Université Assane SECK de Ziguinchor

UFR Sciences et technologies

Département Informatique

MEMOIRE DE FIN D'ETUDES

Pour l'obtention du diplôme de Master

Mention : Informatique

Spécialité : Réseaux et Systèmes

SUJET :

Intégration du fog computing et du cloud aux données IoT : comparaison du temps de latence et du stockage sur ifogsim et raspberry pi.

Présenté par : **M. Moustapha THIAM**

Soutenu le : 04 Mars 2023

Sous la direction de : **Dr. Madiop DIOUF** et **Dr. Thierno Ahmadou DIALLO**

Sous la supervision de : **Professeur Youssou Dieng**

Devant le jury composé de :

M. Youssou DIENG	Professeur assimilé (MC CAMES)	**Président**
M. El. Malick NDOYE	Maître conférence	**Rapporteur**
M. Malaw NDIAYE	Assistant	**Rapporteur**
M. Madiop DIOUF	Maître conférence assimilé	**Encadrant**
M. Thierno Ahmadou DIALLO	Maitre conférence	**Co-encadrant**

Année : 2021/2022

DEDICACES

Je dédie ce mémoire aux Plus belles créatures que DIEU a créées sur terre pleine de source, de tendresse, de patience et de générosité :

Mon défunt Oncle EL Hadj Djim Lo ;

Ma défunte Grand-Mère Mame Ndeye Bitèye ;

Mon père, Omar Thiam ;

Mon Oncle Moustapha Sokhna ;

Ma mère Awa Sokhna ;

Qui ont œuvré pour ma réussite, de par leurs amours, leurs soutiens, tous leurs sacrifices consentis et leurs précieux conseils, pour toutes leurs assistances et leurs présences dans ma vie. Qu'ils reçoivent à travers ce modeste travail, l'expression de mes sentiments et de mon éternelle gratitude.

REMERCIEMENTS

ALHAMDOULILAH tout d'abord, je remercie *Allah, Le tout –miséricordieux, Le très – miséricordieux*, de m'avoir guidé à achever ce travail.

Je tiens à remercier en premier lieu et à exprimer toute ma reconnaissance à mon encadreur, **Dr Madiop DIOUF** qui a su partager son savoir faire, ses connaissances son expérience, ses idées et son temps pour me porter aide pendant et hors de ses heures de travail à la fois compréhensible et serviable. J'aimerais adresser plus qu'un merci pour ses bons conseils, ses bons réflexes, sa disponibilité et surtout pour tout le savoir qu'il m'a transmis. Enfin, veuillez recevoir mes chaleureux remerciements pour tout !

Je tiens aussi à remercier **les membres du jury**.

Je tiens à remercier aussi le Co-encadreur **Dr Thierno Ahmadou DIALLO** pour tout son soutien et ses conseils.

Nous remercions aussi l'ensemble des professeurs qui ont participé à notre formation et à l'ensemble du corps enseignant et du personnel administratif et technique de l'**UFR ST** pour leur disponibilité.

Nous tenons aussi à remercier nos parents, frères et sœurs pour tous les sacrifices consentis, ainsi que leur soutien indéfectible.

Je remercie ma grande sœur Fatoumata Zahra Thiam, mon frère Babacar Thiam, mon ami et cousin Babacar Bitèye, ma tante Badiane Marie Thiam, mes oncles Dr Lamine Sokhna, El Hadj Sokhna, mes grand-mères Aida Lo et Marie Diouf, mes très chers amis Ousseynou Souaré, Mamadou Kane, Ibrihama Sabaly, Diariyatou Niang, Daouda Seydi, El hadji Ndiaye Diallo, Amadou Thiam, Khamad Thiaw, Ndoumbé Gningue, Birame Ndoye, Meissa Gueye, Taha Maiga, mes cousins et cousines Aïssatou Lo, Ndeye Marie Sokhna, Moustapha Sokhna, Hady Lo, Papa Omar Sokhna et notre cher candidat Président Ousmane Sonko (PROS).

Je remercie aussi mes parents, surtout mon oncle et mon très cher papa Cheikh Tidiane Thiam et sa femme Aminata Sokhna pour leurs soutiens, tous leurs sacrifices consentis et leurs précieux conseils, pour toutes leurs assistances et leurs présences dans ma vie.

Enfin, j'adresse mes remerciements toute personne qui m'a aidée pour réaliser ce travail.

Glossaire des Acronymes

IOT : Internet of Things

FC : Fog Computing

TIC : Technologies de l'information et de la communication

QoS : Quality of Service

IDC : international Data Corporation

ONU : Organisation des Nations unies

RFID : radio-identification

3G : Troisième Génération de réseau

2G : Deuxième Génération de réseau

5G : Cinquième Génération de réseau

4G : Quatrième Génération de réseau

LTE-M : Long Term Evolution for Machines

NB-IoT : Narrowband IoT

LPWAN : Low-power Wide-area Network

NFC : Near Field Communication

API : interface de programmation d'application

AVR : Automatic Voltage Regulator

GSM : Global System for Mobile

UMTS : Universal Mobile Telecommunications System

LTE *:* Long Term Evolution

WWAN : Wireless Wide Area Network

Wi-Fi : Wireless Fidelity

AES : Advanced Encryption Standard

CDMA : Code Division Multiple Access

TDMA : Temporal Division Multiple Access

WCDMA : Wide Band Code Division Accès multiple

FDMA : Frequency-Division Multiple Access

BLR : Boucle Locale Radio

WMAN : Wireless Metropolitan Area Network

IEEE : Institute of Electrical and Electronics Engineers

WiMax : World wide Interoperability

WSN : Wireless Sensor Networks

ARM : Advanced RISC Machine

POP : Points de Présence d'Opérateur

IP : Internet Protocol

NIST : National Institute of Standards and Technology

MQTT : Message Queuing Telemetry Transport

CoAP : Constrained Application Protocol

LAN : Local Area Network

WAN : Wide Area Network

RAM : Random access memory

IDE : integrated development environment

CPU : Central Processing Unit

ICSP : In-circuit serial programming

USB : Universal Serial Bus

HDMI : high-definition multimedia interface

GPIO : General Purpose Input/Output

SQL : Structured Query Language

LED : Light Emitting Diode

ISM : Industriel Scientifique Médical

VCC : Tension en Courant Continu

Résumé :

L’être humain a besoin de se nourrir du jour au lendemain, depuis longtemps il le fait grâce à l’agriculture, qui est devenu de nos jours plus intelligente. Avec l’augmentation rapide de la population mondiale, atteindre l’autosuffisance alimentaire est devenue un véritable défi à relever dans certains pays africains. Par conséquent faire de telle sorte que l’agriculture va à une vitesse éclaire serai un véritable exploit pour les agriculteurs. De nombreux chercheurs ont essayé de porter leurs connaissances sur l’agriculture intelligente. Des travaux importants ont été réalisés dans les domaines des réseaux de capteurs sans fil (WSN), du cloud Computing, du Fog Computing et de l'Internet des objets (IoT) pour faciliter les progrès des services de l’agriculture. Et pour appliquer le Fog dans cette domaine agricole nous avons jugé nécessaire de porter notre travail sur ce sujet. Au moment même l’utilisation de la bande passante et la latence sont les préoccupations les plus importantes dans les applications basées sur le Cloud, y compris l’agriculture intelligente. Le Fog Computing, qui consiste à amener les ressources informatiques du nuage à proximité de la périphérie du réseau, permet non seulement de résoudre le problème de latence et de la bande passante, mais apporte également des améliorations significatives, telles que la mise à l'échelle à la demande, la mobilité des ressources et la sécurité. Après avoir démontrer les avantages du Fog, nous allons simuler dans ifogsim notre travail et nous allons vous montrer nos implémentations avec les microcontrôleurs et les résultats sur la latence et sur l’utilisation du réseau (la bande passante) sont plus économique sur l’architecture basée au niveau du Fog plutôt sur celle du Cloud.

Mots clés : Fog Computing, ifogsim, latence, Cloud, IoT.

Abstract

The human being needs to feed himself from one day to the next, for a long time he has been doing it thanks to agriculture, which has become more intelligent nowadays. With the rapid increase in the world population, achieving food self-sufficiency has become a real challenge in some African countries. Therefore, making agriculture go at a lightning speed would be a real feat for the farmers. Many researchers have tried to bring their knowledge to smart agriculture. Important work has been done in the areas of Wireless Sensor Networks (WSN), Cloud Computing, Fog Computing and Internet of Things (IoT) to facilitate the advancement of agriculture services. And to apply the Fog in this agricultural domain we found it necessary to focus our work on this topic. At the moment bandwidth usage and latency are the most important concerns in cloud-based applications, including smart agriculture. Fog computing, which involves bringing cloud computing resources close to the network edge, not only solves the latency and bandwidth problem, but also brings significant improvements, such as on-demand scaling, resource mobility and security. After demonstrating the benefits of Fog, we will simulate in ifogsim our work and show you our implementations with microcontrollers and the results on latency and network usage (bandwidth) are more economical on the Fog-based architecture rather than the Cloud-based one.

Keywords : Fog Computing, ifogsim, latency, Cloud, IoT.

Table des matières

Liste des Figures

Liste des Tableaux

Introduction Générale

L'agriculture joue un rôle fondamental dans le monde, à la fois comme source essentielle de moyens de subsistance et son rôle dans la chaîne d'approvisionnement alimentaire mondiale. Elle est à la base de la survie de l'humanité. Elle est un processus par lequel les êtres humains aménagent leurs écosystèmes et contrôlent le cycle biologique d'espèces domestiquées, dans le but de produire des aliments et d'autres ressources utiles à leurs sociétés.

Cependant, des facteurs comme la croissance démographique, l'expansion du développement industriel et le changement climatique, limitent le développement agricole [1].

Avec la prévision que l'ONU a fait sur la population mondiale, qui nous dit que la population mondiale attiendra 9,7 milliards d'habitants d'ici 2050 [2]. Et si ces prévisions se concrétisent donc la production mondiale augmentera d'environ 60% d'ici 2050 [3].

Et pour mieux améliorer la croissance des récoltes mondiale, nous allons essayer intégrer la technologie afin de proposer un système permettant de réduire le temps de latence dans les systèmes d'arrosage automatique en se basant sur le Fog Computing [2].

L'informatique appliquée à l'agriculture ou mieux l'agriculture intelligente utilise les nouveaux gadgets de la technologie (les objets connectés) pour optimiser les dégâts environnementaux. Les objets connectés, comme les capteurs d'humidité, de température nous donnent des valeurs exactes. Ces dernières permettent, ainsi l'automatisation de l'agriculture. Le système d'arrosage automatique en est un exemple.

Ces capteurs permettent ; d'améliorer un certain nombre de choses à savoir, la prévision des récoltes, les prévisions météorologiques, d'augmenter de la production, la durée de conservation de l'eau ; de collecter des données en temps réel, de produire et réduire les coûts d'exploitation ; de surveillance des équipements, et d'évaluer de façon précise les exploitations et des champs.

Aujourd'hui nous sommes à la quatrième révolution industrielle liée à l'avènement de l'internet des objets (IOT), malgré les nombreux avantages des technologies précédentes, le Cloud Computing suscite l'objet de discussion ; la quatrième révolution est considérée comme l'ère de l'intelligence et leurs aptitudes à communiquer avec d'autres objets, intégrant ainsi, plusieurs technologies et solutions de communication.

Actuellement, avec l'avancé des technologies de l'information et de la communication, les gens ont tendance à les utiliser dans divers secteurs tels que la santé, le transport, de l'agriculture etc.

L’utilisation de l’internet des objets génère beaucoup de données qui sont ensuite stockées la plupart du temps dans un serveur Cloud. Le Fog Computing permet de stocker des données en local, en réduisant la consommation d'énergie et y accéder en un temps record.
Les technologies agricoles intelligentes sont déjà utilisées dans de nombreux pays comme approche, en utilisant les technologies de l'information et de la communication (TIC) pour résoudre les problèmes ruraux tels que la pénurie de main-d'œuvre mentionnée ci-dessus [4, 5].

Le Fog Computing (FC), une technologie récente qui a amélioré la qualité de service (QoS) pour les TIC modernes dans la 4e révolution industrielle. Par contre, le Cloud Computing traite toutes les données au niveau d'un serveur central, les nœuds Fog sont ainsi distribués en FC prenant en charge le traitement sur un serveur central [6].

Ce mémoire est organisé en quatre chapitres :

- Dans le premier chapitre, nous allons donner les généralités et la justification de notre sujet.
- Dans le deuxième chapitre, nous abordons le paradigme de l’Internet des Objets.
- Dans le troisième chapitre, nous allons faire une étude sur le Fog Computing.
- Et dans le dernier chapitre, nous présentons l’implémentation et l’expérimentation réalisées pour valider la solution présentée dans le chapitre trois.

Enfin, on conclut puis donne les perspectives que l’on souhaite réaliser dans le futur.

Chapitre I : Généralités et Justification du Sujet

I-1 Contexte

Pour faire face à l'évolution des applications IoT, le Fog Computing s'est imposé comme une alternative pour éviter la charge des data Center et du réseau dans les infrastructures cloud.

L'internet des objets (IoT) est l'infrastructure qui sert à connecter tout objet à internet, tels que des dispositifs portables, des capteurs, des smartphones, des appareils photos, des appareils électroménagers et des véhicules [7. 8].

Pour étendre le cloud vers la périphérie du réseau, le Fog Computing est capable de supporter les applications IoT distribuées géographiquement, sensibles à la latence ou gourmandes en bande passante. Notons aussi qu'au cours de ces dernières années, le nombre d'objets connectés n'a cessé d'augmenter. Avec la prédiction d'IDEMIA (société française de sécurité numérique spécialisée dans la biométrique, l'identification, l'analyse de données et de vidéos) qui informe d'ici 2025 le nombre d'IoT dépassera 2 zettaoctets de données [9], avec ses données qui vont être générées, l'international Data Corporation estime que (IDC) les IoT produisent 27% des données mondiales [10].

De nos jours les données générées par les IoT sont stockés et traitées dans Cloud [11, 12]. Ce dernier satisfait la majorité des besoins des applications d'IoT en termes d'accès ubiquitaire, de disponibilité et d'évolutivité des performances de traitement et de capacité de stockage. Cependant, le Cloud étant un paradigme centralisé dans des centres de données, les données produites et les requêtes émises par les objets connectés sont transmises à ces centres situés dans le cœur de réseau.

Ainsi, avec le nombre croissant d'objets connectés et la quantité de données produites, l'envoi de toutes les données vers le Cloud générera des goulots d'étranglement [12]. Ceci augmente la latence de transmission de données et, par conséquent, dégrade la qualité de service (QoS) [13]. De nombreuses applications de l'IoT telles que l'Internet des Véhicules (IoV), la télémédecine et l'industrie 4.0 sont très sensibles aux latences. Le moindre retard dans les réponses du système peut entraîner des problèmes critiques [12].

Du fait des problèmes de latence et de trafic réseau générés par le Cloud, le paradigme de Fog Computing a émergé [14.]. Le Fog permet de décentraliser les services traditionnellement fournis par le Cloud sur l'ensemble des équipements réseaux. Il permet d'utiliser les ressources de calcul et de stockage des équipements réseaux tels que les routeurs et les switches localisés

entre les objets connectés et les centres de données du Cloud [15]. Le Fog Computing permet le filtrage, l'agrégation et le traitement des données dans les périphéries du réseau, offrant une réduction de la latence, la préservation de la bande passante du réseau et l'amélioration de la qualité de service (QoS) [16.].

I-2 Problématique

Le Fog Computing est une architecture de calcul qui permet de déplacer les applications et les services de calcul vers le bord du réseau, c'est-à-dire vers des dispositifs de bord de réseau tels que des routeurs, des passerelles, des caméras de surveillances, etc. Le but est de traiter les données de manière plus efficace en les traitant au plus près de leur source, plutôt que de les envoyer vers le Cloud pour le traitement.

Cependant le Fog Computing peut être utile dans les situations ou il y a un grand volume de données à traiter et ou la **latence** et **la bande passante** sont des facteurs importants par exemple dans les applications de réalités augmentée ou de réalité virtuelle, ou dans les applications de contrôle industriel en temps réel.

a- Minimisation de la latence du système dans les infrastructures de Fog

La minimisation de la latence du système est un enjeu crucial dans les infrastructures de Fog Computing. En effet, la latence peut avoir un impact important sur la qualité de service des applications et sur l'efficacité globale du système.
Comme mentionné précédemment, le Fog Computing présente une infrastructure hétérogène et géo-distribuée. Le mauvais placement des données dans une telle infrastructure peut entraîner une augmentation de la latence globale du système.

b- Minimisation de la bande passante du système dans les infrastructures de Fog

La minimisation de la bande passante est un enjeu important dans les infrastructures de Fog Computing, car la bande passante disponible peut être limitée dans certaines situations. Si la bande passante est insuffisante, cela peut entrainer des retards dans la transmission des données et une mauvaise qualité de service des applications.

I-3 Objectifs et Contribution

Dans notre travail de recherche nous allons essayer d'explorer les possibilités offertes par le Fog Computing pour améliorer la minimisation de la latence et de la bande passante des applications de l'Internet des objets (IoT). Nous nous concentrerons sur les mécanismes permettant de déplacer le traitement et le stockage des données vers les appareils de bord de réseau afin de réduire les temps de réponse et la latence, ainsi que les couts de bande passante et stockage. Nous cherchons également à développer des approches permettant d'optimiser la bande passante utilisée dans les applications en répartissant le traitement et le stockage sur de nombreux appareils de bord de réseau.

Nous avons pour objectif est de faire une simulation pratique avec ifogsim sur 30 capteurs humidités au sol. Les capteurs vont récupérer puis envoyer les données au Fog et au Cloud afin d'être traitées et analysées. Ainsi le résultat pourra être visualisé pour ensuite conclure sur l'option la plus optimale en termes de latence et de bande passante. En effet, cette simulation nous a permis de faire un glissement sur la mise en place d'un prototype simple composé de matériels IoT, de microcontrôleurs pour avoir des résultats concrets sur la latence avec comme champs d'application le Fog Computing et le Cloud Computing.

Conclusion

Ce premier chapitre nous a permis de faire un tour sur le cadrage théorique qui lie notre sujet. Ainsi, les lignes qui vont suivre nous permettrons de mettre globalement l'accent sur le paradigme de l'internet des objets.

Chapitre II : Paradigme d'Internet des Objets

II-1 Introduction

Il n'existe pas de définition standard et unifiée de l'internet des objets, certaines définitions traitent les aspects techniques de l'IoT, alors que d'autres évoquent les usages et les fonctionnalités.

La technologie IoT est considérée comme l'émergence de l'Internet du futur, Certains la définissent comme des « objets ayant des identités et des personnalités virtuelles, opérant dans des espaces intelligents et utilisant des interfaces intelligentes pour se connecter et communiquer au sein de contextes d'usages variés ». [1]

D'autres, insistent sur l'aspect ubiquitaire de l'IoT permettant de connecter les gens et les objets n'importe où, n'importe quand, par n'importe quoi. Ce nouveau paradigme informatique est basé non plus sur des PC et des périphériques informatiques, mais sur des objets quotidiens intégrant des capteurs en leurs attribuant une intelligence et la capacité de communiquer via le réseau Internet. [14]

II-2 Enjeux et Défis

a- Enjeux

L'IoT engendre une valeur considérable pour les entreprises en pleine transformation numérique qui doivent collecter et traiter de grandes quantités de données pour comprendre la santé de leur activité, identifier les domaines à améliorer et optimiser les flux de travail pour de meilleurs résultats. Les dispositifs IoT sont des points de collecte de données autonomes qui peuvent fournir des informations instantanées historiques ou actualisées. Ils peuvent interagir avec d'autres systèmes de nombreuses façons, selon leur fonction. Certains peuvent envoyer des données périodiquement, d'autres peuvent recevoir des commandes, alors que d'autres peuvent déclencher des actions et des alertes. Cette polyvalence rend l'IoT essentiel à l'automatisation et à l'optimisation des processus métiers.

b- Défis

L'Internet des objets (IoT) s'est rapidement développé pour devenir une grande partie de la façon dont les êtres humains vivent, communiquent et font des affaires. Partout dans le monde, les appareils compatibles avec le Web transforment nos droits mondiaux en une plus grande zone de vie active. Il existe différents types de défis face à l'IoT.

b-1 Défis de sécurité dans l'IoT :

b-1-1 Absence de sécurité :

Bien que le cryptage soit un excellent moyen d'empêcher les pirates d'accéder aux données, il s'agit également de l'un des principaux défis de sécurité de l'IoT. Ces disques aiment les capacités de stockage et de traitement que l'on trouverait sur un ordinateur traditionnel. Le résultat est une augmentation des attaques où les pirates peuvent facilement manipuler les algorithmes qui ont été conçus pour la protection.

b-1-2 Brute Forcing et Risque de mots de passe par défaut

Des informations d'identification et des informations de connexion faibles rendent presque tous les appareils IoT vulnérables au piratage de mots de passe et à la force brute. Toute entreprise qui utilise des informations d'identification par défaut sur ses appareils expose à la fois son entreprise et ses actifs, ainsi que le client et ses précieuses informations, à un risque d'attaque par force brute.

b-2 Défi de conception en IoT :

b-2-1 Augmentation des coûts et des délais de mise sur le marché

Les systèmes embarqués sont légèrement limités par les coûts. Le besoin naît de conduire de meilleures approches lors de la conception des dispositifs IoT afin de gérer la modélisation des coûts ou le coût optimal avec des composants électroniques numériques. Les concepteurs doivent également résoudre le problème du temps de conception et mettre sur le marché le dispositif embarqué au bon moment.

b-2-2 Sécurité du système

Les systèmes doivent être conçus et mis en œuvre pour être robustes et fiables et doivent être sécurisés avec des algorithmes cryptographiques et des procédures de sécurité. Elle implique différentes approches pour sécuriser tous les composants des systèmes embarqués du prototype au déploiement.

b-3 Défis de déploiement dans l'IoT

b-3-1 Connectivité

C'est la principale préoccupation lors de la connexion d'appareils, d'applications et de plates-formes cloud. Les appareils connectés qui fournissent des informations et des informations utiles sont extrêmement précieux. Mais une mauvaise connectivité devient un défi lorsque des

capteurs IoT sont nécessaires pour surveiller les données de processus et fournir des informations.

b-3-2 Collecte et traitement des données

Dans le développement de l'IoT, les données jouent un rôle important. Ce qui est plus critique ici, c'est le traitement ou l'utilité des données stockées. Outre la sécurité et la confidentialité, les équipes de développement doivent s'assurer qu'elles planifient bien la manière dont les données sont collectées, stockées ou traitées dans un environnement.

