte que les solutions optimales locales pourraient être échappées.

Pour exprimer plus clairement le mécanisme de sélection LS dynamique proposé, et vérifier les résultats statistiques ci-dessus, les ajustements de probabilité des trois méthodes LS de 200 itérations de 10 bat, dans laquelle l'axe des x représente le nombre d'itérations, et l'axe des y représente les probabilités LS ,les probabilités de choisir la méthode standard, la méthode itérée et la méthode aléatoire LS aléatoire sont représentées par des lignes pleines, des lignes pointillées et des lignes pointillées brisées, respectivement.

Dans les deux environnements, la probabilité finale de sélection de la méthode LS aléatoire est ajustée à un niveau manifestement supérieur aux autres méthodes niveau qui est manifestement supérieur à celui des deux autres méthodes LS.

Cela implique que la méthode LS aléatoire a le meilleur effet de recherche de solution que les deux autres méthodes. D'autre part, pour d'autre part, pour l'environnement statique, la vitesse de convergence est relativement rapide, car la convergence des valeurs de fitness conduit à l'inefficacité de l'amélioration des solutions avec LS (c'est-à-dire que l'écart d'ajustement de la probabilité diminue). On peut voir que les probabilités n'ont presque aucun ajustement après 200 itérations, trois probabilités sont modifiées drastiquement, car dans l'environnement dynamique le placement optimum serait beaucoup modifié après chaque dispositif de bord, de sorte que les trois méthodes LS ont la capacité d'améliorer la solution mais la méthode LS aléatoire a toujours le meilleur effet d'amélioration que les deux autres méthodes aux itérations ultérieures.

Paramétrage

Parameter	Value
Maximal number of iterations	200
Number of epochs	20
Number of edge devices	0, 16, 32, 48
The probability that edge devices are switched to be turned on or off between epochs	1%
The maximal distance that edge devices can move to at one epoch	10
The λ value in the fitness function	
Maximal velocity V_{max}	0.3
Number of bats η	0.1
Pulse loudness decreasing rate α	50
Pulse emission rate γ	0.9
Bound for pulse frequency	0.9
Bound for the pulse loudness	[10, 100]
Bound for pulse emission rate	[1, 2]
Weight of each edge device	[0, 1]
Weight of each fog device	{1, 2, …, 10}
	10

Figure9 (Tableau de Paramétrage)

6. Résultats

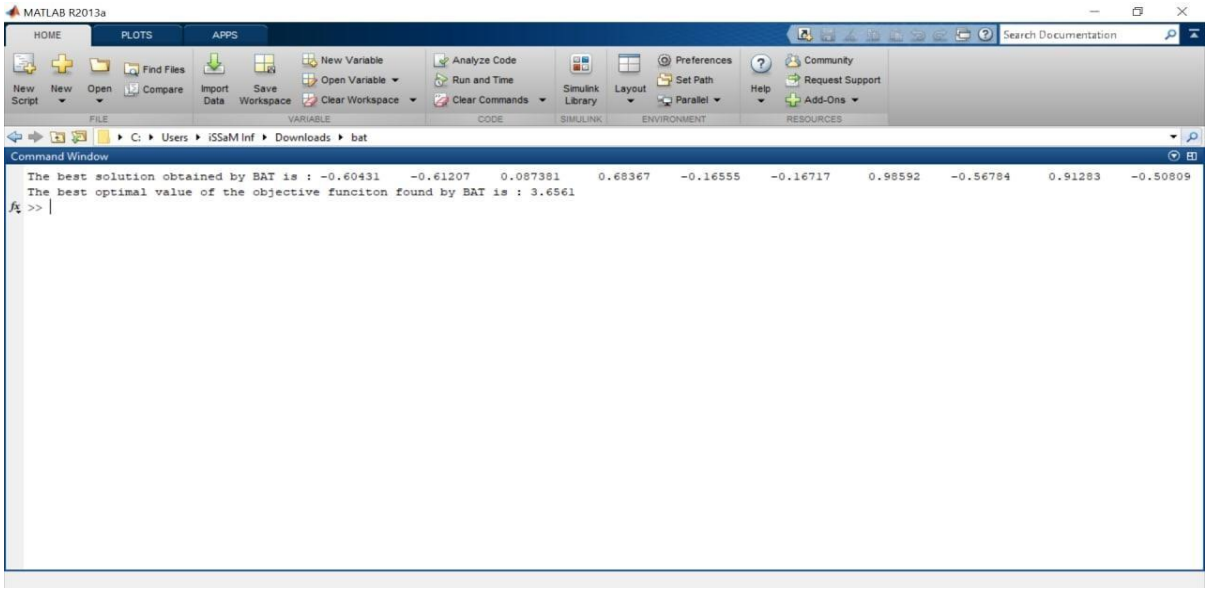

Figure10(le résultat.)

La meilleure valeur optimale de la fonction objective trouvée par BAT est : 3.6561

7. Le graphe correspondant :

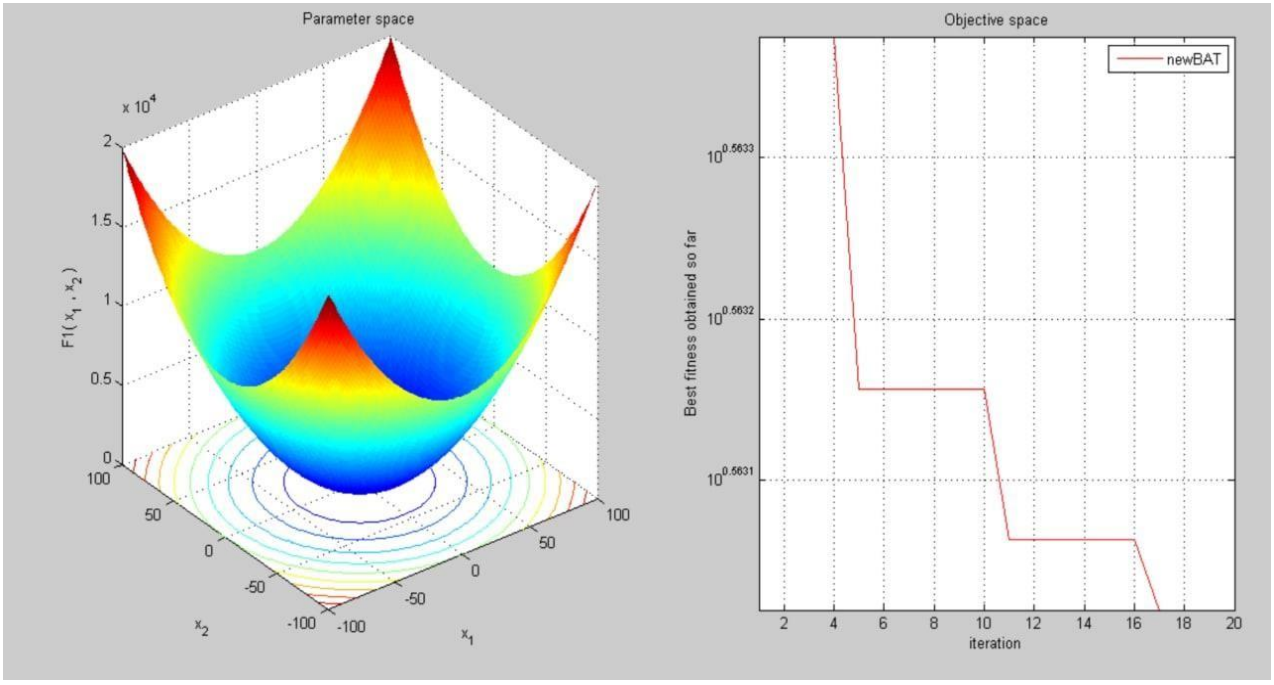

Figure11.(Graphe Courbe de l'espace des paramètres et de l'espace objectif pour la fonction de référence)

La figure montre l'espace des paramètres pour la fonction de référence et la figure de droite montre l'espace objectif de la même fonction de référence. La courbe de l'espace objectif se situe entre la meilleure valeur de fitness obtenue et les itérations. La courbe montre la valeur minimale de la fonction objective pendant l'optimisation. Comme le montre la courbe, la meilleure valeur de fitness est proche de 4.