II-3 Architecture des IoT

L'Internet des objets (IoT) fait maintenant partie de notre vie. Que ce soit sous forme de maisons intelligentes, voitures connectées, usines intelligentes ou villes connectées ; nous les objets connectés cette technologie de manière quotidienne. Cette technologie est là pour rester. Nous voyons clairement que le nombre d'objets connectés ainsi que les données produites est colossal.

Une architecture IoT structurée et efficace est donc nécessaire pour gérer un flux de données et d'objets grandissant.

L'architecture IoT comprend plusieurs briques de systèmes IoT connectés pour garantir que les données des objets générées par les capteurs sont collectées, stockées et traitées dans les big data warehouse et que les actionneurs des objets exécutent les commandes envoyées via une application utilisateur.

C'est un cadre qui définit les composants physiques, l'organisation fonctionnelle et la configuration du réseau, les procédures opérationnelles et les formats de données à utiliser. Cependant, il n'existe pas d'architecture de référence standard unique pour l'IoT car elle englobe une variété de technologies. Cela signifie qu'il n'y a pas un modèle simple qui peut être suivi pour toutes les implémentations possibles.

L'architecture IoT peut en fait varier considérablement en fonction de la mise en œuvre. Elle doit être suffisamment ouverte avec des protocoles ouverts pour pouvoir prendre en charge plusieurs applications réseau.

Dans la majeure partie des cas elle se compose de 4 blocs constitutifs :

- La scalabilité
- La fonctionnalité
- La disponibilité

- La maintenabilité

Même s'il n'existe pas d'architecture IoT unique universellement acceptée, le format le plus basique et le plus largement accepté est une **architecture IoT à quatre couches.**

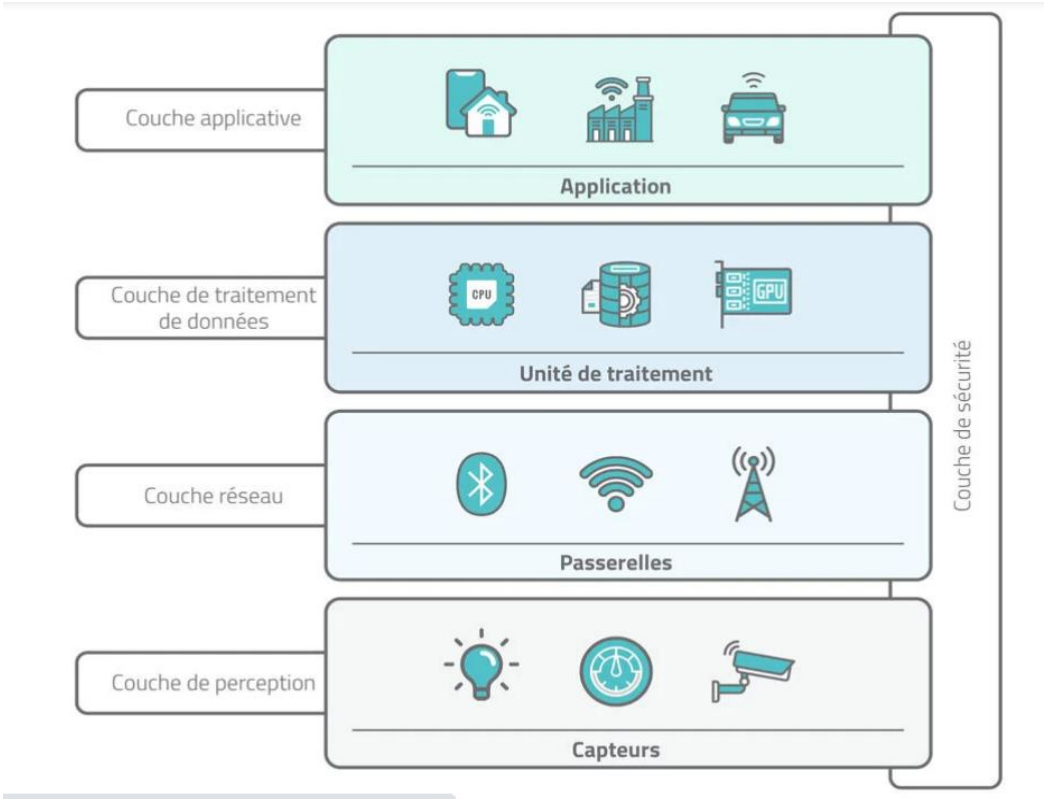


Figure II. 1 : Architecture des IoT

a- Couche de perception

Cette couche est responsable de convertir des signaux analogiques en données numériques et vice versa.

L'étape initiale de tout système IoT englobe un large éventail d'objets qui agissent comme un pont entre les mondes réel et numérique.

Leur forme et leur taille varient, des minuscules puces de silicium aux gros véhicules. Par leurs fonctions, les objets IoT peuvent être divisés en groupes :

- **Capteurs :** tels que sondes, jauges, compteurs et autres. Ils collectent des paramètres physiques tels que la température ou l'humidité, les transforment en signaux électriques et les envoient au système IoT. Les capteurs IoT sont généralement petits et consomment peu d'énergie.

- **Actionneurs :** traduisant les signaux électriques du système IoT en actions physiques. Les actionneurs sont utilisés dans les contrôleurs de moteur, les lasers, les bras robotiques.

- **Machines et dispositifs :** connectés à des capteurs et des actionneurs ou comme étant des parties intégrantes.

Il est important de noter que l'architecture n'impose aucune restriction sur la portée de ses composants ou leur emplacement. En effet la couche latérale peut inclure quelques objets physiquement placées dans une pièce ou des myriades de capteurs et d'appareils répartis dans le monde entier.

b- Couche réseau

Les données collectées par tous ces appareils doivent être transmises et traitées. C'est le travail de la couche réseau, elle connecte donc ces appareils à d'autres objets intelligents, serveurs et appareils réseau. Elle gère également la transmission de toutes les données.

Les communications entre les appareils et les services cloud ou les passerelles impliquent différentes technologies :

- **Ethernet :** connecte des appareils IoT fixes tels que des caméras de sécurité et vidéo, des équipements industriels installés en permanence.

- **Réseaux cellulaires :** Les plus connus actuellement étant la 5G et la 4G, elles offrent un transfert de données fiable et une couverture presque mondiale.
  Il aussi existe deux normes cellulaires développées spécifiquement pour les objets IoT. LTE-M (Long Term Evolution for Machines) permet aux appareils de communiquer directement avec le cloud et d'échanger de gros volumes de données. NB-IoT ou Narrowband IoT utilise des canaux basse fréquence pour envoyer de petits paquets de données.

- **LPWAN (Low-power Wide-area Network) :** a été créée spécifiquement pour les appareils IoT. Cette technologie offre une connectivité sans fil longue portée avec une faible consommation d'énergie avec une autonomie de plus de 10 ans. Nous vous conseillons fortement de consulter notre comparatif sur les différentes technologies LPWAN.

- **WiFi :** la technologie de réseau sans fil la plus populaire, convient parfaitement aux solutions IoT gourmandes en données, faciles à recharger et à utiliser dans une petite zone. Un bon exemple d'utilisation est celui des appareils domestiques intelligents connectés au réseau électrique.

Il existe aussi d'autres technologies permettant la connectivité comme notamment le NFC (Near Field Communication), le Bluetooth ou le ZigBee.

c- Couche de traitement de données

La couche de traitement accumule, stocke et traite les données provenant de la couche précédente. Toutes ces tâches sont généralement traitées via des plateformes IoT et comprennent deux étapes principales.

Étape d'accumulation des données

Les données en temps réel sont capturées via une API et mises au repos pour répondre aux exigences des applications non temps réel.
L'étape du composant d'accumulation de données fonctionne comme une plaque tournante entre la génération de données basée sur les événements et la consommation de données basée sur les requêtes.
Entre autres choses, l'étape définit si les données sont pertinentes pour les besoins de l'entreprise et où elles doivent être placées.
Elle enregistre les données dans une large gamme de solutions de stockage, des data lakes capables de contenir des données non structurées telles que des images et des flux vidéo. L'objectif global étant de trier une grande quantité de données diverses et de les stocker de la manière la plus efficace.

Étape d'abstraction des données

Ici, la préparation des données est finalisée afin que les applications grand public puissent les utiliser pour générer des informations. L'ensemble du processus comprend les étapes suivantes :

- Combiner des données provenant de différentes sources, à la fois IoT et non-IoT

- Concilier plusieurs formats de données

- Agréger les données en un seul endroit ou les rendre accessibles quel que soit leur emplacement grâce à la virtualisation des données.

De même, les données collectées au niveau de la couche application sont reformatées ici pour être envoyées au niveau physique afin que les appareils puissent les comprendre.
Ensemble, les étapes d'accumulation et d'abstraction des données masquent les détails du matériel, améliorant ainsi l'interopérabilité des appareils intelligents.

d- Couche Applicative

La couche application est ce avec quoi l'utilisateur interagit. C'est ce qui est chargé de fournir des services spécifiques à l'application à l'utilisateur.
Actuellement, les applications peuvent être construites directement sur les plates-formes IoT qui offrent une infrastructure de développement logiciel avec des outils prêts à l'emploi pour l'exploration de données, l'analyse avancée et la visualisation de données.
Sinon, les applications IoT utilisent des API pour s'intégrer à la couche précédente.

II-4 Dispositifs des IoT

a- Capteurs

Généralement, les capteurs sont utilisés dans l'architecture des dispositifs Iot. Les capteurs sont utilisés pour détecter des objets et des appareils, etc. Un appareil qui fournit une sortie utilisable en réponse à une mesure spécifiée.
Le capteur atteint un paramètre physique et le convertit en un signal adapté au traitement (par exemple électrique, mécanique) des caractéristiques de tout dispositif ou matériau pour détecter la présence d'une grandeur physique particulière.
La sortie du capteur est un signal qui est converti en une forme lisible par l'homme comme des changements de caractéristiques, des changements de résistance, de capacité, d'impédance, etc. [17].

Figure II. 2 : capteurs des Iots

b- Connecteurs

Il existe trois types de fils de connexions :

b-1 Les fils monobrins

Les fils monobrins sont des accessoires indispensables pour prototyper un circuit sur une breadboard. L'utilisation des câbles monobrins est aisé avec les breadboard et permet une meilleure organisation du câblage d'un prototype qu'avec des fils souples.

En effet en plus d'être suffisamment rigide pour être plié dans une position et la garder, un fil monobrin va pouvoir être raccourci à votre guise et garder toute son utilité. Une grande longueur de fil peut ainsi être coupée en plusieurs longueurs plus petites et être parfaitement utilisable [18].

Figure II. 3: les connecteurs fils monobrins

b-2 fils souples mâle mâle.

Ce genre de fil est un accessoire indispensable pour faciliter les câblages avec vos breadboards et différentes cartes électroniques ou autre montages. Parfaitement adapté pour les connecteurs femelles avec espacement standard de 2.54mm [19].

Figure II. 4 : les connecteurs fils souples male male.

c- Microcontrôleurs

c-1 Arduino

Arduino est un circuit imprimé sur lequel se trouve un microprocesseur (calculateur) qui peut être programmé pour analyser et produire des signaux électriques, de manière à effectuer des tâches très diverses comme la charge de batteries, la domotique (le contrôle des appareils domestique (éclairage, chauffage...), le pilotage d'un robot, etc.

C'est une plateforme basée sur une interface entrée/sortie simple et sur un environnement de développement utilisant la technique du Processing/Wiring. Arduino peut être utilisé pour construire des objets interactifs indépendants (prototypage rapide), ou bien peut être connecté à un ordinateur pour communiquer avec ses logiciels. Les versions vendues actuellement sont préassemblées, des informations sont fournies pour ceux qui souhaitent assembler l'Arduino eux-mêmes.

Un module Arduino est généralement construit autour d'un microcontrôleur Atmel AVR (ATmega328 ou ATmega1280 pour les versions récentes, ATmega168 ou ATmega8 pour les plus anciennes), et de composants complémentaires qui facilitent la programmation et l'interfaçage avec d'autres circuits. Chaque module possède au moins un régulateur linéaire 5V et un oscillateur à quartz 16 MHz (ou un résonateur céramique dans certains modèles). Le microcontrôleur est préprogrammé avec un boot loader de façon à ce qu'un programmateur dédié ne soit pas nécessaire [20].

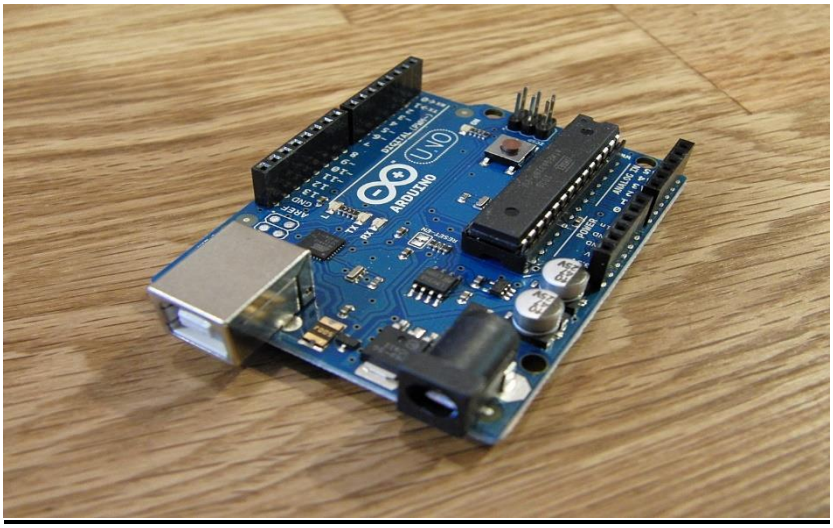


Figure II. 5 : microcontrôleurs Arduino

c-2 Raspberry Pi

Le Raspberry Pi est une curiosité technologique des plus fabuleuses. En effet, il s'agit d'un tout petit appareil faisant office d'ordinateur et ayant la forme d'une seule et simple carte mère. Que l'on puisse regrouper dans un espace aussi petit les capacités d'un ordinateur relève de l'ingéniosité et c'est précisément ce que recherchait David Braben en confectionnant le Raspberry Pi.

En le souhaitant pratique à manipuler de par sa taille et son faible poids, peu cher et facile à utiliser tout en préservant une très bonne performance de l'ensemble, il désirait offrir aux plus jeunes générations la possibilité de s'initier à l'informatique de manière ludique et efficace. De plus, le fait qu'il soit livré tel quel impose à l'utilisateur de construire peu à peu son ordinateur en ajoutant au fur et à mesure le matériel adéquat pour le faire fonctionner [21].

Ainsi en 2006, le tout premier prototype basé sur un microcontrôleur Atmel ATmega 644 était si simple qu'il aurait même été possible de le fabriquer à la maison avec un simple fer à souder.

[22].

Figure II. 6 : microcontrôleurs Raspberry Pi

II-5 Domaines d'applications

a- Domotique

La domotique regroupe l'ensemble des technologies permettant l'automatisation des équipements d'un habitat ou encore mieux elle peut être définit comme étant l'ensemble des techniques permettant de centraliser tous nos appareils de ménage via un smartphone.

Elle vise à apporter des fonctions de confort : commandes à distance, gestion d'énergie (optimisation de l'éclairage et du chauffage… etc.), sécurité (comme les alarmes) et de communication (contacts et discussion avec des personnes extérieures**)** [23].

Les services offerts par la domotique couvrent 3 domaines principaux :

1. Assurer la protection des personnes et des biens en domotique par la prévenir des risques d'accident (incendie, fuite de gaz, etc.).
2. Confort de la vie quotidienne surtout pour les personnes âgées ou handicapées
3. Faciliter les économies d'énergie grâce à la réactivité maîtrisée d'une maison intelligente.

Figure II. 7 : Maison intelligente

b- Santé

Le secteur de la santé a connu un très grand nombre d'applications permettant à un patient et à son docteur de recevoir des informations, parfois même en temps réels, qu'il aurait été impossible de connaître avant l'apparition d'IoT.

Par exemple, (Porteuse Digital Health) qui est le premier médicament connecté sur le marché grâce à un capteur directement intégré dans l'être humain qui permet après ça le suivi des patients à distance.

Il existe plusieurs autres dispositifs disponibles, fixé autour du poignet et permettent également de suivre l'activité physique quotidienne du patient, mesurer le taux de sucre, compter le nombre de pas, les kms parcourus, le nombre de calories brûlées…, le dispositif lui envoie une alerte dans les cas anormaux [24].

Figure II. 8 : Santé intelligente

c- Transport

IoT dans les cas d'utilisation dans le domaine des transports connaît une croissance rapide et permet de réaliser des gains en termes d'efficacité opérationnelle, de réduction des coûts, de sécurité et de mobilité.

Transports publics Les services seront également améliorés et l'industrie automobile bénéficiera d'une série d'innovations rendues possibles par le site IoT, des bornes de recharge des véhicules électriques aux technologies des véhicules connectés [25].

Figure II. 9 : Iot appliqué au Transport

d- Industrie 4.0

L'Internet des objets a révolutionné le monde de l'industrie manufacturière. Le fait de connecter la chaîne de valeur permet une réduction majeure des inefficacités liées à la gestion des ressources, aux arrêts non planifiés et à la logistique non optimisée.

Au début, les premières implémentations des systèmes IoT au sein des manufactures offraient aux opérateurs une transparence et une accessibilité à l'état de la production. Le développement de ces technologies et des disciplines connexes telles que l'intelligence artificielle, a permis au rôle de l'IoT de migrer d'un simple outil de suivi à un outil de gestion automatisée où les programmes informatiques dictent la cadence des machines et suggèrent de nouvelles planifications de production [26].

Figure II. 10 : Industrie 4.0

e- Elevage

Le monde de l'élevage est pleine évolution. Dans un contexte d'élevage de précision, il est crucial d'avoir des données fraîches sur l'état sanitaire des individus. Le développement du numérique permet de proposer des outils performants pour optimiser par exemple le suivi personnalisé des animaux, la détection précoce des troubles de santé et du comportement. Le gestionnaire de l'élevage peut aussi disposer en temps réel des données les plus fraîches sur ses bâtiments et les individus présents. La promesse de l'internet des objets prend alors tout son sens et plusieurs expérimentations ont déjà eu lieux autour d'élevage de poulets de porcs et de bovins et d'oies [27].

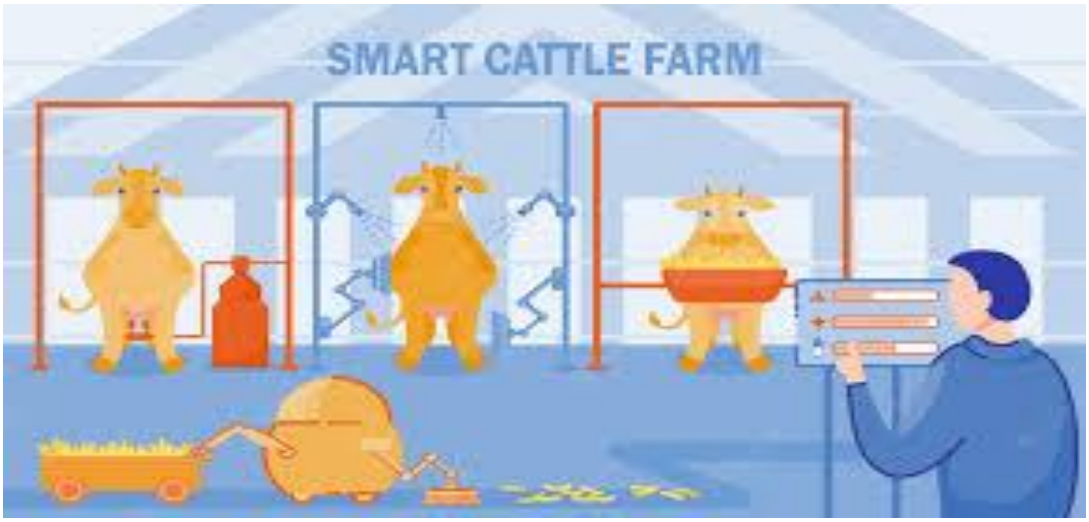


Figure II. 11 : Elevage intelligente

f- Smart City

La ville intelligente est un des cas phares du déploiement des technologies de l'Internet des objets en milieu urbain. Équiper les villes de capteurs et d'équipements autonomes leur permet de gagner en efficacité. Dans un contexte de ville intelligente, plusieurs capteurs sont déployés à différents endroits pour relever un large spectre de données démographiques (liées à l'achalandage, à l'empreinte de carbone, au niveau sonore, etc.). Cette collecte permet de constituer un panorama clair de la dynamique de la ville et de comprendre l'évolution de ses besoins. Des données factuelles et à jour permettent de gérer davantage les flux d'achalandage, de prévenir les dégagements excessifs de CO_2, de mieux organiser les ressources disponibles (p. ex. la gestion intelligente des déchets) et de mieux déployer les ressources humaines à la disposition de la ville [28].

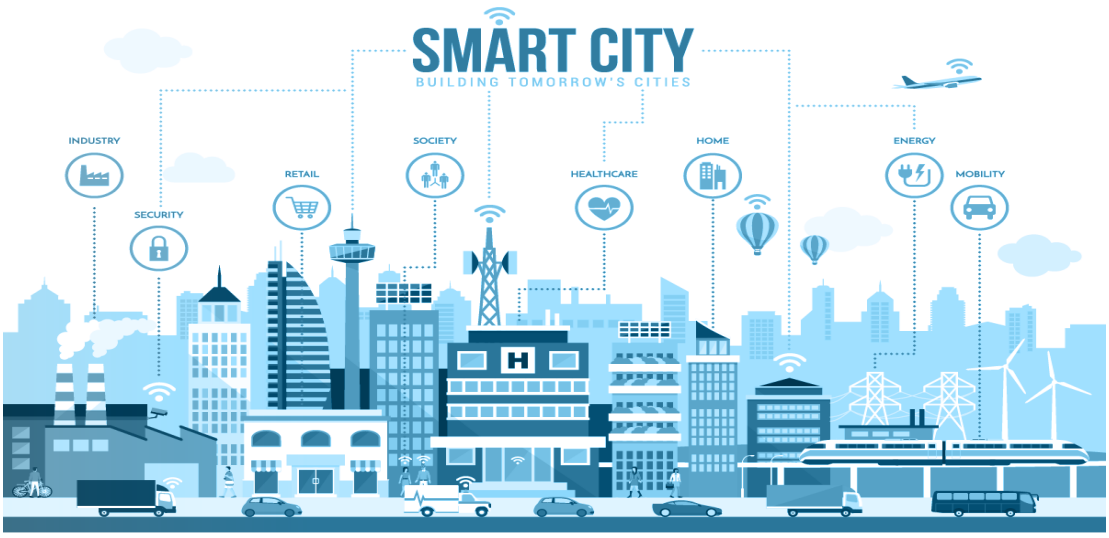


Figure II. 12 : Smart City (ville connectée)

g- Agriculture

Dans le monde agricole, les solutions IoT prennent la forme de capteurs reliés à Internet pour collecter des mesures environnementales et mécaniques. Leur déploiement permet aux agriculteurs de prendre des décisions éclairées et améliore presque tous les aspects de leur travail, de l'élevage à l'agriculture. L'agriculture intelligente basée sur les technologies IoT offre par ailleurs l'opportunité aux agriculteurs et aux producteurs de réduire les déchets et d'améliorer la productivité. Cela va de la quantité d'engrais utilisée au nombre de trajets effectués par les véhicules agricoles, en passant par une utilisation plus efficace des ressources

telles que l'eau et l'électricité. Grâce aux nouveaux outils IoT, les agriculteurs peuvent surveiller les conditions de leurs champs depuis n'importe où. Ils peuvent également choisir entre des options manuelles et automatisées pour prendre les mesures nécessaires en fonction des données collectées (lumière, humidité, température, santé des cultures, etc.) [29].

Figure II. 13 : Agriculture intelligente

II-6 Technologies de communication d'IoT

Pour la communication des Iot plusieurs technologies sont disponibles, ces technologies sont classées par catégories, par rapport à un ensemble de caractéristiques communes, tel que le débit, la portée et la bande de fréquences dans laquelle ils opèrent [30].

a- Réseaux étendus sans fil (WWAN)

Souvent considéré comme les réseaux les plus étendus dans la communication de la technologie. Ils sont également connu sous le nom de réseau cellulaire mobile (GSM, UMTS, et LTE.), représentent généralement les réseaux à liaisons sans fil à faible consommation l'énergétique (LoRaWAN et Sigfox) [31].

a-1 LoRaWAN

LoRaWAN est un protocole de réseau étendu à faible consommation basé sur la technique de modulation radio LoRa. Il connecte sans fil les appareils à Internet et gère la communication entre les appareils terminaux et les passerelles réseau.

L'utilisation de LoRaWAN dans les espaces industriels et les villes intelligentes se développe car il s'agit d'un protocole de communication bidirectionnel à longue portée abordable avec une très faible consommation d'énergie - les appareils peuvent fonctionner pendant dix ans avec une petite batterie. Il utilise les bandes radio ISM (Industriel, Scientifique, Médical) sans licence pour les déploiements de réseau [32].

Elle possède une architecture totalement adaptée à l'IoT, lui permettant de localiser facilement les objets mobiles. Elle est déployée pour des réseaux nationaux par des grands opérateurs de télécommunications (ex. Orange). Les réseaux LoRaWAN sont généralement présentes par une topologie en étoile dans laquelle des passerelles relient des terminaux (ex capteurs, ordinateurs, etc.) a un serveur réseau central, qui est relie à son tour à un serveur d'applications.

Architecture :

L'architecture LoRaWAN est composée de nœuds d'extrémité, de passerelles, d'un serveur de réseau et d'un serveur d'applications comme présenté dans la figure ci-dessous [33].

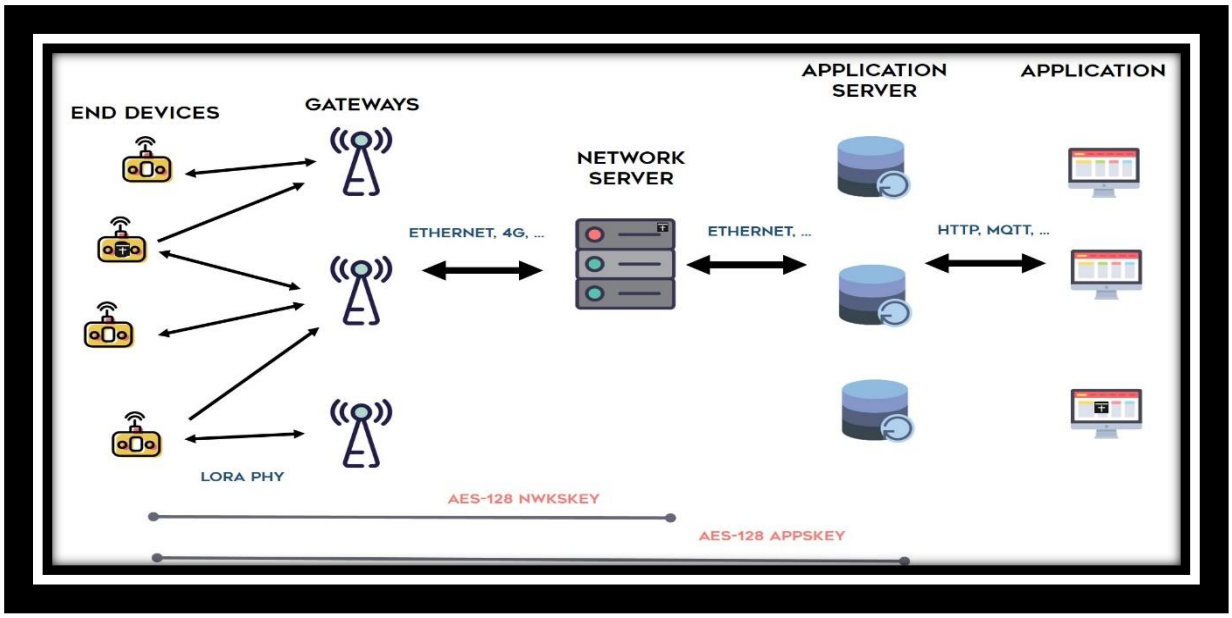


Figure II. 14 : Architecture des Réseaux LoRaWAN

La topologie d'un réseau LoRaWan est en étoile : les équipements appelés end-devices communiquent en LoRa avec des concentrateurs appelés gateways. Ces concentrateurs centralisent les messages pour les transmettre au serveur de gestion du réseau. La liaison entre les concentrateurs et le serveur de gestion du réseau s'appuie sur des technologies très haut débit (Ethernet, 4G).

Toute l'intelligence, à savoir la gestion du débit adaptatif, de la sécurité des données ou encore de la redondance des données reçues, est assurée par le serveur du réseau. Enfin, ce dernier communique avec un ou plusieurs serveurs applicatifs au travers desquels les fournisseurs d'applications exploitent les données de leur(s) équipement(s).

Car en effet, une des particularités d'un réseau LoRaWAN, est qu'un équipement ne communique pas exclusivement à travers un concentrateur. Tous les concentrateurs couvrant l'équipement peuvent recevoir les données transmises par ce dernier.

Cela facilite grandement la communication avec les équipements en mobilité en dispensant le réseau de mécanismes de hand-over (passage d'un concentrateur à un autre) qui auraient pour effet de complexifier sa gestion et très probablement de réduire ses performances. Par contre, lorsque le serveur envoie un message à destination d'un équipement, c'est par le biais d'un seul concentrateur. C'est le cas en particulier des messages requérant un acquittement par le serveur [34].

SECURITE

La politique de sécurité de LoRaWAN assure les mécanismes de base qui sont l'authentification des objets, la confidentialité et l'intégrité des données. Cette politique définie également des techniques de partage de clés [35].

CONFIDENTIALITE

Une fois que l'objet soit associe au réseau de LoRaWAN, tous les messages échangent doivent être chiffres pour assurer la confidentialité en utilisant les clés de session connus uniquement par le serveur réseau et l'objet concerne. Le chiffrement de message est établi via le standard AES128.

a-2 Technologies Cellulaire

Ce sont des réseaux longue portée (de quelques kilomètres en ville `a 30 km en zone rurale) et consommateurs d'énergie. A l'image des réseaux GSM, 2G, 3G ou 4G, ils permettent le transport de grands volumes de données (vidéos, images, etc.) et ont une bonne couverture au niveau national et international [36].

a-2-1 technologies 2G

La deuxième génération (2G) de systèmes cellulaires repose sur une technologie numérique a été développée à la fin des années 1980. Ces systèmes cellulaires utilisent une technologie

numérique pour la liaison ainsi que pour le signal vocal. Ce système apporte une meilleure qualité ainsi qu'une plus grande capacité à moindre coût pour l'utilisateur. La deuxième génération de systèmes cellulaires (2G) utilise essentiellement les standards suivants :GSM (2G), CDMA**,** TDMA. [37].

a-2-2 Les technologies 3G

Le système 3G utilise le CDMA (Code Division Multiple Access) et WCDMA (Wide Band Code Division Accès multiple). Le CDMA est une technique dans laquelle un code unique est attribué à chaque utilisateur utilisant son code en même temps. Après avoir attribué un code unique, la largeur de bande entièrement disponible est utilisée efficacement en elle. De ce fait, un très grand nombre d'utilisateurs peuvent utiliser la chaine en même temps par rapport `a la TDMA et FDMA [38].

a-2-3 Les technologies 4G

Le LTE (Long Term Evolution), aussi appelé 4G (quatrième génération) : cette technologie, la plus moderne actuellement commercialisée, offre des débits jusqu'à 6 fois supérieurs à la H+ dans sa déclinaison la plus moderne (4G+). Elle utilise les fréquences 2600 MHz, 1800 MHz, 700 MHz et 800 MHz. Cette dernière était auparavant employée par la télévision analogique, aujourd'hui disparue au profit de la TNT [39].

a-2-4 Les technologies 5G

La 5G, qui est développée depuis quelques années et dont les premiers forfaits ont été lancés fin 2020. Cette technologie représente les plus gros investissements à venir des opérateurs télécom. La principale bande de fréquence de la 5G est la bande 3.5 GHz, mais les bandes 700 Mhz et 2100 Mhz peuvent être également utilisées. Les débits de la 5G peuvent, en théorie, dépasser les 1Gbit/s [40].

b- Réseaux métropolitains sans fil (WMAN)

Le réseau métropolitain sans fil (WMAN pour Wireless Metropolitan Area Network) est connu sous le nom de Boucle Locale Radio (BLR). Les WMAN sont basés sur la norme IEEE 802.16. La boucle locale radio offre un débit utile de 1 à 10 Mbit/s pour une portée de 4 à 10 kilomètres, ce qui destine principalement cette technologie aux opérateurs de télécommunication. La norme de réseau métropolitain sans fil la plus connue est le WiMAX, permettant d'obtenir des débits de l'ordre de 70 Mbit/s sur un rayon de plusieurs kilomètres [41].

b-1 WIMAX

Le WiMAX ou World wide Interoperability for Microwave Access est une famille de normes, définissant des connexions à haut-débit par voie hertzienne, développée par le Consortium WiMAX Forum et ratifié en 2001 par l'IEEE sous le nom IEEE-802.16. Le WiMAX est aussi le nom commercial délivré par le WiMAX Forum aux équipements conformes à la norme IEEE 802.16, afin de garantir un haut niveau d'interopérabilité entre ces différents équipements [42].

c- Réseaux locaux sans fil (WLAN)

Le réseau local sans fil (note WLAN pour Wireless Local Area Network) est un réseau permettant de couvrir l'équivalent d'un réseau local d'entreprise, soit une portée d'environ une centaine de mètres. Il permet de relier entre eux les terminaux présents dans la zone de couverture. Il existe plusieurs technologies concurrentes exemple : wifi ; hepperlan [43].

d- Réseaux personnels sans fil (WPAN)

Concernent les réseaux sans fil à faible portée, de l'ordre de quelques dizaines de mètres. Tout comme la portée qui varie d'une technologie WPAN `a une autre, le débit varie aussi. Ce dernier peut être à 250 Kbits/s (ZigBee) jusqu'à 1Mbits/S (cas du Bluetooth). Ces technologies suivent la famille IEEE 802.15, les plus connues celles de la famille sous norme IEEE 802.15.1(Bluetooth), et celles qui sont utilisées dans le domaine des réseaux de capteurs sans fil (WSN pour Wireless Sensor Networks) qui suivent principalement la sous norme IEEE 802.15.4 tel que ZigBee, OCARI, 6LoWPAN, etc [44].

d-1 ZIGBee

Le Zigbee est une technologie WPAN à faible débit et à faible consommation de ressources (Énergie, calcul, et mémorisation) qui peut être déployé avec une topologie en mode étoile ou maillée [45]. La bande de fréquences 2,4 GHz, les débits de données peuvent atteindre 250 Kb/s, tandis que dans la bande de fréquences 868 MHz, il n'a que20 Kb/s [46].

Conclusion

L'Internet des objets (IoT) est un concept qui consiste à connecter des objets de la vie quotidienne à Internet afin de collecter et partager des données. Cette technologie a le potentiel de transformer de nombreux aspects de notre vie, allant de l'automatisation de la maison au suivi de la santé en passant par la gestion efficace des ressources. Cependant, l'IoT pose également des défis en termes de sécurité et de confidentialité des données, ainsi que des questions éthiques sur l'utilisation de ces données par les entreprises et les gouvernements. Il

est important de trouver un équilibre entre les avantages potentiels de l'IoT et les préoccupations légitimes des individus et de la société pour garantir un développement responsable de cette technologie. Ce qui conduit à l'étude des différents aspects relatifs au Fog Computing afin de voir son application dans l'IoT.

CHAPITRE III : Paradigme du Fog Computing

III- 1 Introduction

Pour faire face à l'évolution des applications IoT, le Fog computing est apparu comme une alternative pour éviter la charge des centres de données et du réseau dans les infrastructures Cloud. En étendant le Cloud vers la périphérie du réseau, le Fog est capable de prendre en charge les applications IoT géographiquement distribuées, sensibles à la latence ou exigeantes en bande passante.

Plusieurs recherches ont défini l'informatique en brouillard de manière similaire mais complémentaire, par exemple comme :

-L'informatique en brouillard est un scénario dans lequel un très grand nombre de systèmes hétérogènes (sans fil et parfois autonomes) omniprésents et décentralisés communiquent et potentiellement coopèrent entre eux et avec le réseau pour effectuer des tâches de stockage et de traitement sans l'intervention de tierces parties.

Ces tâches peuvent être destinées à soutenir fonctions de base du réseau ou de nouveaux services et applications qui s'exécutent dans un environnement "sandbox". Les utilisateurs qui louent une partie de leurs appareils pour héberger ces services reçoivent des incitations pour le faire." [47].

- Le Fog Computing est une plate-forme hautement virtualisée qui fournit des services de calcul, de stockage et de mise en réseau entre les dispositifs finaux et les données traditionnelles du Cloud Computing [14].

- L'informatique en brouillard est un modèle en couches pour permettre un accès omniprésent à un continuum partagé de ressources informatiques évolutives. Le modèle permet le déploiement d'applications et de services distribués, sensibles à la latence, et se compose de nœuds de brouillard (physiques ou virtuels), résidant entre les appareils finaux intelligents et les services centralisés (nuage) [48].

L'informatique en brouillard est principalement une plateforme virtualisée qui fournit des services de calcul, de stockage et de réseau à n'importe quel endroit du continent, du nuage aux utilisateurs finaux.

III- 2 Architecture et éléments d'une infrastructure Fog Computing

Le Fog présente une infrastructure virtuelle et géo-distribuée intégrant un grand nombre d'équipements hétérogènes et qui sont interconnectés [61]. Comme illustré dans la Figure III.1, l'infrastructure du Fog comprend trois couches principales, la couche terminale, la couche de brouillard (Fog) et la couche de Cloud. Ci-après, nous décrivons chaque couche de l'infrastructure illustrée dans la Figure III.1.

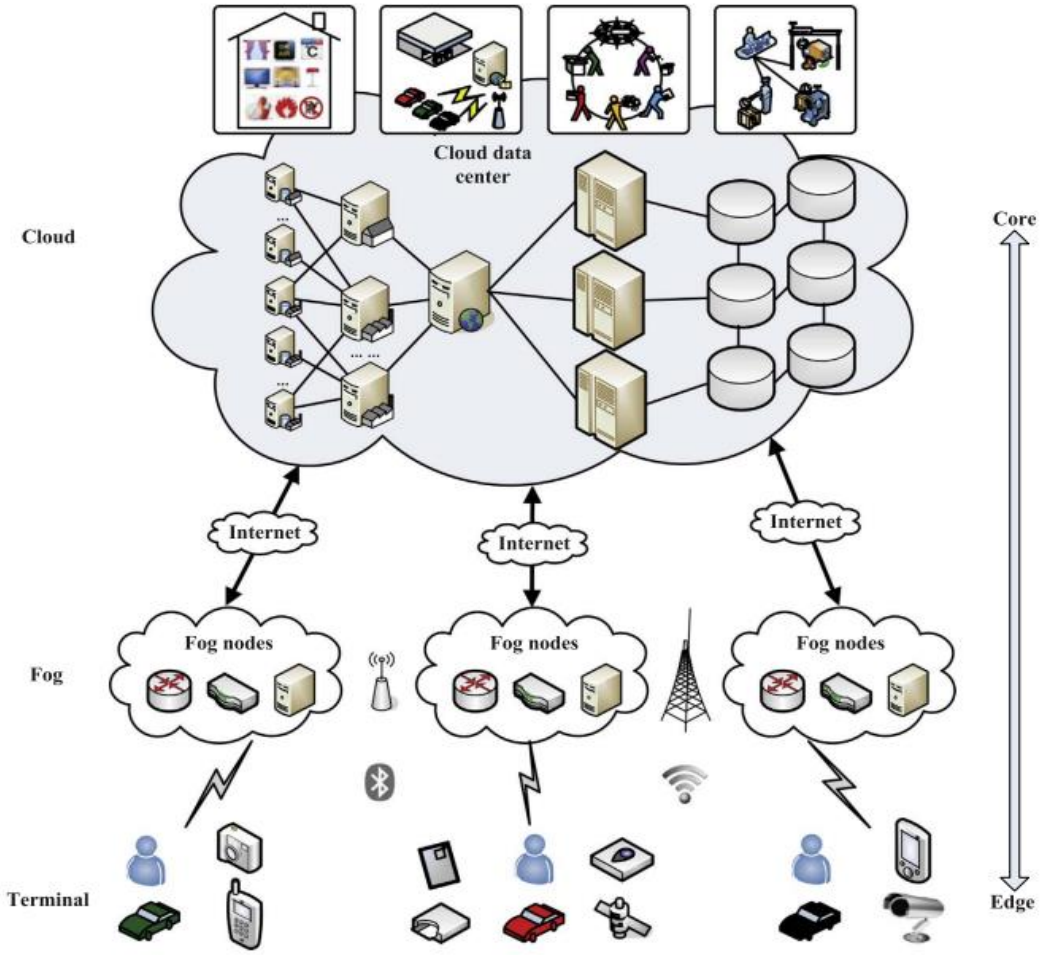


Figure III. 1 : Architecture d'une infrastructure Fog Computing

Couche terminale : Il s'agit de la couche la plus proche de l'utilisateur final et de l'environnement physique. Elle est constituée de divers dispositifs IoT, par exemple, capteurs, téléphones mobiles, véhicules intelligents, cartes à puce, lecteurs, etc. et ainsi de suite. En particulier, bien que les téléphones mobiles et les véhicules intelligents disposent d'une puissance de calcul, nous ne les utilisons ici que comme dispositifs de détection intelligents. Ces dispositifs sont largement distribués géographiquement en généralement. Ils sont chargés

de détecter les données caractéristiques d'objets physiques ou d'événements physiques et de transmettre ces données détectées à la couche supérieure pour traitement et stockage.