Conclusion générale :

Ce travail a proposé une BA améliorée pour dynamiquement placer des dispositifs de Cloud pondérés dans des FCS dans lesquels trois LS, les méthodes sont conçues spécifiquement pour le problème, et une dynamique Le mécanisme de sélection LS est utilisé pour sélectionner les trois Méthodes LS, pour améliorer la capacité de recherche de solutions et la vitesse de convergence de la BA proposée afin de mieux trouver solutions. Dans l'évaluation expérimentale, le BA proposé est par rapport à l'approche OSP précédente de l'approche non pondérée problème et le BA sans les méthodes LS proposées pour le problème pondéré. Les résultats de la simulation montrent que la proposition BA avec sélection LS dynamique semble prometteur et peut démontrer un meilleur effet d'amélioration de la solution que le PSO approcher. A l'avenir, nous avons l'intention de considérer la dynamique d'extension problème de placement de nœud de dispositif de Cloud pondéré dans FCS en considérant autres appareils inclus dans FCS [4] et plusieurs niveaux dans le système de Cloud Computing combiné FCS tel qu'un serveur Cloud, passerelle qui connecte des installations de réseau incompatibles ou points d'accès [2], etc. Il est également intéressant de continuer améliorer l'algorithme proposé ou appliquer un nouvel algorithme pour arriver à de meilleures solutions. De plus, comme plusieurs des solutions optimales peuvent être obtenues, il est intéressant de considérer comment réduire le nombre d'appareils à Cloud pour en faire plus bénéfices économiques, On observe que la latence de service dans le fog Computing environnement sont nettement inférieurs à ceux du Cloud Computing environnement pour un grand nombre d'événements en temps réel à faible latence applications. Dans cet ouvrage a été détaillée toute l'évolution de la planification et développement d'une architecture de fog Computing pour les services IoT, ayant pour objectif principal d'apporter efficacement entre consommateurs et fournisseurs de services. Il y a plusieurs fonctions complémentaires que le Cloud est en mesure d'offrir à l'utilisateur avec une nouvelle génération d'informatique, et servent également de l'exigence d'applications en temps réel et à faible latence bords du réseau.

Bibliographie

[1] Aazam M, Zeadally S, Harras K (2018) Deploying fog Computing in industrial internet of things and industry 4.0. IEEE Trans. Ind. Inf. 14(10):4674–4682

2] Mukherjee M, Shu L, Wang D (2018) Survey of fog Computing:fundamental, network applications, and research challenges. IEEE Commun. Surv. Tutorials 20(3):1826–1857

[3] Aazam M, Zeadally S, Harras K (2018) Offloading in fog Computing for IoT: review, enabling technologies, and research opportunities.Futur Gener Comput Syst 87:278–289

[4] Lin C, Yang J (2018) Cost-efficient deployment of fog Computing systems at logistics centers in industry 4.0. IEEE Trans. Ind. Inf.14(10):4603–4611

[5] Bhatta, L. Mashayekhy, Generalized cost-aware Cloudlet placement for vehicular edge Computing systems, in: 2019 IEEE International Conference on Cloud Computing Technology and Science, CloudCom, 2019, pp.159.166. 6. M. Bouet, V. Conan, Mobile edge Computing resources optimization: a geo-clustering approach, IEEE Trans. Netw. Serv. Manag. 15 (2) (2018)787–796. 7. L. Chen, J. Wu, G. Zhou, L. Ma, QUICK: QoS-guaranteed efficient Cloudlet placement in wireless metropolitan area networks, J. Supercomput. 74 (8) (2018).