Couche de brouillard : Cette couche est située à la périphérie du réseau. La couche de calcul du brouillard Computing est composée d'un grand nombre de nœuds de brouillard, qui comprennent généralement des routeurs, des passerelles, des commutateurs, des points d'accès, des points de base, des stations de base stations de base, des serveurs de brouillard spécifiques, etc. Ces nœuds de brouillard sont largement distribués entre les dispositifs finaux et le nuage, par exemple, les cafés, centres commerciaux, terminaux de bus, rues, parcs, etc. Ils peuvent être statiques à un endroit fixe, ou mobiles sur un support en mouvement. Les dispositifs d'extrémité peuvent se connecter facilement aux nœuds de brouillard pour obtenir des services.

Ils ont la capacité de calculer, de transmettre et de stocker temporairement les données détectées reçues. L'analyse en temps réel et les applications sensibles à la latence peuvent être réalisées dans la couche de brouillard. En outre, les nœuds de brouillard sont également connectés au centre de données en nuage par un réseau central IP et sont responsables de l'interaction et de la coopération avec le nuage pour obtenir des capacités de calcul et de stockage plus puissantes.

Couche de cloud Computing : La couche d'informatique en nuage se compose de plusieurs serveurs et dispositifs de stockage à haute performance, et fournit divers services d'application, tels que la maison intelligente, le transport intelligent, l'usine intelligente, etc. Elle dispose de puissantes capacités de calcul et de stockage pour prendre en charge l'analyse de calculs extensifs et le stockage permanent d'une énorme quantité de données. Il dispose de puissantes capacités de calcul et de stockage pour prendre en charge l'analyse de calculs étendus et le stockage permanent d'une énorme quantité de données. Cependant, à la différence de l'architecture traditionnelle de l'informatique en nuage, toutes les tâches de calcul et de stockage ne passent pas par le nuage. En fonction de la charge de la demande, les modules centraux du nuage sont gérés et programmés de manière efficace par certaines stratégies de contrôle permettent d'améliorer l'utilisation des ressources du nuage.

III- 3 Caractéristiques du Fog Computing

Le National Institute of Standards and Technology (NIST) et l'OpenFog consortium ont identifié un certain nombre de caractéristiques définissant les infrastructures Fog et que nous résumons dans la liste non exhaustive suivante :

a- Distribution géographique

L'infrastructure Fog peut s'étendre sur une très large zone géographique à l'échelle d'une ville ou d'une région. Cette infrastructure va héberger des applications dont les composants peuvent être distribués dans n'importe quelle région.

b- Hétérogénéité et hiérarchisation des nœuds

L'architecture Fog la plus répandue dans la littérature et qui est aussi la plus générique (également utilisée par le NIST) définit l'architecture Fog comme une architecture en couches 3 tiers. Ce modèle représente une hiérarchie entre les nœuds de calcul, allant des équipements utilisateurs jusqu'au centre de données en passant par des équipements Edge et Fog Computing. Le principe est que les capacités de calcul, de stockage et la consommation énergétique des nœuds augmentent en allant de la première couche de nœuds utilisateurs à la dernière représentant les centres de données.

c- Sensibilité à la mobilité et à la géolocalisation des utilisateurs

Le Fog est un environnement hétérogène et dynamique. Ceci est principalement dû à la mobilité des utilisateurs et des équipements à la périphérie du réseau. Il est donc important d'avoir les positions géographiques des utilisateurs et de connaître la portée des applications qu'ils utilisent.

d- Sécurité

Le Fog Computing peut pallier aux failles de sécurités des réseaux de capteurs en offrant des nœuds intermédiaires avec des protocoles de communication plus sécurisés que ceux des objets connectés, que nous verrons dans la suite du chapitre. Le Fog permet également de gérer les services et applications au plus proche des utilisateurs. Les opérations de stockage de données, de filtrage et traitement ayant lieu au plus proche des utilisateurs évitent de faire transiter des informations sensibles vers les serveurs Cloud.

e- Interopérabilité et fédération

Une infrastructure Fog étendue regroupe plusieurs domaines réseau et de calcul gérés par différents organismes. Ces domaines doivent coopérer pour assurer un déploiement fluide et une gestion transparente des services pour les utilisateurs.

f- Autonomie et programmable

L'une des principales caractéristiques des infrastructures Fog est l'autonomie de décision pour la gestion du cycle de vie des applications qui y sont déployées. La facilité de programmable et de reconfiguration des équipements grâce à la virtualisation simplifie la gestion et assure l'évolutivité de l'infrastructure face à la dynamicité de l'environnement.

III- 4 Concepts connexes au Fog : Edge et Mist computing

a- Cloud Computing

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 [62].

Figure III. 2: Cloud Computing

b- Edge Computing

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 [63].

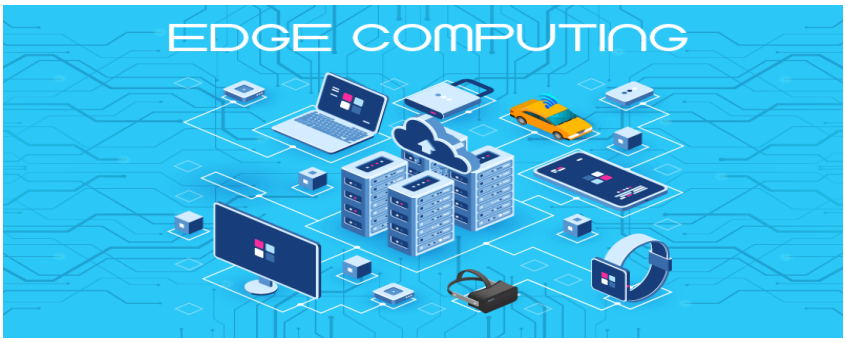


Figure III. 3: Edge Computing

c- Mist Computing

Les nœuds mist se trouvent à la périphérie, dans le brouillard et dans le nuage, et fonctionnent tous de concert comme un seul ordinateur maillé décentralisé et distribué, c'est-à-dire le mist. Le mist computing, sans interface humaine, gère l'acheminement des calculs vers le bon nœud, au bon moment, au bon endroit, généralement aussi près que possible de la source des données. Pour cette raison, le mist computing est parfois appelé Everything Computing Everywhere, ce qui est proche de l'informatique ubiquitaire [64].

d- Fog Computing

Le Fog computing, aussi appelé "informatique dans le brouillard", définit une infrastructure chargée de stocker et de traiter des données issues d'objets connectés. Concurrent direct, alternative ou solution complémentaire au cloud computing, le Fog computing a comme particularité de stocker et de traiter les données via le recours à des équipements implantés à la périphérie du réseau. Il permet donc de réaliser ces deux actions en local, sans avoir à solliciter un Datacenter situé à plusieurs centaines de kilomètres ou un cloud. Dans ce domaine du stockage et du traitement des données et de l'IoT, le Fog computing crée une interface supplémentaire que l'on peut situer entre le Edge Computing et le Cloud Computing.

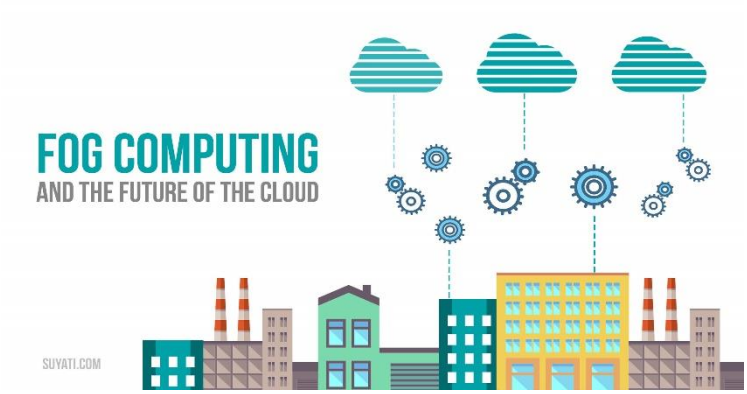


Figure III. 4 : Fog Computing le future du Cloud

III- 5 Les Avantages et Les Défis

a- Avantages

Plusieurs travaux de recherche ont listé les bénéfices apportés par le Fog computing par rapport au paradigme du Cloud computing [52, 58]. Ci-après nous en citons quelques-uns.
• Réduction des latences *:* en raison de sa proximité avec les utilisateurs finaux, le Fog a la capacité de supporter des applications qui nécessitent des latences courtes et stables.
• Réduction du trafic réseau : le Fog computing permet d'exécuter des fonctions de traitement, de filtrage et d'agrégation de données tout au long du chemin du réseau. En effet, la pertinence de l'envoi de données est examinée à chaque étape de la transmission permettant une réduction importante du trafic réseau.

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

• Passage à l'échelle : la géodistribution des nœuds de Fog et leur proximité avec les utilisateurs finaux permettent de gérer un grand nombre d'objets connectés (la répartition de la charge de travail). Les nœuds de Fog sont de nature hétérogène avec des performances et des coûts différents. Ce deuxième point permets de déployer, selon le besoin, de nouveaux nœuds de manière simple et moins couteuse (en comparaison avec le Cloud).

• Interopérabilité et fédération : les infrastructures de Fog comprennent un grand nombre de nœuds géo-distribués. Plusieurs de ces nœuds peuvent fédérer/coopérer dans un cluster pour réaliser des taches complexes telle que l'analyse de données massives.

• Décharge de traitement (offloading) : en raison de sa proximité, le Fog peut aider les objets ayant des ressources limitées à décharger une partie de leurs traitements aux nœuds de calcul localisés en périphérie du réseau. Cela permet aux objets, par exemple, de préserver leur énergie, de réduire le temps de traitement ou d'augmenter la capacité de stockage de données.
• La disponibilité : la géo-distribution des infrastructures de Fog aide à lancer une tache ou à stocker plusieurs copies d'une donnée sur des nœuds de Fog différents. Ceci permet d'augmenter la disponibilité du service et la résilience contre la perte de données.

b- Défis

Bien que le paradigme de Fog computing ait amené beaucoup d'avantages pour la gestion du traitement et du stockage de données, ce paradigme reste nouveau et nécessite d'investiguer les défis suivants.

1. Hétérogénéité : en plus de l'hétérogénéité trouvée dans l'IoT au regard des différents types d'objets, de données, de technologies de communication et de services, les infrastructures de Fog comprennent des nœuds de nature et de performances différentes. La gestion de l'hétérogénéité dans un environnement de Fog et d'IoT représente un défi majeur aujourd'hui [54].
2. Sécurité : les nœuds du Fog peuvent être déployés à l'extérieur et laissés sans surveillance (i.e. dans les rues, au-dessus des bâtiments, etc.). Ils sont donc vulnérables à des attaques telles que le détournement de données et l'écoute indiscrète [50]. Afin de préserver la sécurité du système, le déploiement des solutions pour le contrôle d'accès, l'authentification et la détection d'intrusion sont nécessaires dans chaque niveau de l'infrastructure [58, 63].

3. Gestion et provision de ressources *:* les nœuds de Fog sont en général des équipements réseaux ayant une puissance de calcul et une capacité de stockage limitées. Des solutions efficaces sont nécessaires pour gérer l'ordonnancement des taches dans ces infrastructures (ex. des solutions basées sur la priorité et la migration) [58]. De plus, afin de fournir un support de mobilité notamment dans le cas des objets connectés, les ressources doivent être pré-allouées, par exemple suivant des méthodes probabilistes basées sur l'historique des utilisateurs [65].
4. Gestion des données *:* le suivi et la gestion des données dans les différents niveaux de

l'infrastructure du Fog (i.e. périphérie, agrégation et cœur) est un défi majeur. La découverte, la réplication, le placement et la persistance des données nécessitent un examen attentif dans ce contexte [66].

5. Gestion de l'énergie : comme mentionné ci-dessus, les infrastructures de Fog comprennent un grand nombre de nœuds répartis géographiquement. La consommation énergétique doit être donc plus élevée en comparaison avec celle du Cloud [65, 50]. De grands efforts de recherches sont nécessaires pour développer des solutions efficaces pour la gestion d'énergie, par exemple, des processus de traitement de données et des protocoles de communications moins coûteux en termes de consommation énergétique sont à développer [60].

6. Modèle de programmation : dans le Cloud, les infrastructures sont transparentes pour les utilisateurs. Les traitements sont réalisés dans des serveurs virtualisés dans les centres de données. En revanche, dans le Fog, les traitements sont faits à différents niveaux de l'infrastructure (i.e. périphéries, agrégation et cœur de réseau) qui sont des plates-formes dynamiques et hétérogènes. Afin de faciliter le développement des applications sur les plates-formes de Fog computing, il est nécessaire de fournir un modèle unifié qui prend en compte l'aspect dynamique, hiérarchique et hétérogène des ressources du Fog [50, 63].

7. Qualité de service (QoS) : dans [63], les auteurs ont étudié la QoS dans le Fog computing selon quatre aspects : 1) la connectivité, 2) la fiabilité, 3) la capacité et 4) la latence. Ci-après nous décrivons chacun de ces aspects en précisant les défis associés.

a) La connectivité : le Fog étend les services du Cloud jusqu'aux périphéries du réseau. Dans un tel réseau hétérogène, le partitionnement, le regroupement et la collaboration fournissent une opportunité pour optimiser le coût, augmenter le débit, ou réduire la consommation énergétique [54, 63]. Le défi présenté ici consiste à concevoir des algorithmes de communication efficaces pour optimiser les métriques cités ci-dessus [54].

b) La fiabilité : comme le Fog computing est réalisé par l'intégration d'un grand nombre d'équipements répartis géographiquement, la fiabilité est l'un des principaux défis à considérer lors de la conception d'un tel système [65]. La fiabilité peut être améliorée grâce à une vérification périodique pour reprendre après un échec, à un ré-ordonnancement des tâches échouées ou à une réplication pour exploiter le traitement en parallèle. Toutefois, les points de contrôle et d'ordonnancement ne peuvent pas s'adapter à l'environnement hautement dynamique du Fog à cause des latences. La réplication semble plus prometteuse, mais elle repose sur le fonctionnement synchronisé de plusieurs nœuds de Fog [63].

c) La latence : le Fog computing a été proposé principalement pour lutter contre les latences élevées du Cloud. Le Fog est utilisé pour déployer des applications sensibles aux latences comme l'Internet des véhicules, la santé et l'industrie 4.0. Afin de réduire la latence, il est important d'étudier comment les données et leurs traitements sont placés dans l'infrastructure. Par exemple, un nœud de Fog peut avoir besoin de traiter des données distribuées dans plusieurs nœuds éloignés. Le calcul ne peut être démarré qu'après avoir récupéré toutes les données requises, ce qui ajoute de la latence au service [63].

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

e) Complexité : les algorithmes d'optimisation existants sont en général ciblés sur le temps de traitement et l'utilisation de ressources. Vu que le Fog fournit une plateforme avec des milliers de nœuds pour servir des milliards d'objets connectés, il est nécessaire de concevoir des solutions de gestion décentralisées et coopératives. Par exemple, afin d'accélérer le temps de placement de données ou de traitements, il est nécessaire d'utiliser des algorithmes parallèles ou des méthodes approchées (ex. basées sur le concept de diviser pour régner), plutôt que d'utiliser des méthodes exactes et centralisées [54].

III- 6 Outils d'Evaluation

a- Plateforme réelle

Une plate-forme d'analyse en temps réel permet aux organisations de tirer le meilleur parti des données en temps réel en les aidant à en extraire les précieuses informations et tendances. Ces plates-formes aident à mesurer les données du point de vue commercial en temps réel, ce qui optimise encore l'utilisation des données.

a-1 FIT-IoT-LAB

Dans [67] ont proposé FIT-IoT-LAB, une plateforme expérimentale fédérée à grande échelle pour la conception et la comparaison des protocoles, applications et services d'IoT. Elle comprend 2700 capteurs physiques sans fil dont 117 nœuds de robots mobiles déployés sur six sites en France. Cette plateforme n'intègre ni nœuds de Fog ni stratégies de placement de

données. FIT-IoT-LAB est une plate-forme d'expérimentation à large échelle dans le domaine de l'Internet des Objets. De manière complètement automatisée, un chercheur réserve à distance via Internet plusieurs centaines d'objets connectés pour un temps donné. L'infrastructure de la plate-forme IoT-LAB se charge de déployer de manière rapide et transparente l'application étudiée sur la grille d'objets préalablement réservée. Tout au long de l'expérimentation, les outils de plate-forme assurent un monitoring afin d'évaluer les performances de l'application étudiée, notamment en terme de consommation énergétique et métriques réseaux.

De par le nombre d'objets connectés et de sites disponibles, IoT-LAB autorise différentes topologies de déploiement possibles. Actuellement, 1500 objets connectés sont disponibles, dont 250 pour le site de Strasbourg/ICube. Une fonctionnalité particulière du site de ICube est la gestion de la mobilité pour certains objets connectés embarqués sur des robots mobiles [68].

a-2 YOUPI

YOUPI [69] est une plateforme Fog/Edge Computing pouvant communiquer avec d'autres infrastructures de calcul comme Grid500 et FIT IoT Lab. YOUPI permet de capturer des propriétés dynamiques de l'environnement comme la mobilité, les interférences, les comportements utilisateurs. Les expérimentations peuvent être rejouées pour effectuer des analyses hors lignes.

b- Emulateur

Concernant les outils basés sur l'émulation, Mayer et al ont proposé EmuFog [68], et Coutinho et al ont introduit Fogbed [43], deux émulateurs d'environnements de Fog. Les deux considèrent des infrastructures de Fog comprenant des switchs et des routeurs mais ils n'émulent pas d'objets connectés.

Cependant, Fogbed utilise la simulation pour manipuler les objets connectés. De plus, ces deux émulateurs n'intègrent pas la gestion du placement de données. En raison de la grande densité des infrastructures de Fog, les solutions basées sur l'émulation peuvent être très couteuses en termes de ressources et de temps d'émulation (plusieurs semaines, voire plusieurs mois).

b-1 Emufog

EmuFog [70] est un émulateur d'environnements Fog en libre accès sous licence MIT. Cet outil développé en JAVA, offre la possibilité de déployer des applications sous forme de containers Docker. Cet émulateur est construit à partir de MaxiNet qui est une version étendue de MiniNet pouvant être utilisée sur plusieurs machines physiques.
MiniNet permet de créer des réseaux définis par logiciels [70]. Il est possible d'utiliser des topologies réseaux générées à partir d'outils comme BRITE ou CAIDA. EmuFog permet de garder les historiques des consommations CPU et mémoire de chaque noeud. Il est possible de spécifier les emplacements des services IoT sur les machines physiques. L'outil n'a pas d'interface permettant de récupérer des métriques globales comme le temps de réponse d'une application ou l'énergie consommée par cette dernière.

b-2 Fogbed

Fogbed est un Framework qui étend l'émulateur Mininet pour créer des testbeds de brouillard dans des environnements virtualisés. En utilisant une approche de bureau, Fogbed permet le déploiement de nœuds de brouillard virtuels en tant que conteneurs Docker sous différentes configurations de réseau.
L'API de Fogbed fournit des fonctionnalités permettant d'ajouter, de connecter et de supprimer dynamiquement des conteneurs de la topologie du réseau. Ces fonctionnalités permettent l'émulation d'infrastructures cloud et Fog réelles dans lesquelles il est possible de démarrer et d'arrêter des instances de calcul à tout moment. Il est également possible de modifier, au moment de l'exécution, les limites des ressources d'un conteneur, telles que le temps d'unité centrale et la mémoire disponible.
Un environnement d'émulation Fogbed peut être créé en déployant des nœuds virtuels, des commutateurs virtuels, des connexions virtuelles et des instances virtuelles dans un environnement de réseau virtuel fonctionnant sur une machine hôte. La configuration flexible est obtenue en utilisant des images de conteneurs Docker préconfigurées. Chaque image de conteneur comprend une partie d'une application distribuée, les services et les protocoles requis. Différents types d'images de conteneur peuvent être utilisés pour instancier des nœuds virtuels [71].

c- Simulateur

Les logiciels de simulations informatiques sont essentiellement des programmes qui permettent aux utilisateurs d'observer une opération par le biais d'une simulation sans effectuer cette opération. Ils sont souvent utilisés pour faire en sorte que le produit final soit aussi proche que possible des spécifications de conception afin d'éviter tout coût supplémentaire. Ce type de logiciel peut être utilisé pour la recherche, le testing ou la formation. Il fournit des données critiques sur presque tout type de projet avant la phase de prototypage.

✓ 1 GridSim

GridSim [72] est un simulateur d'environnements pair-à-pair et de Grilles de calcul. Il permet de modéliser et de simuler des ressources de grille et des réseaux, et d'évaluer des solutions d'allocation et d'ordonnancement des tâches, et enfin de gérer le placement de données dans l'infrastructure simulée. Cependant, il n'y a pas de support dédié aux environnements de *Fog* et d'IoT dans GridSim. Ainsi, les utilisateurs doivent définir des modèles de nœuds de *Fog* et des objets d'IoT afin de mettre en œuvre leurs stratégies de placement de données dans ce contexte.

✓ 2 Fog-torch

FogTorch [73] est un outil libre d'accès sous licence MIT et développé en Java. L'outil permet de tester des stratégies de déploiement d'applications en établissant des critères de QoS. Le simulateur utilise des simulations de Monte Carlo pour implémenter les variations de la bande passante des liens réseau.

Le simulateur a deux métriques de sortie : La garantie de QoS assurance et La consommation des ressources Fog (pourcentage de mémoire consommée). Le simulateur permet de modéliser différentes topologies réseau mais ne supporte pas la mobilité des nœuds.

✓ 3 Yet Another Fog Simulateur

YAFS [74] est un simulateur pour les réseaux Cloud/Fog. Le principal objectif de YAFS est l'évaluation des performances des stratégies de placement, de programmation et de routage. Parmi les mesures rapportées par YAFS figurent l'utilisation réseau, le temps de réponse et le temps d'attente.

Les données brutes résultantes sont rapportées dans un journal et permettent aux utilisateur de calculer d'autres mesures de qualité de service. YAFS est un logiciel libre d'accès sous licence MIT et est développé en Python.

✓ 4 FogNetSim++

FogNetSim++ [75] est un simulateur en libre accès, qui se concentre sur les aspects réseau. Il a été développé à partir du simulateur à évènements discrets OMNeT++ (en C++). Le simulateur offre la possibilité de faire du hand-over et propose différents protocoles de communication IoT comme MQTT ou CoAP. Il propose également quelques modelés de mobilité.

✓ 5 Edge CloudSim

Edge CloudSim [76] est une extension de CloudSim en libre accès sous licence GNU General et développé en java. Ce simulateur se focalise sur divers concepts du Edge computing et permet de modéliser des aspects réseau (propriété des liens et capacités réseaux), des aspects de calculs (exécution de taches, ordonnancement de machines virtuelles) et des aspects Fog (mobilité, offloading).

Il est possible de déployer des architectures multicouches avec plusieurs serveurs Edge/Fog gérés par un nœud Cloud. Les métriques de sortie sont les délais dans les réseaux locaux (Local Area Network -LAN) et dans les réseaux étendus (Wide Area Network -WAN), le taux d'échecs d'exécution des services et le taux moyen d'échecs d'exécution des services dus à la mobilité, le taux d'utilisation des VMs et le temps d'exécution des services.

Il est possible de définir des modèles de mobilité avec le module "mobile client layer". Le simulateur ne propose pas de modèles de consommation énergétique mais l'utilisateur peut définir ses propres modèles de coûts. La migration de services n'est pas supportée et les méthodes pour créer des clustersde calcul n'est possible qu'avec les noeuds appartenant à la même couche.

✓ 6 IfogSim

IFogSim [12] est un simulateur en libre accès développé en JAVA à partir de CloudSim. Le simulateur permet de modéliser et de simuler un environnement Fog Computing incluant des objets IoT, il permet d'évaluer la gestion des ressources de calcul et de tester des politiques de placement et d'ordonnancement de services dans le Fog.

Les métriques de sortie sont la consommation énergétique du calcul, l'utilisation réseau et les

temps d'exécution des services. IFogSim est jusqu'à présent le simulateur le plus utilisé dans la littérature pour les travaux liés au Fog. Cependant, il a deux principales limites:
-La mobilité des nœuds n'est pas prise en charge.

-Les classes liées au réseau et à la gestion des communications sont très limitées et contiennent uniquement des attributs statiques (latence, bande passante). En plus d'une architecture arborescente simple qui est peut représentative des architectures réelles souvent denses et hyper connectée le routage est imposé dans un seul sens : les informations du capteur doivent aller du nœud le plus bas vers le plus haut et la réponse vers les actionneurs doit descendre d'un nœud père à un nœud fils. Cet aspect limite l'étude et les possibilités pour les stratégies de placement de services et nous place dans un contexte peu réaliste.

III- 7 Synthèse

Pour cette synthèse, nous allons essayer de le faire sous forme de tableau en accentuant sur le placement de données (R=Plateforme réelle, S=Simulateur, E=Emulateur.)

Tableau III. 1 : Synthèse des simulateurs et émulateurs

	Placement de données	Fog	IoT	Plate-Forme
IoT-LAB	Non	Non	Oui	R
YOUPI				R
GridSim	Oui	Non	Non	S
EmuFog	Non	Oui	Non	E
FogBeb	Non	Oui	Oui	E
Fog-Torch	Non	Oui	Oui	S
IfogSim	Non	Oui	Oui	S

Au vu des caractéristiques d'iFogSim (passage à l'échelle, modélisation des environnements de Fog et d'IoT), nous avons choisi d'adapter ce simulateur pour mettre en œuvre et évaluer nos stratégies de gestion du placement de données. De plus, ce simulateur a été utilisé dans le cadre de nombreux travaux dans le contexte du Fog et de l'IoT [77, 78].

III- 8 Etat de l'Art

L'informatique géo distribuée, aussi appelé informatique en brouillard ou Fog Computing en anglais est une infrastructure informatique de traitement de données qui a été proposée par Cisco en 2012 [14].

Le terme Fog Computing a été utilisé par Jonathan Bar-Magen Numbarser pour la première fois dans sa thèse 'Fog Computing introduction to a new Cloud Evolution' [49]. Dans sa proposition, il nous présente les limites du Cloud Computing faces aux exigences en QOS des nouvelles applications telles que les applications de l'IOT et à la progression externe des utilisateurs ainsi le trafic du réseau. Il est considéré comme une extension du cloud du cœur jusqu'au périphérie du réseau [50, 51, 13].

Mais aussi il faut savoir que le Fog a la capacité d'héberger des applications de l'IOT qui nécessitent des réponses rapides et en temps contrairement au cloud [51].

Le Fog a pour but d'étendre les capacités et les services offerts par les infrastructures du Cloud Computing.

Après la déclaration de [49], c'est par la suite que Flavio Bonomi reprend le terme en 2012 dans son article Fog and Its role in the [14], sur cet article l'auteur met en exergue les avantages et les limites de ce nouveau paradigme de calcul dans des environnements de types IOT. Les années fil jusqu'en 2015 Cisco Systems, Intel, ARM et Microsoft en Collaboration avec l'université de Princeton fondent l'OpenFog consortium afin de promouvoir le Fog Computing dans les industries. Le fruit de cette collaboration a donné un article [52] qui expose les principales caractéristiques des Architectures Fog.

Il a aussi le but d'optimiser les communications entre un grand nombre d'objets IOT et les services de traitement distants Cloud en tenant compte des volumes de données et de la variabilité de la latence dans un réseau distribué tout en tenant le meilleur contrôle des données transmises [53].

Y a aussi Cisco qui nous propose le paradigme du Fog Computing pour faire face aux problématiques des latences élevées et du trafic réseau important causé par l'utilisation du Cloud [54, 55]. Le Fog Computing est un paradigme qui utilise les ressources disponibles dans des équipements du réseau localisé entre les centres de données du Cloud et les utilisateurs finaux.

Ces équipements sont appelés nœuds de Fog ou Fog node en anglais, c'est au sein de ces nœuds qui sont à la périphérie du réseau où se passe la majeure partie du traitement des données, [56], parmi ces équipements on peut citer les appareils mobiles, les routeurs, les Routeurs, les Points Accès, les stations de base etc.

Le Fog est apparu pour compléter le Cloud pas pour le remplacer donc, ces deux paradigmes sont complémentaires [57]. Sur ce on peut en déduire que ces deux paradigmes sont là pour former une nouvelle plateforme qui supportera les applications de latence courte et prédictibles ainsi les applications qui nécessitent des traitements de données complexes [13, 58].

Donc l'idée est de gérer les requêtes qui nécessitent une courte latences telle que (le trafic routier, parking intelligent, voiture intelligent, arrosage automatique), par des équipements localisés dans le Fog, par contre les requêtes qui nécessitent des traitements complexes (ex photos, flux vidéos, données des réseaux sociaux, etc.) Sont gérés par le Cloud [13].

On constate que ces nœuds de fog et les utilisateurs finaux sont séparés par une distance qui est très variable, de quelques mètres comme les Points d'Accès jusqu'à des kilomètres comme les Points de Présence d'Opérateur (POP) qui sont des serveurs de collecte du réseaux internet. Donc on peut en déduire que dans les infrastructures de types Fog, les équipements modestes en ressources sont localisés près des utilisateurs finaux, et ils donnent des latences courtes, par contre les équipements riches en ressources sont localisés loin des utilisateurs finaux et fournissent des grandes latences et imprédictibles [59].

Le Fog Computing ou Fog Network est un paradigme de Calculs distribué à large échelle et hiérarchique, qui permet d'augmenter les capacités de calcul, de stockages et de communication des serveurs cloud par celles des nœuds intermédiaires situés entre ces serveurs et les utilisateurs finaux [52, 60].

Compte tenu des solutions déjà proposées nous allons montrer notre démarche et l'importance de notre proposition. Celle-ci consiste à réduire considérablement le temps de stockage des données dans le Fog. Dans cette partie nous allons présenter notre simulateur ifogsim qui permet de stocker les données dans le Cloud ou dans le Fog et de montrer en même temps le temps mis pour le stockage des données.

Avec l'acquisition de matériel IoT, nous avons pu réaliser concrètement la proposition de notre système, appliqué dans le domaine de l'agriculture (arrosage automatique).

Conclusion

Le Fog Computing est une architecture informatique qui vise à déplacer le traitement des données et les services informatiques du nuage vers des dispositifs de bord situés à proximité des utilisateurs finaux. Cette approche permet de réduire la latence, d'améliorer la fiabilité et de traiter de grandes quantités de données en temps réel. Les simulateurs de Fog Computing permettent de modéliser et d'étudier les performances et les caractéristiques de cette

architecture, en prenant en compte divers scénarios de déploiement et de chargement de travail. Les résultats de ces simulations peuvent être utilisés pour améliorer la conception et l'optimisation des systèmes de Fog Computing. En conclusion, les simulateurs de Fog Computing sont des outils précieux pour comprendre et optimiser les performances de cette architecture en évolution rapide.

CHAPITRE IV : IMPLEMENTATION ET RESULTATS

IV- 1 Introduction

IFogSim est un simulateur d'environnements de Fog et d'IoT [12]. C'est une extension du simulateur d'environnements de Cloud : CloudSim [79]. iFogSim est conçu pour pouvoir évaluer des stratégies de gestion des ressources applicables aux environnements de Fog et d'IoT en ce qui concerne leur impact sur la latence, la consommation d'énergie, la congestion du réseau et les couts opérationnels.

IFogSim permet de simuler un environnement de Fog et d'IoT incluant des capteurs, des actionneurs, des nœuds de Fog et des centres de données de Cloud. Les applications d'IoT simulées dans iFogSim sont basées sur le modèle Perception Traitement-Action [12]. Dans ce modèle, un ensemble de capteurs (ex. de température) collectent d'abord des informations relatives à l'environnement physique de l'application, puis les publient de manière périodique ou événementielle (ex. si la température dépasse un certain seuil). Ensuite, ces informations sont récupérées puis traitées par un ensemble d'instances de services d'IoT déployées dans le Fog et dans le Cloud. Enfin, conformément aux résultats du traitement, des messages sont envoyés aux actionneurs pour agir sur l'environnement physique de l'application (ex. éteindre le chauffage ou déclencher une alarme).

Afin de réaliser une simulation dans iFogSim, les utilisateurs spécifient l'infrastructure physique de système (i.e. capteurs, actionneurs, nœuds de Fog, liens réseaux, etc.), le scénario (i.e. services d'IoT, flux de données, etc.), leurs stratégies de placement et/ou d'ordonnancement de services, et le temps maximal de la simulation. IfogSim produit en sortie la valeur d'un ou de plusieurs critères (selon les objectifs des utilisateurs) parmi ceux qui sont mentionnés auparavant.

Et pour en beauté ce chapitre de notre mémoire nous allons aussi essayer de faire la réalisation d'un prototype d'arrosage automatique par le biais des matériels Iot et des microcontrôleurs aussi.

IV- 2 Architecture d'IfogSim

La Figure IV.1 montre l'architecture initiale d'iFogSim. Ce simulateur est composé : d'un ensemble d'entités permettant de modéliser les équipements physiques de l'infrastructure, un ensemble d'entités permettant de modéliser les éléments logiques du scénario simulé (ex. les données et les instances des services de l'IoT), et un ensemble d'entités permettant de gérer les ressources de traitement dans les nœuds de Fog et dans les centres de données. Ces ressources sont manipulées pour placer et ordonnancer les instances de services dans les entités physiques afin d'optimiser la latence du service, le trafic réseau, la consommation énergétique ou le coût

opérationnel du système. Ci-après nous décrivons chaque ensemble d'entités définies dans iFogSim.

Les entités physiques : ce sont les modules des équipements physiques trouvés dans une infrastructure de Fog (ex : serveurs, capteurs, actionneurs etc.).

1.FogDevice : cette entité modélise les nœuds de Fog (ex : les switchs, les routeurs, les stations de base, les passerelles, etc.). Elle spécifie les caractéristiques matérielles d'un nœud de Fog comme le modèle du processeur, la taille de la RAM, la capacité du stockage et la bande passante du réseau.

2.Capteur : cette entité modélise les différents capteurs déployés dans l'infrastructure simulée. Elle spécifie les attributs d'un capteur allant de sa connectivité réseau jusqu'aux données capturées.

3.Actionneur : cette entité modélise les actionneurs et les effets de leur action sur l'environnement simulé. Elle spécifie les attributs d'un actionneur comme sa connectivité, sa latence et la passerelle associée.

Les entités Logiques : ce sont des modelés représentant les éléments logiques trouvés dans un environnement informatique (ex : données, service etc.).

1.Tuple : cette entité représente l'unité fondamentale des communications entre les entités physiques (c'est-à-dire les données et leurs métadonnées) dans ifogsim. Un tuple est caractérisé par son type (ex : humidité), le nœud source, le nœud destinataire, la capacité du traitement de données demandée (en million d'instructions) et la taille des données.

2.AppModule : cette entité représente les unités de traitements du scénario simulé qui sont les instances de services de l'IoT. Ces instances de services sont déployées dans les nœuds de Fog et les centres de données. Pour chaque donnée d'entrée (Tuple), une AppModule s'occupe de son traitement et génère des données de sortie qui sont envoyées par la suite à une ou à plusieurs AppModules. Le taux de production des données de sortie est exprimé par un modèle de sélection des données déployé dans le simulateur iFogSim. Par exemple, un taux de sélection de 10% consiste à générer une donnée (i.e. une Tuple) de sortie pour 10 données d'entrée.

3.AppEdge : cette entité représente les dépendances de données existantes entre deux AppModules (i.e. un producteur et un consommateur de données). De plus, une AppEdge spécifie le mode de production de données parmi les deux modes possibles : périodique ou

événementiel. Le modèle de dépendance de données (i.e. les dépendances entre les producteurs et les consommateurs de données) du scénario simulé est représenté par un graphe dirigé acyclique [12] composé d'un ensemble d'AppEdges interconnectées entre elles. Les nœuds du graphe représentent les AppModules et les arêtes représentent les flux de données.

• **Les entités de gestion de ressources** *:* ce sont des modèles représentant les stratégies de gestion de ressources physiques et logiques dans iFogSim (ex. allocation, ordonnancement, migration.).

1. AppModule Placement *:* cette entité définit un ensemble de méthodes permettant de gérer le placement des AppModules dans l'infrastructure simulée. Ainsi, les utilisateurs utilisent ces méthodes pour définir leur stratégie de placement des AppModules afin d'optimiser un ou plusieurs critères parmi ceux qui sont cités au début de cette section. Une fois les AppModules placés dans les nœuds de Fog, ils sont ordonnancés suivant l'entité AppModule Scheduler qui est définie par la suite.

2. AppModule Mapping *:* cette entité définit pour chaque instance de service d'IoT, le nœud de Fog ou le centre de données qui l'héberge. Cette entité est configurée suivant la politique implantée dans l'entité de gestion de ressources AppModule Placement.

3. AppModule Schedule *:* cette entité définit un ensemble de méthodes pour gérer l'ordonnancement des AppModules dans un nœud de Fog ou un centre de données. Les utilisateurs utilisent ces méthodes pour définir leur stratégie d'ordonnancement des AppModules. Le mode d'ordonnancement mis en œuvre par défaut dans iFogSim subdivise équitablement les ressources de traitement entre l'ensemble des AppModules hébergés dans une entité physique de l'infrastructure.

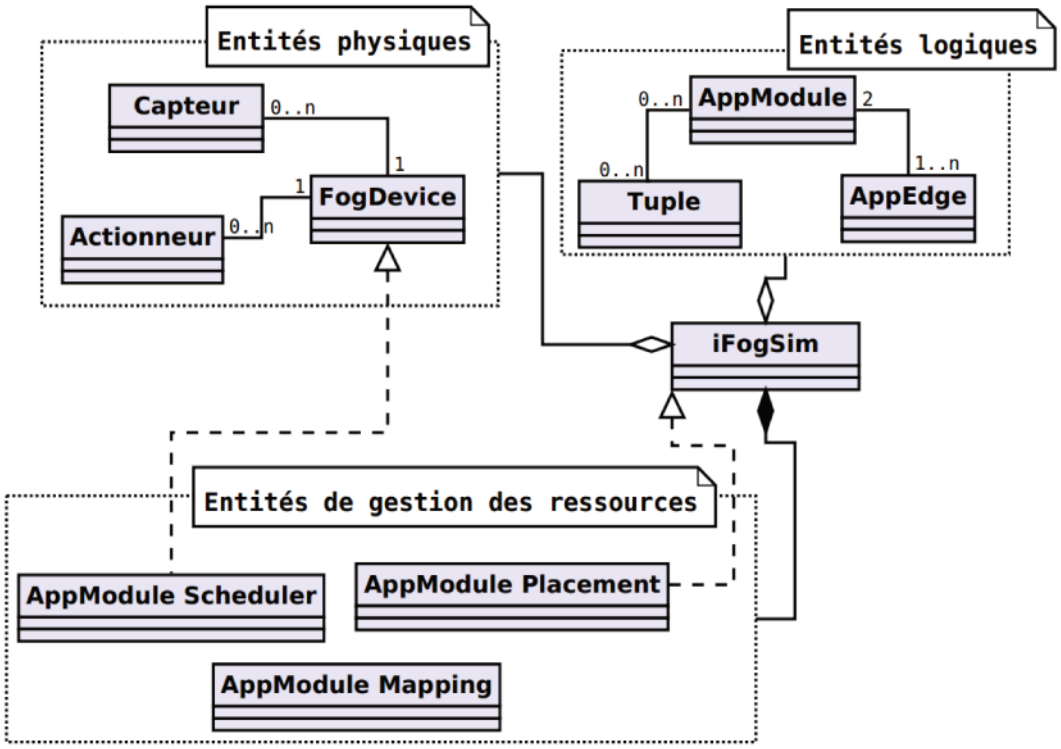


Figure IV. 1 : L'architecture d'Ifogsim

Dans ifogsim, les entités physiques communiquent entre elles en utilisant des événements prédéfinis. Par exemple, il existe un événement pour lancer une instance de service dans un nœud de Fog, un événement pour connecter un capteur à une passerelle, et un événement pour envoyer des données à une autre entité. Les utilisateurs peuvent enrichir le système par l'ajout d'événements pour des cas spécifiques.

IV- 3 Utilisation d'IfogSim

Pour utiliser le simulateur ifogsim, il faut installer d'abord éclipse

IV- 3-1 Eclipse

Eclipse est un environnement de développement intégré libre extensible, universel et polyvalent, permettant de créer des projets de développement mettant en œuvre n'importe quel langage de programmation. Eclipse IDE est principalement écrit en Java (à l'aide de la bibliothèque graphique SWT, d'IBM), et ce langage, grâce à des bibliothèques spécifiques, est également utilisé pour écrire des extensions.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin (en conformité avec la norme OSGi) : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in. Plusieurs logiciels commerciaux sont basés sur le logiciel libre, comme par exemple IBM lotus Notes 8, IBM Symphony ou WebSphere Studio Application Developer.