[8] G. Cui, Q. He, F. Chen, H. Jin, Y. Yang, Trading off between user coverage and network robustness for edge server placement, IEEE Trans. Cloud Comput. (2020) 1, http://dx.doi.org/10.1109/TCC.2020.3008440.

[9]R.A.C. da Silva, N.L.S. da Fonseca, On the location of fog nodes in fog-Cloud infrastructures, Sensors 19 (11) (2019) 2445.

[10] Q. Fan, N. Ansari, Cost Aware Cloudlet Placement for big data processing at the edge, in: 2017 IEEE International Conference on Communications, ICC, 2017, pp. 1–6.

[11]J. Gedeon, M. Stein, J. Krisztinkovics, P. Felka, K. Keller, C. Meurisch, L. Wang, M. Mⁿhlh∑user, From cell towers to smart street lamps: placing Cloudlets on existing urban infrastructures, in: 3rd ACM/IEEE Symposium on Edge Computing (SEC 2018), IEEE, 2018, pp. 182–202.

[20] X. Guan, X. Wan, T. Wang, Y. Li, A long-term cost-oriented Cloudlet planning method in wireless metropolitan area networks, Electronics 8 (11) (2019) 1213, URL https://www.mdpi.com/2079-9292/8/11/1213.

[12]Y. Guo, S. Wang, A. Zhou, J. Xu, J. Yuan, C.-H. Hsu, User allocation-aware edge Cloud placement in mobile edge Computing, Softw. - Pract. Exp.
(2019) 1–14.

[13] M. Jia, J. Cao, W. Liang, Optimal Cloudlet placement and user to Cloudlet allocation in wireless metropolitan area networks, IEEE Trans. Cloud Comput. 5 (4) (2017) 725–737.

[14] J. Jiao, L. Chen, X. Hong, J. Shi, A heuristic algorithm for optimal facility placement in mobile edge networks, KSII Trans. Internet Inf. Syst. 11 (7) (2017) 3329–3350.
[15] S. Kang, R. Linna, S. Guo, W. Li, X. Qiu, Geographic clustering based mobile edge Computing resource allocation optimization mechanism, in: 15th International Conference on Network and Service Management, CNSM2019, 2019
[16] I. Leyva-Pupo, A. Santoyo-Gonzßlez, C. Cervell≤-Pastor, A framework for the joint placement of edge service infrastructure and user plane functions for
[17]5G, Sensors 19 (18) (2019) 3975.
[18] Y. Li, S. Wang, An energy-aware edge server placement algorithm in mobile edge Computing, in: 2018 IEEE International Conference on Edge Computing (EDGE), IEEE,
2018, pp. 66–73.
[19] B. Li, K. Wang, D. Xue, Y. Pei, K-means based edge server deployment algorithm for edge Computing environments, in: 2018
[20] IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud
& Big Data Computing, Internet of People and Smart City Innovation
(SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2018,
pp. 1169–1174.
[21] J. Liu, U. Paul, S. Troia, O. Falowo, G. Maier, K-means based spatial base station clustering for facility location problem in 5G, in: J. Lewis, Z. Ndlela (Eds.), Proceedings of Southern Africa Telecommunication Networks and Applications Conference, SATNAC, 2018, pp. 406–409.
[22] L. Ma, J. Wu, L. Chen, Z. Liu, Fast algorithms for capacitated Cloudlet placements, in: IEEE 21st International Conference on Computer Supported Cooperative Work in Design, 2017, pp. 439–444.
[23] A. Machen, S. Wang, K.K. Leung, B.J. Ko, T. Salonidis, Live service migration in mobile edge Clouds, IEEE Wirel. Commun. 25 (1) (2018) 140–147.
[24] J. Meng, W. Shi, H. Tan, X. Li, Cloudlet placement and minimum-delay routing in Cloudlet Computing, in: 3rd International Conference on Big Data Computing and
Communications, IEEE, 2017, pp. 297–304.
[25] S. Wang, Y. Zhao, J. Xu, J. Yuan, C.-H. Hsu, Edge server placement in mobile edge Computing, J. Parallel Distrib. Comput. 127 (2019) 160–168.
[26] Z. Xu, W. Liang, W. Xu, M. Jia, S. Guo, Efficient algorithms for capacitated Cloudlet placements, IEEE Trans. Parallel Distrib. Syst. 27 (10) (2016) 2866–2880.
[27] H. Yao, C. Bai, M. Xiong, D. Zeng, Z. Fu, Heterogeneous Cloudlet deployment and user-Cloudlet association toward cost effective fog Computing, Concurr. Comput.: Pract. Exper.