IV-4 Comment faire fonctionner Ifogsim sur Eclipse

IfogSim permet la modélisation et la simulation des environnements informatiques Fog pour l'évaluation des politiques de gestion et de planification des ressources sur les ressources de périphérie et de cloud dans différents scénarios. Le simulateur prend en charge l'évaluation des politiques de gestion des ressources en se concentrant sur leur impact sur la latence (rapidité), la consommation d'énergie, la congestion du réseau et les coûts opérationnels. Il simule les appareils périphériques, les centres de données cloud et les liens réseau pour mesurer les mesures de performance. Le principal modèle d'application pris en charge par iFogSim est le modèle Sense-Process-Actuate. Dans de tels modèles, les capteurs publient des données sur les réseaux IoT, les applications exécutées sur les appareils Fog s'abonnent et traitent les données provenant des capteurs, et enfin les informations obtenues sont traduites en actions transmises aux actionneurs.

Déjà pour utiliser ifogsim il faut télécharger le code source sur GitHub, et voici les étapes à suivre pour faire le téléchargement :

- Il faut joindre le site web github.com/cloudslab/ifogsim.
- Une fois dans la plateforme vous télécharger le code source sous format zippé

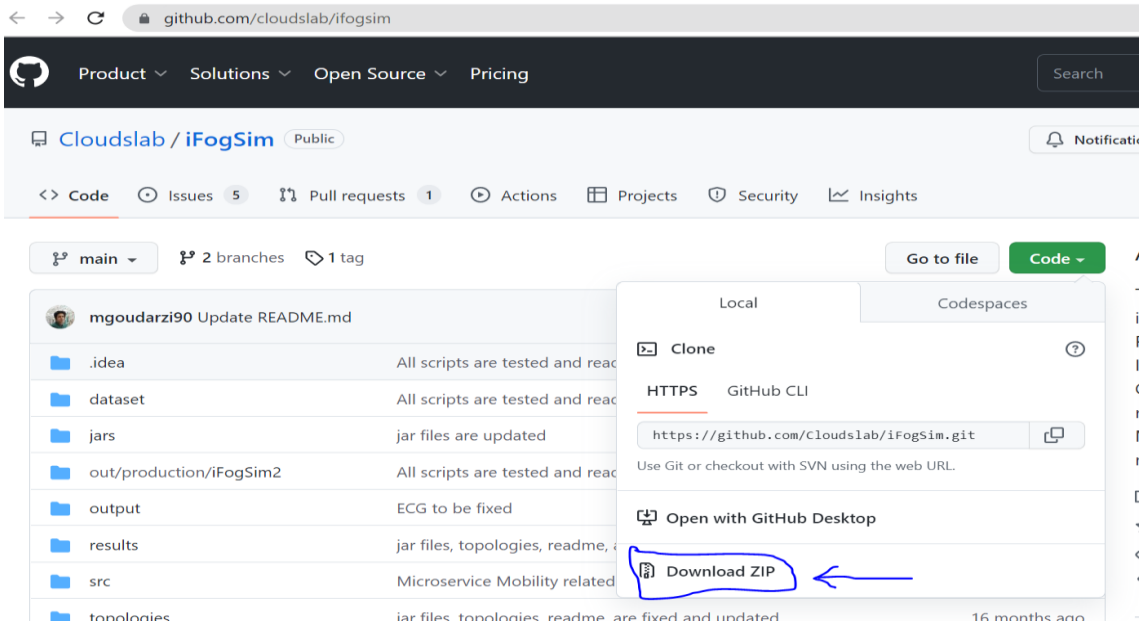


Figure IV. 2 : Téléchargement du projet ifogsim

Après le télécharger du code source il faut :

- ✓ Créer un projet Java
- ✓ Importer le projet ifogsim déjà télécharger, suivre les étapes de création de projet.

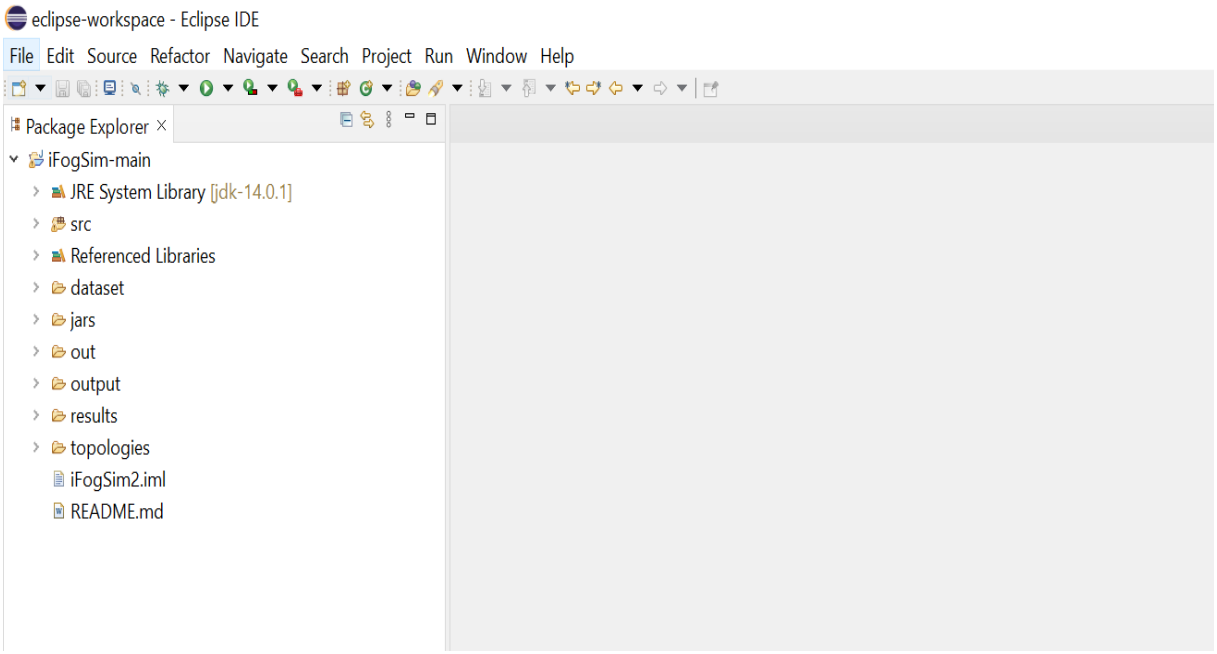


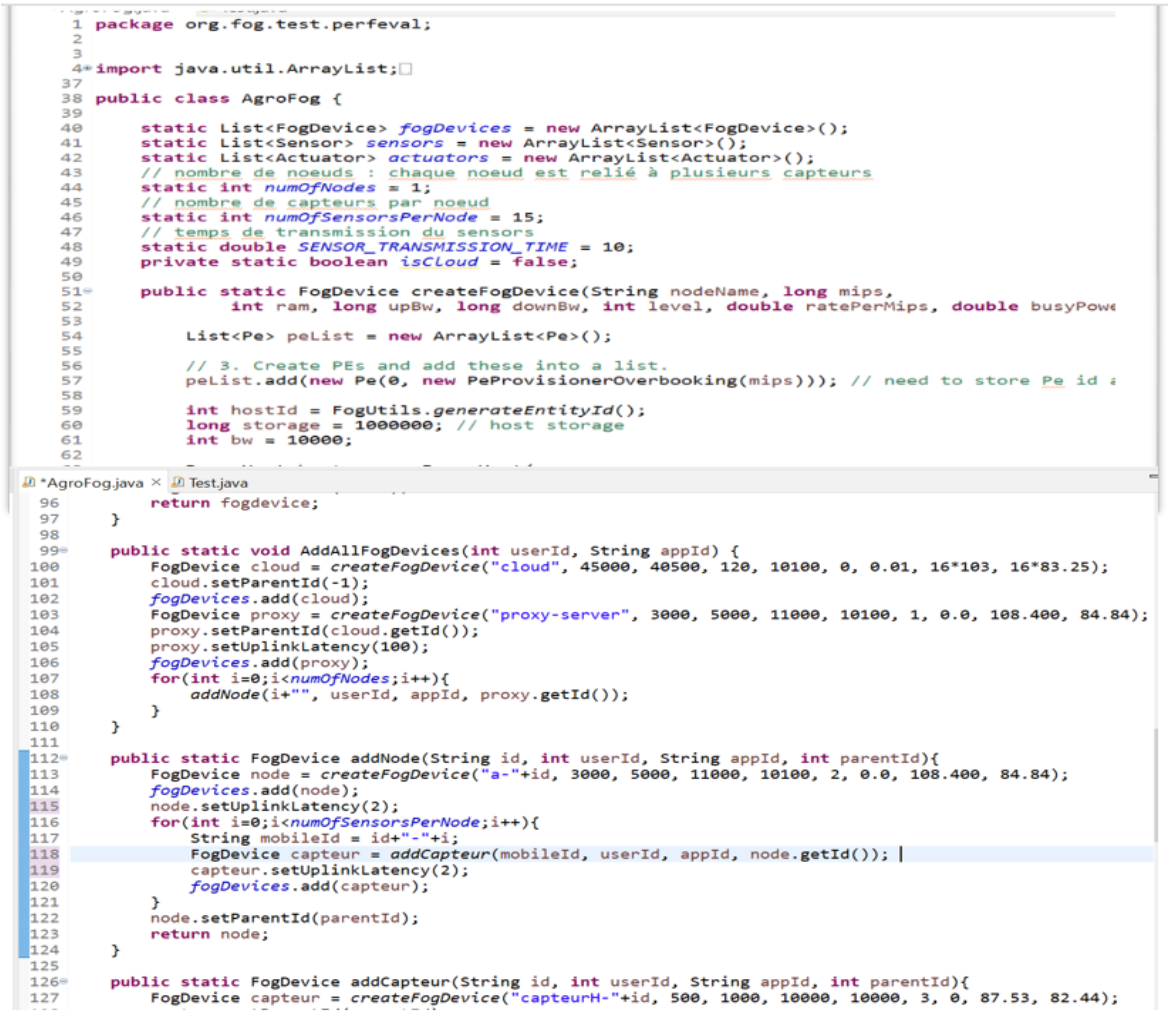
Figure IV. 3 : Importation du projet ifogsim sur eclipse

IV-4-1 Notre Travail sur Ifogsim

Après avoir importé le code source, il faut maintenant écrire son code de travail. Dans nos recherches nous avons accentués notre travail sur l'agriculture, essayer de stocker les données des capteurs d'humidité du sol dans nos nœuds de Fog (dans le Fog Computing) et dans le Cloud, et d'étudier le temps de latence, l'énergie utiliser et même la bande passante.

Donc nous avons écrit un code avec des capteurs et des nœuds Fog et un serveur Cloud pour simuler notre travail dans un premier temps avant de réaliser le travail avec des objets IoT.

Dans les lignes qui suivent nous allons essayer de vous montrer et de vous expliquer en même temps le code de simulation de notre travail.



```
package org.fog.test.perfeval;


import java.util.ArrayList;

public class AgroFog {

    static List<FogDevice> fogDevices = new ArrayList<FogDevice>();
    static List<Sensor> sensors = new ArrayList<Sensor>();
    static List<Actuator> actuators = new ArrayList<Actuator>();
    // nombre de noeuds : chaque noeud est relié à plusieurs capteurs
    static int numOfNodes = 1;
    // nombre de capteurs par noeud
    static int numOfSensorsPerNode = 15;
    // temps de transmission du sensors
    static double SENSOR_TRANSMISSION_TIME = 10;
    private static boolean isCloud = false;

    public static FogDevice createFogDevice(String nodeName, long mips,
            int ram, long upBw, long downBw, int level, double ratePerMips, double busyPow

        List<Pe> peList = new ArrayList<Pe>();

        // 3. Create PEs and add these into a list.
        peList.add(new Pe(0, new PeProvisionerOverbooking(mips))); // need to store Pe id

        int hostId = FogUtils.generateEntityId();
        long storage = 1000000; // host storage
        int bw = 10000;
```

*AgroFog.java × Test.java

```
        return fogdevice;
    }

    public static void AddAllFogDevices(int userId, String appId) {
        FogDevice cloud = createFogDevice("cloud", 45000, 40500, 120, 10100, 0, 0.01, 16*103, 16*83.25);
        cloud.setParentId(-1);
        fogDevices.add(cloud);
        FogDevice proxy = createFogDevice("proxy-server", 3000, 5000, 11000, 10100, 1, 0.0, 108.400, 84.84);
        proxy.setParentId(cloud.getId());
        proxy.setUplinkLatency(100);
        fogDevices.add(proxy);
        for(int i=0;i<numOfNodes;i++){
            addNode(i+"", userId, appId, proxy.getId());
        }
    }

    public static FogDevice addNode(String id, int userId, String appId, int parentId){
        FogDevice node = createFogDevice("a-"+id, 3000, 5000, 11000, 10100, 2, 0.0, 108.400, 84.84);
        fogDevices.add(node);
        node.setUplinkLatency(2);
        for(int i=0;i<numOfSensorsPerNode;i++){
            String mobileId = id+"-"+i;
            FogDevice capteur = addCapteur(mobileId, userId, appId, node.getId());
            capteur.setUplinkLatency(2);
            fogDevices.add(capteur);
        }
        node.setParentId(parentId);
        return node;
    }

    public static FogDevice addCapteur(String id, int userId, String appId, int parentId){
        FogDevice capteur = createFogDevice("capteurH-"+id, 500, 1000, 10000, 10000, 3, 0, 87.53, 82.44);
```

Figure IV. 4 : code de simulation sur ifogsim 1

```
158         final AppLoop loop1 = new AppLoop(new ArrayList<String>() {{
159             add("CAPTEUR");
160             add("donnée du cap");
161             add("slot-detector");
162             add("PTZ_CONTROL");
163         }});
164
165         List<AppLoop> loops = new ArrayList<AppLoop>(){{add(loop1);}};
166         application.setLoops(loops);
167
168         return application;
169     }
170 }
```

Figure IV. 5 : code de simulation sur ifogsim 2

Dans notre code de simulation, nous avons créé des nœuds de Fog c'est le FogDevices qui les symbolisent, des sensors et des actionneurs qui représentent les objets Iots. Dans notre code nous avons créé un serveur proxy qui lie directement les nœuds de Fog et le serveur Cloud, ce qui laisse à entendre qu'on a créé dans notre code un serveur Cloud.

Et pour notre travail de simulation les sensors et actionneurs qui représentent les capteurs d'humidité du sol et de température, détectent l'humidité du sol en valeur numérique et la température de l'environnement l'envoie directe au nœud de Fog le plus proche, avant d'être envoyé dans le Cloud. Et pouvons voir les résultats de notre simulation qui s'accentuent sur la latence, l'énergie utilisé et la bande passante.

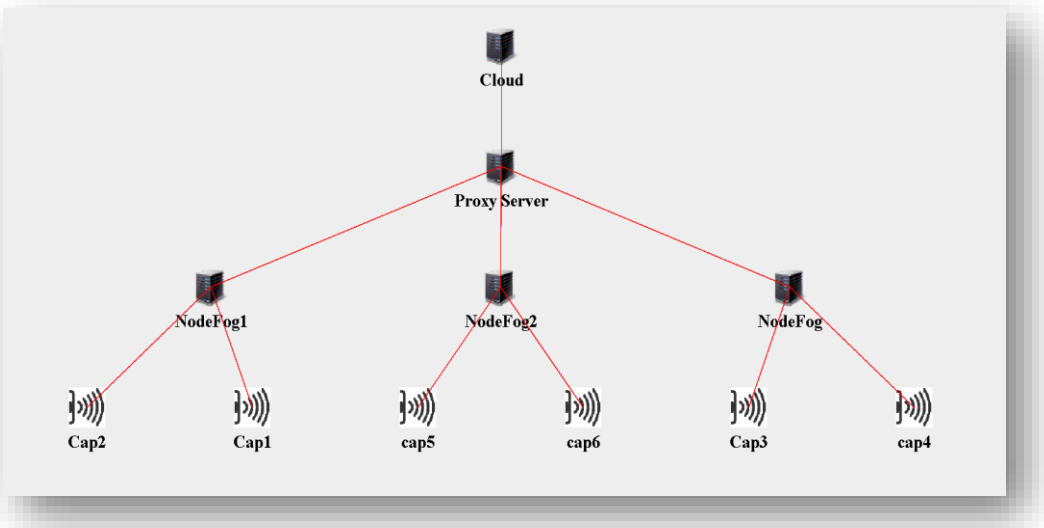


Figure IV. 6 : Architecture de simulation des capteurs sur ifogsim

Et nous avons ici l'architecture de notre simulation dans ifogsim, qui comporte six capteurs, trois nœud de Fog, un serveur proxy et un serveur Cloud

Tableau IV. 1 : valeurs des paramètres des dispositifs du scenario basé sur le Cloud.

Paramètres	**Proxy**	**Cloud**
CPU	3000	45000
RAM	5000	40500
UpLink BandWidth	11000	120
DownLink BandWidth	10100	10100
Level	1	0
RatePerMips	0	0,01
Busy Power (watt)	108,400	16*103
Idle Power (watt)	84,4	16*83,25

Le **tableau IV.1** nous présente les valeurs de paramètre de la configuration du routeur et aussi du serveur cloud crées lors de la simulation de nos scenarios basés sue le Cloud.

Tableau IV. 2 : Données des paramètres du scenario basé sur le Fog

Paramètres	**Fog**	Proxy	**Cloud**
CPU	3000	3000	45000
RAM	5000	5000	40500
UpLink BandWidth	11000	11000	120
DownLink BandWidth	10100	10100	10100
Level	2	1	0
RatePerMips	0	0	0,01
Busy Power (watt)	108,400	108,400	16*103
Idle Power (watt)	84,4	84,4	16*83,25

Le **tableau IV.2** nous présente les valeurs de paramètre de configuration du serveur Fog, du serveur proxy et aussi du serveur cloud crées lors de la simulation de nos scenarios avec les trente capteurs de température et d'humidité basés sur le Fog. Dans les paramètres de configuration, on a en première lieu le CPU (le processeur), la mémoire de la RAM, la bande passante montante et descendante, le niveau, le taux, la puissance occupée et la puissance inactive.

IV-4-2 Résultats et Discussion :

On a ici les paramètres du serveur proxy et du serveur Cloud, en se basant sur le CPU, la RAM, la bande passante montante et descente etc. Et après simulation nous avons les résultats suivants :

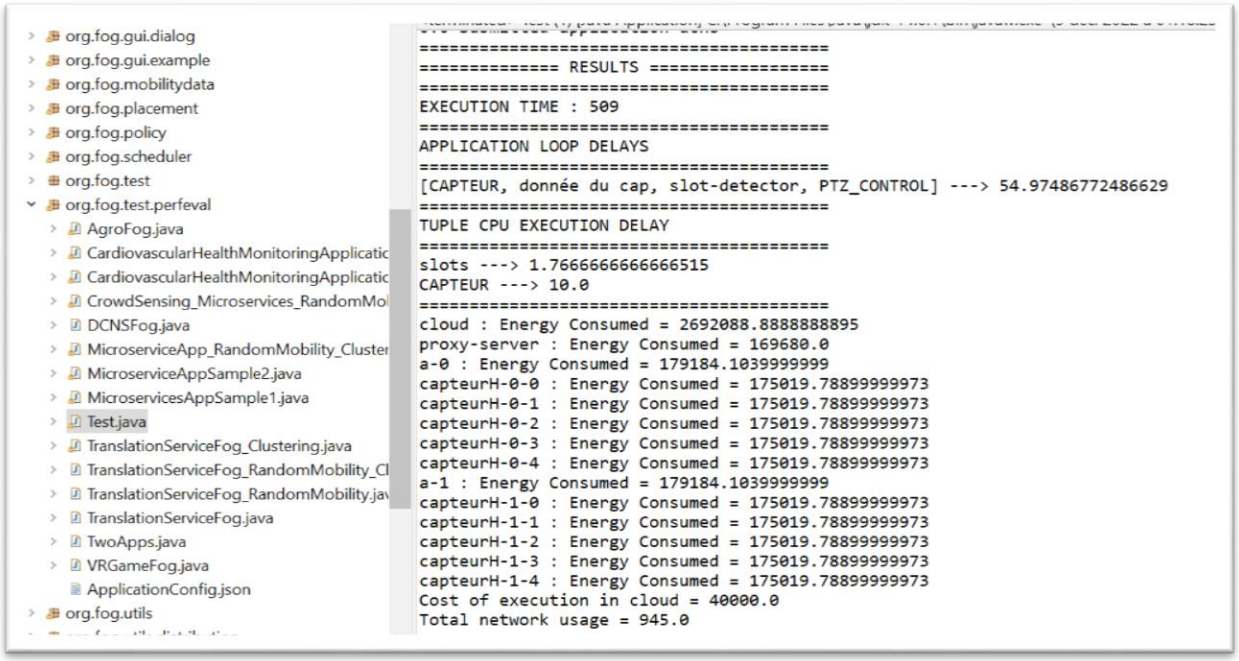


Figure IV. 7 : Résultats de la simulation sur ifogsim

IV-4-3 La latence et de la bande passante :

Le temps et la bande passante sont importants dans tous ce qu'on fait dans le monde de la technologie, plus particulièrement dans le domaine de l'informatique, donc il faut faire tout pour réduire le temps et la consommation de la bande passante, surtout dans les environnements qui nécessitent des requêtes rapides. Et l'un des principaux avantages du Fog est la réduction du temps de latence et de la bande passante surtout dans les environnements qui ont besoin des requêtes rapides, elle évite les accès courant au cloud et opère les calculs au pourtour du réseau pour avoir une réponse très rapide, ce qui diminue le temps de latence et la bande passante. La latence et l'utilisation du réseau (bande passante) sont calculées en utilisant l'équation de [80].