[28] H. Yin, X. Zhang, H.H. Liu, Y. Luo, C. Tian, S. Zhao, F. Li, Edge provisioning with flexible server placement, IEEE Trans. Parallel Distrib. Syst. 28 (4) (2017) 1031–1045.
[29].F. Zeng, Y. Ren, X. Deng, W. Li, Cost-effective edge server placement in wireless metropolitan area networks, Sensors 19 (1) (2018).

Résumé

Le Fog Computing est apparu comme une technologie favorable qui peuvent rapprocher les applications Cloud de la physique Appareils IoT à la périphérie du réseau, mais il n'y a ni un commun architecture de calcul de Cloud ou comment elle prend en charge Internet en temps réel exécution du service des objets (IoT). Les périphériques tels que le Commutateur, routeur, passerelle, téléphones portables, voiture intelligente, etc., sont les candidats au déploiement de nœuds de Cloud mais le déploiement diffère selon l'application. Dans ce travail, nous avons pris passerelles en tant que candidats au déploiement de nœuds de Cloud. La passerelle collecte des données à partir de capteurs intelligents, mais il n'en a pas capacités de prétraitement ou de prise de décision. Par conséquent, la Passerelle est rendue plus intelligente avec les capacités Fog et nommée comme Fog Smart Gateway (FSG). Le traitement du trafic IoT est pris en charge par des Machines Virtuelles (VM) facilitées par le Fog distribué nœuds. Nous avons optimisé le nombre de nœuds de Cloud pour le déploiement réduire la latence totale induite par l'agrégation du trafic et en traitement. Nos résultats montrent que le déploiement optimal du Cloud les nœuds du réseau IoT pourrait entraîner une réduction de la latence par rapport au traitement des données IoT dans un système Cloud conventionnel.

Passer de la théorie à la pratique dans les réseaux de Cloud pose la question du nombre optimal de nœuds de Cloud qui seront mis à niveau à partir des nœuds existants. Ce mémoire trouve le nombre optimal de nœuds de Cloud pour un nombre total de nœuds ordinaires résidant dans la zone d'intérêt pour différentes conditions de canal. La détermination du nombre optimal de nœuds de Cloud est très bénéfique, car il peut fortement affecter le SINR, et donc le moyen débit de données et délai de transmission. Les résultats numériques indiquent que le débit de données moyen augmente de près d'un ordre de magnitude pour un nombre optimisé de nœuds de Cloud en cas d'ombrage et d'évanouissement. Ce travail étudie le déploiement dynamique pondéré de dispositifs mobiles de calcul de Cloud pour prendre en charge une informatique de pointe mobile environnement, dans lequel chaque périphérique est associé à un poids pour refléter son importance en fonction de l'application. Puisque les appareils périphériques sont mobiles et peuvent être éteints, il est difficile d'optimiser dynamiquement le déploiement pour s'adapter à l’aspect dynamique du problème. Ce travail modélise davantage le problème mathématiquement et le résout par un algorithme inspiré des chauves-souris (BA), qui

recherche les solutions optimales en simulant le comportement de recherche de nourriture des chauves-souris via l'écholocation. De plus, trois méthodes de recherche locale conçus spécifiquement pour ce problème sont intégrées dans le BA, et un mécanisme de sélection de recherche locale dynamique est utilisé pour ajuster les probabilités de choisir les trois méthodes de recherche locale de manière itérative dans la boucle principale de BA. Les résultats de la simulation montrent surperformance de la BA proposée par rapport à la BA sans recherche locale et l'approche précédente.

Mots clés: Fog Computing. Bord mobile. Algorithme inspiré de la nature. Déploiement. Dynamique. Graphique pondéré par les nœuds, Edge devices, Fog node, Service Latence, machines virtuelles.

Abstract

Fog Computing has emerged as a favorable technology that can bring Cloud applications closer to physical IoT devices at the network edge, but there is neither common fog Computing architecture or how it supports real-time internet. Execution of the service of objects (IoT). Peripherals such as peripherals such as the switch, router, gateway, mobile phones, smart car, etc., are the Candidates for deploying fog nodes, but the deployment differs by application. In this work, we took gateways as candidates for the deployment of fog nodes. The gateway collects data from smart sensors, but it does not have any pre-processing or decision-making capabilities. Therefore, the Gateway is made smarter with Fog capabilities and named as Fog Smart Gateway (FSG). IoT traffic processing is supported by Virtual Machines (VMs) facilitated by distributed Fog nodes. We have optimized the number of fog nodes to reduce the total latency induced by traffic aggregation and Processing. Our results show that the optimal deployment of fog IoT network nodes could result in reduced latency compared to processing IoT data in a conventional Cloud system.

Moving from theory to practice in fog networks poses the question of the optimal number of fog nodes that will be upgraded from existing nodes. This paper finds the optimal number of fog nodes for a total number of ordinary nodes residing in the area of interest for different channel conditions. Determining the optimal number of fog nodes is highly beneficial, as it can strongly affect the SINR, and thus the average data rate and transmission delay. The numerical results indicate that the average data rate increases by almost an order of magnitude for an optimized number of fog nodes under shading and fading. It is further shown that the optimal number of fog nodes does not increase in direct proportion to the increase in the total number of nodes. Moreover, the optimal number of fog nodes decreases when the channels have high path loss exponents. These results suggest that fog nodes should be selected among those with the highest computational capacity for densely deployed networks and channels with high path loss exponents. Index terms Fog networking, hierarchical networks, SINR, average throughput, transmission delay.

This work investigates the weighted dynamic deployment of mobile fog-Computing devices to support a mobile edge Computing environment, in which each peripheral device is associated with a weight to reflect its importance depending on the application.

Since peripheral devices are mobile and can be switched off, it is difficult to dynamically optimize the deployment to adapt to the dynamic currency. This work further models the problem mathematically and solves it with a bat-inspired (BA) algorithm, which finds the optimal solutions by simulating the foraging behavior of bats via echolocation. Moreover, three local search methods designed specifically for this problem are integrated into the BA, and a dynamic local search selection mechanism is proposed to adjust the probabilities of choosing the three local search methods iteratively in the main loop of BA. The simulation results show outperformance of the proposed BA compared to the BA without local search and the previous approach.

Keywords: Fog Computing. Movable edge. Algorithm inspired by nature. Deployment. Dynamic. Graph weighted by nodes, Edge devices, Fog node, Service Latency, virtual machines.

الملخص

ظهرت الحوسبة الضبابية كتقنية مواتية يمكن أن تجعل التطبيقات السحابية أقرب إلى أجهزة إنترنت الأشياء المادية على حافة الشبكة ، ولكن لا توجد بنية حوسبة ضبابية شائعة أو كيف تدعم الإنترنت في الوقت الفعلي. تنفيذ خدمة الأشياء (IoT).الأجهزة الطرفية مثل المحول ، والموجه ، والبوابة ، والهواتف المحمولة ، والسيارة الذكية ، وما إلى ذلك ، هي المرشحون لنشر عُقد الضباب ، لكن النشر يختلف باختلاف التطبيق. في هذا العمل ، اتخذنا البوابات كمرشحين لنشر عُقد الضباب. تقوم البوابة بجمع البيانات من أجهزة الاستشعار الذكية ، ولكنها لا تحتوي على أي قدرات معالجة مسبقة أو اتخاذ القرار. لذلك ، أصبحت البوابة أكثر ذكاءً من خلال إمكانات الضباب وسميت باسم .Fog Smart Gateway (FSG) يتم دعم معالجة حركة مرور إنترنت الأشياء من خلال الأجهزة الافتراضية (VMs) التي تسهلها عُقد الضباب الموزعة. لقد قمنا بتحسين عدد عقد الضباب لتقليل زمن الوصول الإجمالي الناجم عن تجميع حركة المرور والمعالجة. تظهر نتائجنا أن النشر الأمثل لعقد شبكة إنترنت الأشياء الضبابية يمكن أن يؤدي إلى تقليل زمن الوصول مقارنة بمعالجة بيانات إنترنت الأشياء في نظام سحابي تقليدي.