Le tableau IV.3 nous présente la latence qu'on a sur le Fog et le Cloud en fonction des augmentations des capteurs, et le tableau 4 celle de la bande passante.

Tableau IV. 3 : résultats des simulations sur la latence basée sur le Fog et Cloud

Capteurs	15	20	25	30
Fog latence	0,713	0,87	0,94	1,0052
Cloud latence	0,770	0,92	1,0686	1,1542

Tableau IV. 4 : résultats des simulations sur la bande passante basés sur le Fog et Cloud

Capteurs	15	20	25	30
Fog NW	991	1644,3	2220,75	2849,742
Cloud NW	1091,475	1739	2525	3272,157

On a aussi les résultats de notre réalisation en termes de temps, le temps de latence qu'il a fait pour stocker les données dans notre nœud de Fog et celui pour stocker les données dans le Cloud.

Tableau IV. 5 : Résultats de notre réalisation sur les données envoyées sur le Fog et le Cloud en fonction du temps.

Données du Capteur(g/kg air sec et %)	979	980	982	979
Fog latence(ms)	61,55	35,98	34,04	31,6
Cloud Latence(ms)	2194,09	688,34	697,07	788,06

D'après nos résultats qui se figure sur **le tableau IV.5**, nous voyons nettement que pour stocker les données dans le Cloud, il te faut beaucoup de temps pour le faire, tant disque le temps que fait le Fog pour stocker les données c'est vraiment satisfaisant.

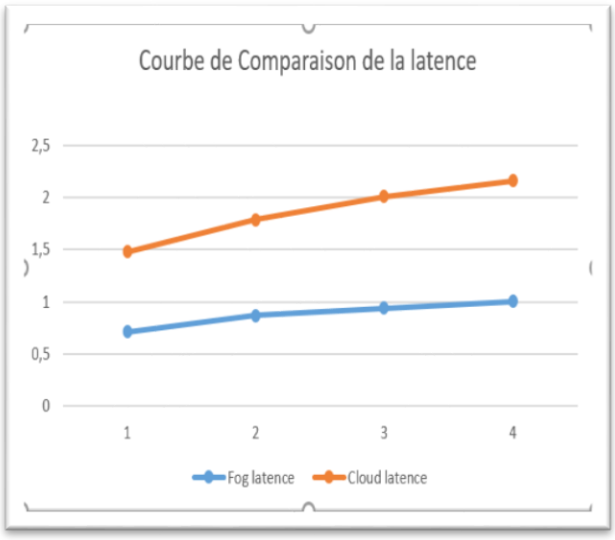
IV-4-4 Comparaison de la latence et de la bande passante

On voit que dans les résultats de nos simulations, on peut voir qu'avec le Fog Computing la latence est plus réduite tant disque au niveau du Cloud le temps est plus important, **la figure IV.7** nous montre la comparaison des temps de latence entre le Fog et le Cloud. Dans nos simulations on augmente progressivement le nombre de capteurs pour voir les différences sur la latence et la bande passante. **La figure IV.8** présente la comparaison de la bande passante entre le Fog et le Cloud en jouant toujours sur les capteurs.

Et d'après les résultats on peut voir la bande passante et le temps de latence est beaucoup plus important avec le Cloud, on peut souligner aussi qu'avec le Fog Computing le temps et la bande passante sont trop économique.

Les résultats expérimentaux pour le brouillard et la mise en œuvre basée sur le cloud pour les deux paramètres d'évaluation, à savoir la latence et l'utilisation du réseau, démontre l'efficacité de l'architecture basée sur le brouillard proposée pour un système d'arrosage automatique. Le fog computing permet d'effectuer le traitement des données localement, à proximité des capteurs et des actionneurs du système d'arrosage automatique. Cela réduit la latence, c'est-à-dire le temps de traitement et de réponse du système. Lorsque des capteurs détectent une variation dans les conditions environnementales (par exemple, une baisse d'humidité du sol), le fog computing peut permettre une réaction immédiate en ajustant les cycles d'arrosage en temps réel. Cela garantit que les plantes reçoivent l'eau nécessaire au bon moment, ce qui est crucial pour leur santé. Lorsque le fog computing est utilisé dans un système d'arrosage automatique, il peut y avoir une communication constante entre les capteurs, les actionneurs et les nœuds de calculs locaux. Ces dispositifs échangent des données en temps réel pour surveiller les conditions environnementales, prendre des décisions d'arrosage et activer les actionneurs pour délivrer l'eau nécessaire aux plantes. Une bande passante suffisante est donc nécessaire pour garantir que ces échanges de données se font de manière fluide et sans délais, afin que le système puisse réagir rapidement aux changements environnementaux.

De plus, les résultats nous aident également à reconnaître le potentiel de calcul du brouillard dans les paramètres IoT où plus les délais d'exécution court, plus ils sont très souhaitables. En résumé, la faible latence et la faible utilisation du réseau rendent l'architecture basée sur le brouillard plus pratique pour les applications et les scénarios en temps réel.



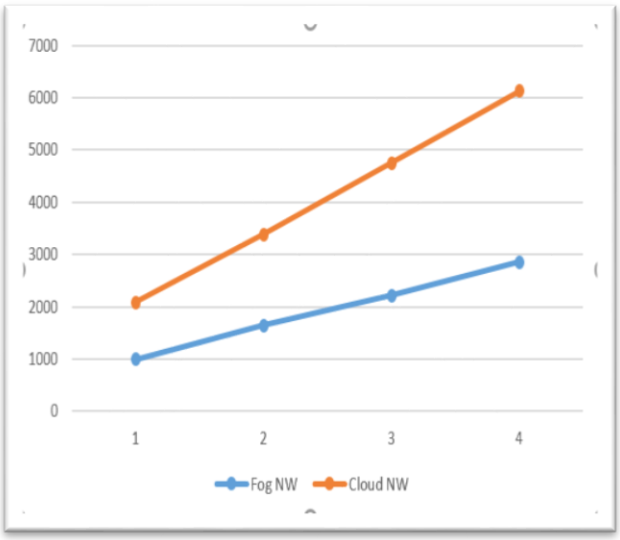


Figure IV. 8 : comparaison de la latence. **Figure IV. 9 : comparaison de la bande passante.**

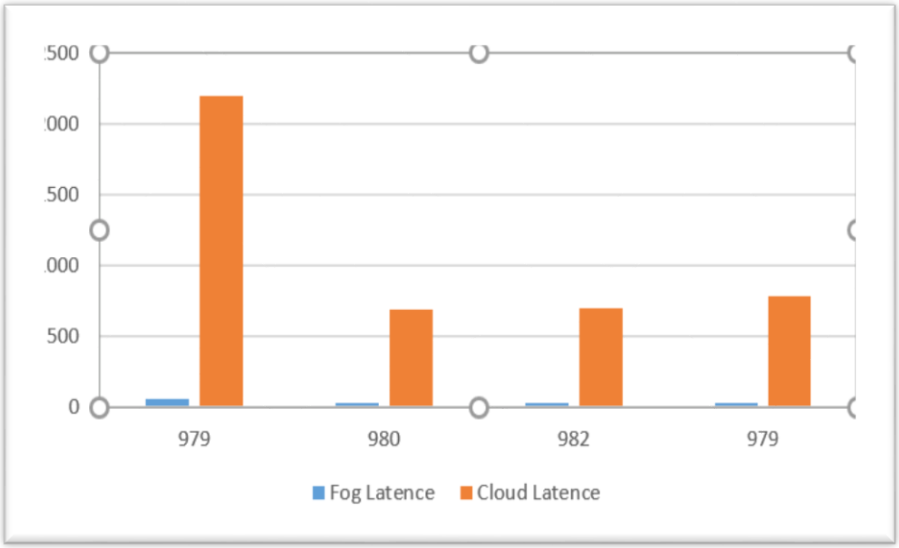


Figure IV. 10 : Comparaison du temps de latence du Fog et du Cloud.

Et avec les résultats de la figure IV.10 on peut confirmer que le Fog Computing est plus économique en terme de temps et de bande passante. On voit pour une même donnée du capteur le Cloud met 40 fois plus de temps pour stocker la valeur envoyée par le capteur. Étant donné que le fog computing effectue le traitement des données à proximité de leur source, il réduit la latence, c'est-à-dire le temps de déplacement des données entre l'appareil qui les a générés et le centre de données loin. Cela permet d'obtenir des résultats plus rapidement, ce qui peut être

crucial pour les applications en temps réel, telles que les voitures autonomes ou les systèmes de surveillance.

IV-5 LES MATERIELS UTILISES

Pour notre travail nous avons utilisé les matériels suivants :

- Arduino Mega 2560
- Relais électrique
- Capteur d'humidité du sol
- Pompe à eau
- Raspberry Pi 3

IV-5-1 RASPBERRY PI

Le Raspberry Pi est un petit ordinateur mono carte développé par la Fondation Raspberry Pi au Royaume-Uni. Il a été conçu pour promouvoir l'enseignement de l'informatique de base dans les écoles et les pays en développement. Le Raspberry Pi est un appareil peu coûteux, suffisamment petit pour tenir dans la paume de la main, mais suffisamment puissant pour exécuter une grande variété d'applications. Il possède une carte mère de la taille d'une carte de crédit et peut être alimenté par un adaptateur micro-USB.

Le Raspberry Pi dispose d'une série d'entrées et de sorties, notamment des ports USB, un port HDMI et une prise audio de 3,5 mm, ainsi qu'un certain nombre de broches d'entrée/sortie à usage général (GPIO) qui peuvent être utilisées pour connecter des capteurs, des moteurs et d'autres dispositifs. Il peut fonctionner sous divers systèmes d'exploitation, dont Linux, et peut être utilisé pour un large éventail d'applications, telles que les centres multimédias, les systèmes domotiques et même les serveurs à petite échelle.

Dans l'ensemble, le Raspberry Pi est un appareil polyvalent et convivial qui a acquis une grande popularité auprès des amateurs, des éducateurs et des professionnels en raison de son faible coût et de ses performances élevées. Et mon choix s'est porté sur le Raspberry Pi 3.

Voici un tableau comparatif des différents types de Raspberry :

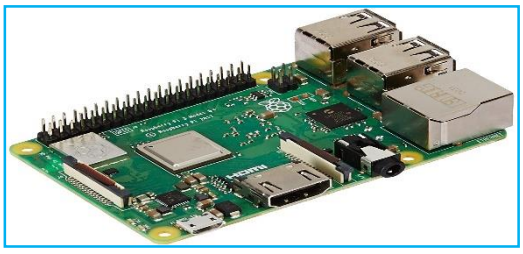


Figure IV. 11 : Raspberry Pi 3

Tableau IV. 6 : Comparaison des différents types de Raspberry

Raspberry Pi	Date de sortie	USB Ports	CPU	RAM	Bluetooth	WiFi
Raspberry Pi 1B	févr-12	2x USB 2.0	700MHz	512MB	Non	Non
Raspberry Pi 1B+	juill-14	4x USB 2.0	700MHz	512MB	Non	Non
Raspberry Pi 1A+	nov-14	1x USB	700MHz	512MB	Non	Non
Raspberry Pi 2	févr-15	4x USB	900MHz	1GB	Non	Non
Raspberry Pi Zéro	nov-15	1x Micro USB	1GHz	512MB	Non	Non
Raspberry Pi 3	févr-16	4x USB	1,2GHz	1GB	4.1 LE	Oui
Raspberry Pi Zéro W	févr-17	1x Micro USB	1GHz	512M	4.1	Oui
Raspberry Pi 3B+	mars-18	4x USB 2.0	1.4GHz	1GB	4.2 BLE	Oui
Raspberry Pi 3A+	nov-2018	1x USB 2.0	1.4GHz	512MB	4.2 BLE	Oui
Raspberry Pi 4	juin-19	2X USB 3.0, 2X USB 2.0	1.5GHz	1GB, 2GB, 4GB, or 8GB	5.0 BLE	Oui
Raspberry Pi 400	nov-20	2X USB 3.0, 2X USB 2.0	1.8GHz	4GB	5.0 BLE	Oui
Raspberry Pi Pico	janv-21	USB C	133MHz	264KB	Non	Non

IV-6- LES BRANCHEMENTS

IV-6-1 LA CONNEXION DU CAPTEUR D'HUMIDITE DU SOL AVEC L'ARDUINO

Pour connecter un capteur d'humidité su sol à l'Arduino Mega 2560, nous devons utiliser quatre fils de câblage. Le fil violet doit être connecté au pin 5V de l'Arduino, le fil bleu doit être connecté au pin GND de l'Arduino, le fil vert doit être connecté au pin Tx01 de l'Arduino et le fil jaune ou vert doit être connecté au pin A01 de l'Arduino. Vérifiez que les connections sont solidement en place et que tous les fils sont bien attachés aux bonnes pins.

Tableau IV. 7 : Branchement du capteur d'humidité

CAPTEUR D'HUMIDITE	CONNECTEUR	ARDUINO
GNG	**BLEU**	**GND**
VCC	**VIOLET**	**5V**
D0	**VERT**	**Tx0 1**
A0	**JAUNE**	**A01**

IV-6-2 CONNEXION DU RELAIS AVEC L'ARDUINO

Pour connecter un relais électrique à l'Arduino Mega 2560, nous devons utiliser trois fils de câblage. Voici comment procéder :

Localisez les pins 5V, GND, et le PIN 13 sur votre Arduino Mega 2560. Prenez le fil rouge et connectez-le au pin 5V de l'Arduino. Prenez le fil noir et connectez-le au pin GND de l'Arduino. Prenez le fil jaune et connectez-le au pin PIN 13 de l'Arduino. Vérifiez que les connections sont solidement en place et que tous les fils sont bien attachés aux bonnes pins.

Tableau IV. 8 : branchement du relais électrique

RELAIS	CONNECTEUR	ARDUINO
GND	**NOIR**	**GND**
VCC	**ROUGE**	**5V**
IN3	**JAUNE**	**PIN 13**

IV-6-5 CODE DU PROJET SUR ARDUINO

Consiste à la création du programme qui permet le fonctionnement des objets connectés à Arduino exemple pour faire l'arrosage automatique sous figure suivante :

```
//////////////*******Variable Capteur Humidite Sol*****//////////
int vccPin = 1; // D0
long dataPin = A1 ; //A1
long humidite; //valeur en pourcentage du capteur

//////////////////******Variable Relais****///////////
const int pump1 = 13;    const int L2 =2;

//////////////////////////Led témoin***************////////////////
byte L3=  3; //Led temoin de
byte L4 = 4; //Led temoin d'humidité

//////////////////////************mesure humidité**********////////////////
int readHumidity() {
digitalWrite(vccPin, HIGH);
delay(5000); // you need to test how long you pre-power before measurement
int value = analogRead(dataPin);
//Serial.print("humidite sol VN : ");
//Serial.println(analogRead(dataPin));
digitalWrite(vccPin, LOW);
return value;
                  }

//////////////////////************** arrosageAutomatique **********////////////////

void arrosageAutomatique() {
    //Serial.begin(9600);
```

```
//////////////////////************** arrosageAutomatique **********/////

void arrosageAutomatique() {
    //Serial.begin(9600);
    digitalWrite(L6, HIGH); // LED allumée

     //Serial.println("*** Arrosage automatique activée ***");
     //Serial.println(" Niveau Humidité Sol: ");
     Serial.println(readHumidity());
     long m ;
     if (readHumidity()>=910)
        {
          m += millis();
          //Serial.println("*** Arrosage en cours ***");
          digitalWrite(L2, HIGH); // LED allumée
          digitalWrite(L4, LOW); // LED off
          digitalWrite(L3, HIGH); // LED allumée
          digitalWrite(pump1, LOW); // pump1 activated
          delay(5000);
          //Serial.print("Pourcentage humidité: 100%");
          }

       else if ((readHumidity()>=650)&&(readHumidity()<=900))
        {
          //Serial.println("Pourcentage humidité: 60%");
          m += millis();
```

Figure IV. 12 : code de l'arrosage automatique sur le logiciel Arduino 1

```
            //Serial.println("*** Arrosage en cours ***");
            digitalWrite(L2, HIGH); // LED allumée
            digitalWrite(L4, HIGH); // LED off
            digitalWrite(L3, LOW); // LED allumée
            digitalWrite(pump1, LOW); // pump1 activated
            delay(500);
           }
           else if ((readHumidity()>=-50)&&(readHumidity()<=530))
          {
            //Serial.println("Pourcentage humidité: 100%");
            digitalWrite(pump1, HIGH); // pump1 deactivated
            digitalWrite(L2, LOW); // LED off
            digitalWrite(L3, LOW); // LED off
            digitalWrite(L4, HIGH); // LED off
             }

         void setup() {

Serial.begin(1000000);
pinMode(L2, OUTPUT); //Led temoin relais ou pompe *****bleu*****
pinMode(L3, OUTPUT); //LED témoin secheresse ******rouge****
pinMode(L4, OUTPUT); //Led temoin humidité ***********vert******
pinMode(L5, OUTPUT); //LED témoin Remplisage reservoir ******rouge****
pinMode(pump1, OUTPUT); // variant low/high //POMPE
pinMode(vccPin, OUTPUT); //broche sortie capteur sol D0
digitalWrite(vccPin, LOW);
```

```
            digitalWrite(L4, HIGH); // LED off
             }

         void setup() {

Serial.begin(1000000);
pinMode(L2, OUTPUT); //Led temoin relais ou pompe *****bleu*****
pinMode(L3, OUTPUT); //LED témoin secheresse ******rouge****
pinMode(L4, OUTPUT); //Led temoin humidité ***********vert******
pinMode(L5, OUTPUT); //LED témoin Remplisage reservoir ******rouge****
pinMode(pump1, OUTPUT); // variant low/high //POMPE
pinMode(vccPin, OUTPUT); //broche sortie capteur sol D0
digitalWrite(vccPin, LOW);

pinMode(pump1, OUTPUT); // variant low/high

void loop() {
  arrosageAutomatique();
       digitalWrite(L5, LOW);

}
```

Figure IV. 13 : code de l'arrosage automatique sur le logiciel Arduino 2

Après la création du programme, on vérifie le programme puis le compile (transforme le code source écrit dans un langage source en langage cible, pour qu'il puisse être exploité. Enfin, on téléverse le programme dans la carte Arduino via un câble de connexion USB pour l'enregistrer dans cette dernière.

IV-6-6 RESULTATS OBTENU AVEC LE CAPTEUR D'HUMIDITE DU SOL

Nous montrons le résultat obtenu via le moniteur de série Arduino on remarque les divers mesures capté par le capteur d'humidité du sol, lorsqu'il était en état de sec et en état humide.

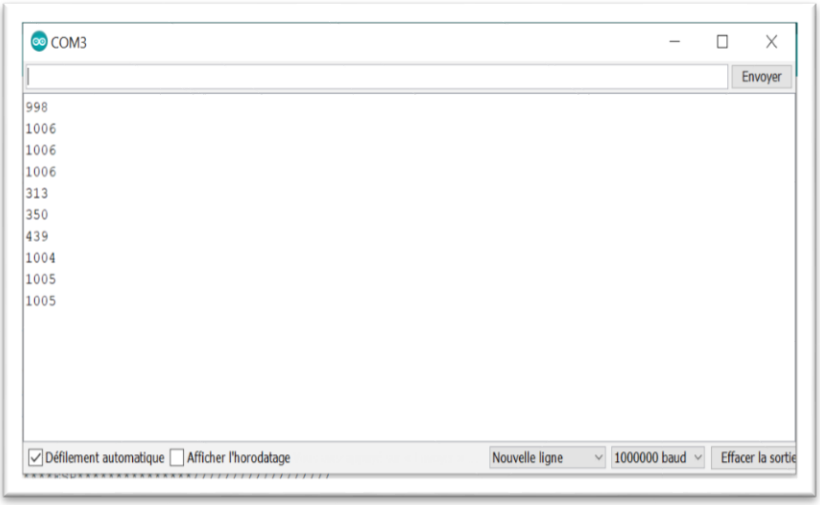


Figure IV. 14 : Résultats du capteur d'humidité du sol

IV-6-7 ORGANIGRAMME DE FONCTIONNEMENT POUR L'ARROSAGE AUTOMATIQUE AVEC LE NŒUD DE FOG (RASPBERRY PI)

Si nous voulons faire déclencher l'arrosage automatique, d'abord on essaye de programmer le capteur d'humidité selon notre besoin. Dans notre projet nous l'avons programmé de telle sorte qu'il capte des valeurs entre 650 et 1000 l'arrosage se déclenche automatiquement, sinon le sol est humide. Si le système détecte que les valeurs sont dans l'intervalle donnée, il demande directement au pompe à eau de faire l'arrosage. Et d'envoyer les données sur notre nœud de Fog, qui est représenté ici par le Raspberry pi et aussi dans le Cloud.