يطرح الانتقال من النظرية إلى الممارسة في شبكات الضباب مسألة العدد الأمثل لعقد الضباب التي سيتم ترقيتها من العقد الحالية. تجد هذه الورقة العدد الأمثل لعقد الضباب لعدد إجمالي من العقد العادية المقيمة في منطقة الاهتمام لظروف القناة المختلفة. يعد تحديد العدد الأمثل لعقد الضباب مفيدًا للغاية ، حيث يمكن أن يؤثر بشدة على SINR ، وبالتالي متوسط معدل البيانات وتأخير الإرسال. تشير النتائج العددية إلى أن متوسط معدل البيانات يزداد بما يقارب الترتيب من حيث الحجم لعدد مُحسَّن من عُقد الضباب تحت التظليل والبهت. ويتضح كذلك أن العدد الأمثل لعقد الضباب لا يزيد بالتناسب المباشر مع الزيادة في العدد الإجمالي للعقد. علاوة على ذلك ، يتناقص العدد الأمثل لعقد الضباب عندما يكون للقنوات قيم عالية لفقدان المسار. تشير هذه النتائج إلى أنه يجب اختيار عُقد الضباب من بين تلك التي تتمتع بأعلى قدرة حسابية للشبكات والقنوات المنتشرة بكثافة ذات الأسس العالية لفقدان المسار. مصطلحات الفهرس شبكات الضباب ، الشبكات الهرمية ، SINR، متوسط الإنتاجية ، تأخير الإرسال.

يبحث هذا العمل في النشر الديناميكي المرجح لأجهزة الحوسبة الضبابية المتنقلة لدعم بيئة حوسبة الحافة المتنقلة ، حيث يرتبط كل جهاز طرفي بوزن ليعكس أهميته اعتمادًا على التطبيق. نظرًا لأن الأجهزة الطرفية متحركة ويمكن إيقاف تشغيلها ، فمن الصعب تحسين النشر ديناميكيًا للتكيف مع العملة الديناميكية. يقوم هذا العمل بنمذجة المشكلة رياضيًا بشكل أكبر ويحلها باستخدام خوارزمية مستوحاة من الخفافيش (BA) ، والتي تجد الحلول المثلى من خلال محاكاة سلوك البحث عن الطعام للخفافيش عبر تحديد الموقع بالصدى. علاوة على ذلك ، تم دمج ثلاث طرق بحث محلية مصممة خصيصًا لهذه المشكلة في مكتبة الإسكندرية ، وتم اقتراح آلية اختيار بحث محلي ديناميكي لضبط احتمالات اختيار طرق البحث المحلية الثلاثة بشكل متكرر في الحلقة الرئيسية لـ .BA أظهرت نتائج المحاكاة تفوقًا في أداء مكتبة الإسكندرية المقترحة مقارنة بمكتبة الإسكندرية دون البحث المحلي والمنهج السابق.

الكلمات الرئيسية: حوسبة الضباب. حافة متحركة. خوارزمية مستوحاة من الطبيعة. تعيين. متحرك. رسم بياني مرجح بالعقد ، أجهزة Edge ، عقدة الضباب ، زمن انتقال الخدمة ، الأجهزة الافتراضية.