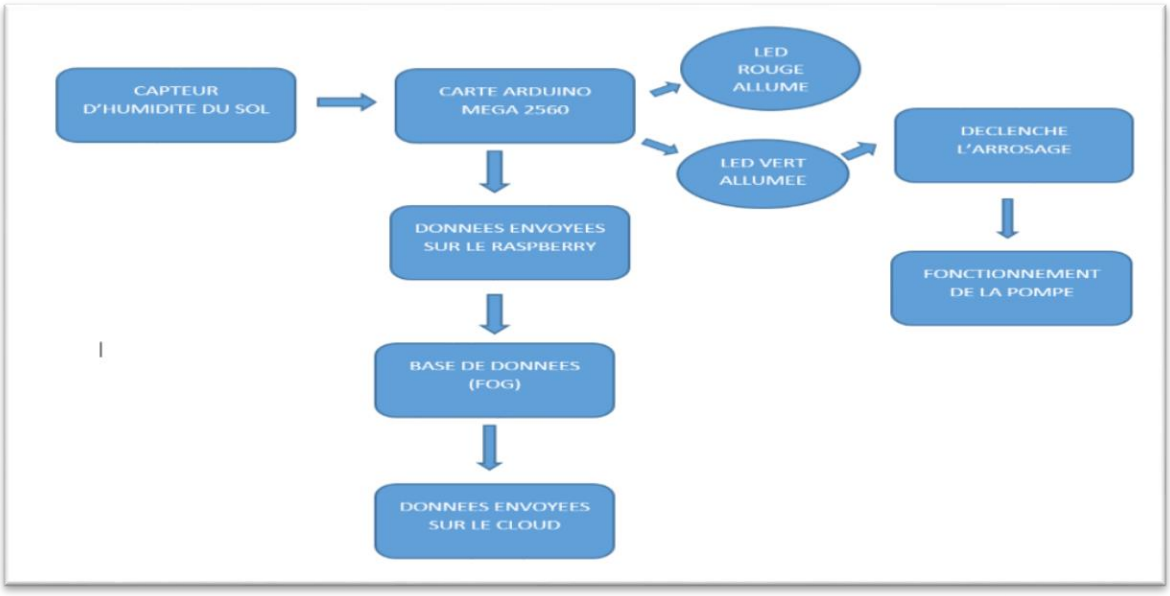


Figure IV. 15 : Organigramme de l'arrosage automatique

IV-6-8 ARCHITECTURE DE NOTRE TRAVAIL

Cette infrastructure comprend des capteurs d'humidité du sol, un ensemble de nœuds de Fog, représenté par les Raspberry, un ensemble de centres de données et un microcontrôleur Arduino.

Dans ce système, les capteurs (producteurs de données) considérés collectent des informations relatives à l'environnement physique, des valeurs de l'humidité du sol. Ces informations sont envoyées aux nœuds de Fog. À la réception, les nœuds de Fog traitent les informations envoyées pour prendre des décisions ou agir sur l'environnement (activer l'arrosage).

Les capteurs d'humidité du sol sont connectés directement à notre Arduino par des fils de connexion, ce dernier connecté directement avec le Raspberry pi par liaison serial. Dans notre infrastructure nous avons des Raspberry pi qui nous sert de nœud de Fog, dans lequel on installer notre base de donnée pour stocker les données des capteurs avant de les envoyées au Cloud. Dans notre infrastructure nous voyons que les données sont transférées dans un temps réduit dans notre nœud de Fog que dans notre Cloud.

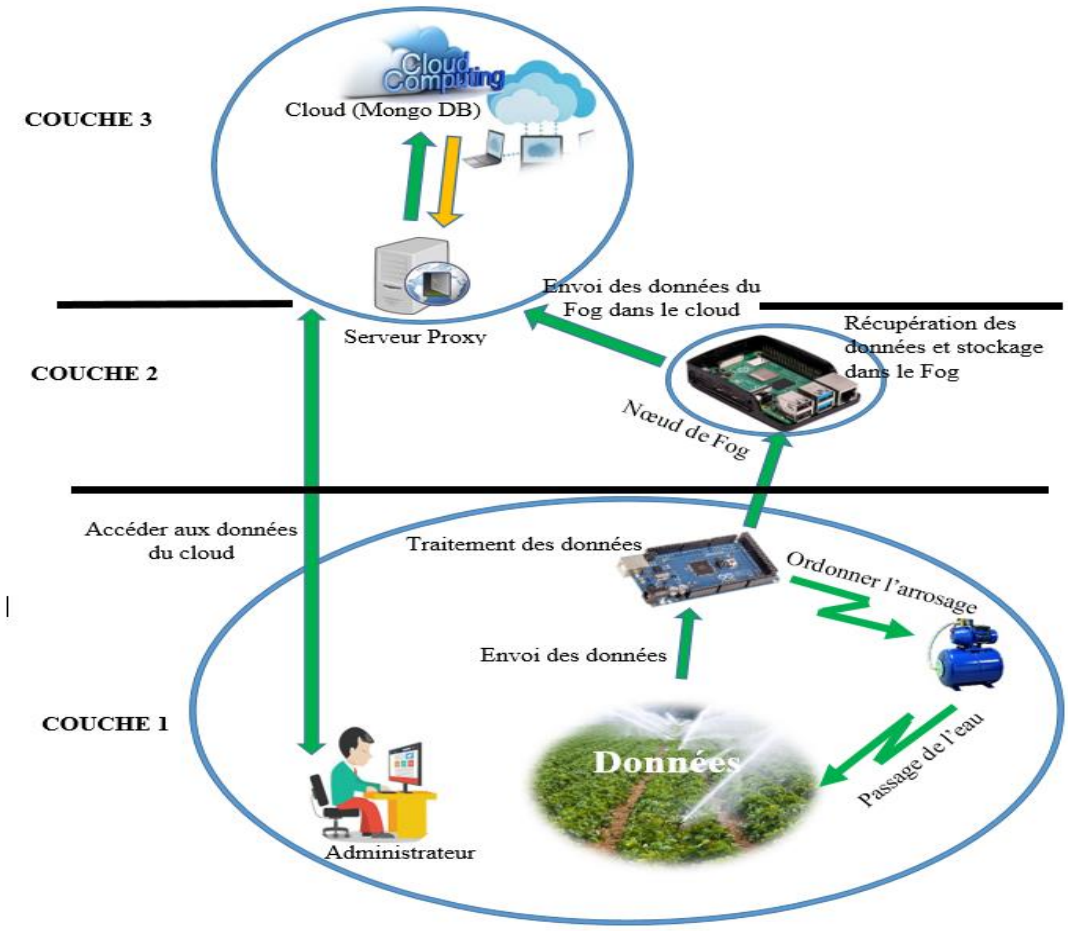


Figure IV. 16 : Architecture du système

IV-6-9 LE BRANCHEMENT REELE

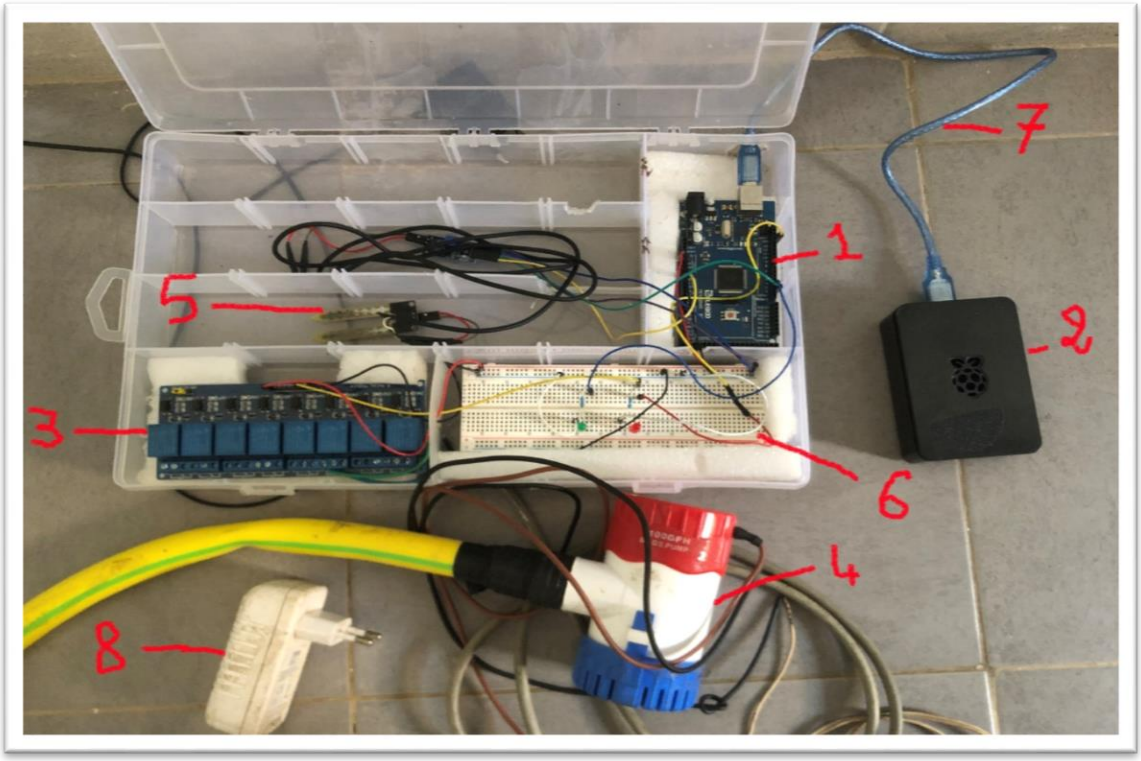


Figure IV. 17 : Branchement réelle du système

Nous avons ici notre réalisation avec nos branchements, qu'on va vous décrire.

- 1-Nous avons ici notre Arduino Méga 2560.
- 2-En 2 nous avons notre Raspberry qui nous sert de nœud de Fog.
- 3-Nous avons le relais électronique.
- 4-Et en 4 nous notre pompe à eau.
- 5-En 5 nous avons notre capteur d'humidité du sol.
- 6-En 6 nous avons le breakbord qui nous permet de faire les connexions.
- 7-Nous avons ici le câble serial qui lie notre système d'arrosage avec notre nœud de Fog.
- 8-Et enfin en 8 nous avons alimentation de notre pompe à eau.

IV-7 INSTALLATION ET CONFIGURATION DE LA BASE DE DONNEE

IV-7-1 SQLITE 3

SQLite est une bibliothèque en langage C qui implémente un moteur de base de données SQL petit, rapide, autonome, très fiable et complet. SQLite est le moteur de base de données le plus utilisé au monde. SQLite est intégré à tous les téléphones mobiles et à la plupart des ordinateurs, ainsi qu'à d'innombrables autres applications que les gens utilisent tous les jours. Plus d'informations...

Le format de fichier SQLite est stable, multiplateforme et rétro compatible, et les développeurs s'engagent à ce qu'il en soit ainsi jusqu'en 2050. Les fichiers de base de données SQLite sont couramment utilisés comme conteneurs pour transférer du contenu riche entre les systèmes et comme format d'archivage à long terme des données. Il y a plus de 1 trillion de bases de données SQLite en utilisation active. Le code source de SQLite est dans le domaine public et est libre d'utilisation pour tous.

Figure IV. 18 : Base de donnée Sqlite

IV-7-2 INSTALLATION ET CONFIGURATION

SQLite est également une base de données relationnelle similaire à la base de données SQL qui est utilisée pour stocker les données d'un site web ou d'une application mobile. SQLite est une base de données légère et, contrairement aux autres bases de données, elle n'a pas besoin d'un moteur de serveur de base de données basé sur le client car c'est un système autonome. SQLite est le plus adapté au Raspberry Pi en raison de sa propriété d'être indépendant du serveur.

Avant de procéder à l'installation du Raspberry Pi, il faut d'abord mettre à jour le référentiel du Raspberry Pi en utilisant la commande :

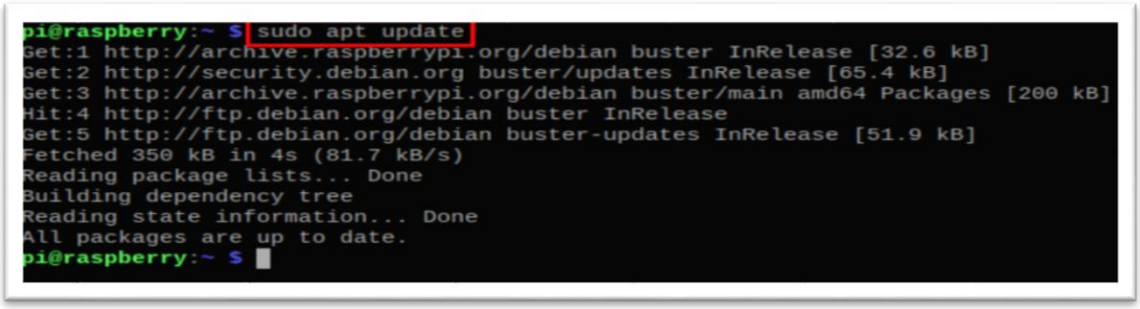


Figure IV. 19 : Mise à jour du système Raspberry Pi

Tous les paquets du dépôt sont à jour, nous allons maintenant installer SQLite en utilisant le gestionnaire de paquets apt :

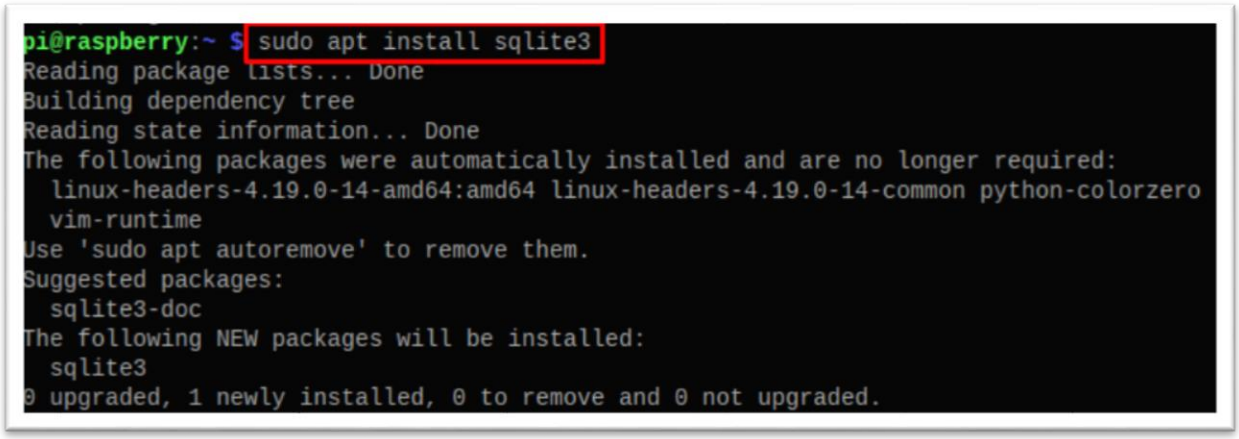


Figure IV. 20 : installation de la base de donnée sqlite

Une fois que le SQLite a été installé, nous pouvons vérifier la version du SQLite installé pour authentifier son installation :

Figure IV. 21 : vérification de la version de sqlite

Nous allons initialiser le serveur SQLite en utilisant la commande et on crée la table de base de donnée avec la commande suivante :

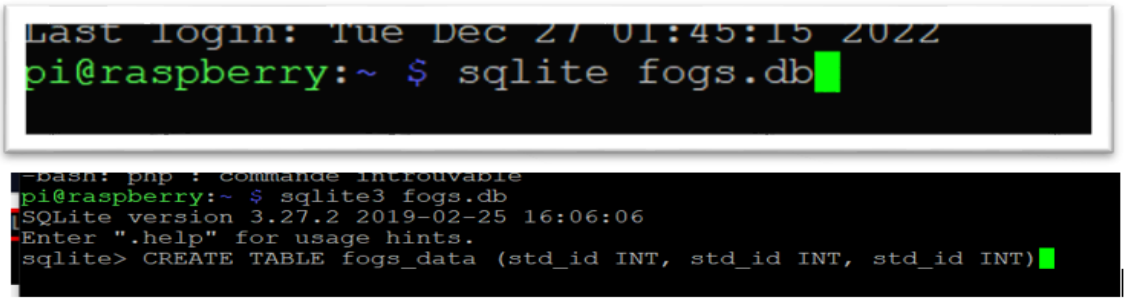


Figure IV. 22 : création de la table de base donnée

IV-7-3 CONFIGURATION DE NOTRE NŒUD DE FOG

Pour configurer notre nœud de Fog et de stocker nos données dans notre nœud de Fog, nous avons utilisés des scripts pour stocker les données du capteur d'humidité du sol qui nous viennent de notre Arduino, ses scripts nous permettent de visualiser le temps de latence que fait les données pour être stocker dans notre nœud de Fog et aussi dans notre Cloud pour enfin les comparées.

Nous avons trois scripts dans notre travail :

Ce script nous permet de stocker tous les données dans le Fog avec les informations de la latence dans la base de donnée sqlite.

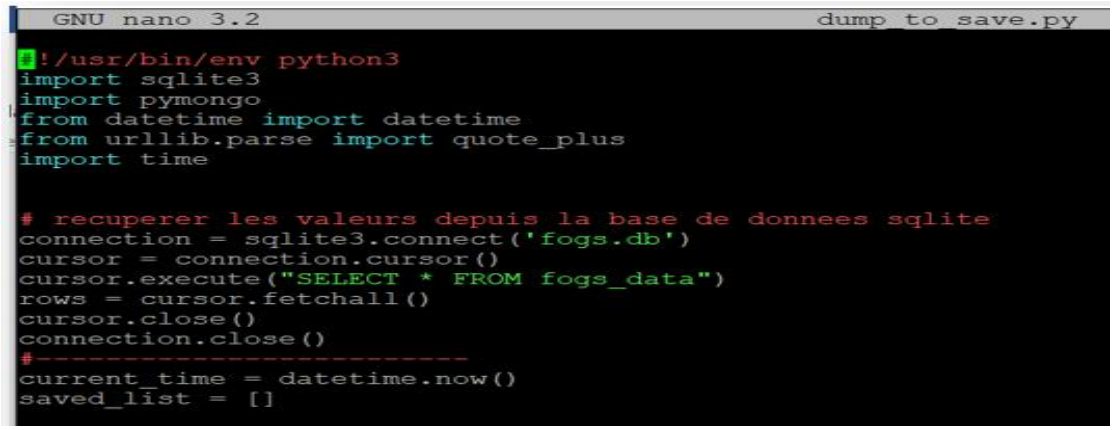


Figure IV. 23 : script pour stocker les données dans le nœud de Fog

Et voilà le script de notre fameux nœud de Fog, qui stocke les données avec leurs latences données dans le Fog.

Figure IV. 24 : script pour le temps de latence sur le Fog

Et enfin nous ce script qui en envoie en même temps les données dans le Cloud, ce script nous permet de savoir le temps de latence des données envoyées dans le Cloud.

```
 GNU nano 3.2                                    fog_cloud.py

#!/usr/bin/env python3
import serial
import pymongo
from datetime import datetime
import time
from urllib.parse import quote_plus


if __name__ == '__main__':
        ser = serial.Serial('/dev/ttyACM0', 1000000, timeout=1)
        ser.reset_input_buffer()

        username = quote_plus('tafa')
        password = quote_plus('evy5@XufWa6yD7e')
        uri = 'mongodb+srv://' + username + ':' + password + "@cluster0.
        client = pymongo.MongoClient(uri)
        db = client.fogs
        collection = db["fogs_data"]
```

Figure IV. 25 : script pour le temps de latence sur le Cloud

IV-7-4 LE CLOUD MONGO DB

Les services en Cloud MongoDB consistent en une suite complète de produits de données qui accélèrent et simplifient la création de données pour toutes les applications. Avec Atlas Database (la base de données en tant que service pour MongoDB), Search et Data Federation, vous pouvez servir n'importe quelle classe de charge de travail via une API commune. En outre, la synchronisation bidirectionnelle entre Atlas et Realm vous permet d'étendre votre back-end en nuage à la périphérie et aux appareils mobiles.

Figure IV. 26 : le cloud MongoDB

IV-7-5 RESULTATS ET COMPARASION

Dans cette section, nous discutons des résultats de notre réalisation pour les latences entre le Fog et le Cloud. Nous avons mené une étude pour évaluer l'efficacité de notre travail sur la réduction de la latence dans les infrastructures Fog.

La figure IV.30 et IV.31 nous montre le temps (en ms) que font les données de notre capteur être sauvegarder dans nos serveurs respectifs, le Fog et le Cloud. Nous avons notre capteur qui envoie des capteurs chaque 30ms dans notre microcontrôleurs Arduino, ce dernier en envoie ces données directement dans notre serveur Fog qui utilise la base de donnée sqlite pour être stocker les données avant de les envoyer dans le Cloud pour une sauvegarde beaucoup plus longue. Et nous observons que le temps de latence lié au Fog est beaucoup plus faible que celui du Cloud.

Les résultats de l'étude sont très positifs, nous avons observé une réduction grave de la latence avec notre Fog. Cette réduction de la latence a entrainé une amélioration significative sur notre travail.

Nous sommes ravis de ces résultats et croyons que notre travail peut être bénéfique pour la communauté des chercheurs. Nous allons maintenant poursuivre nos recherches pour affiner notre travail et étudier son impact sur d'autres aspects, tels que la sécurité des données sauvegardées. Nous sommes impatients de partager nos résultats futurs avec la communauté des chercheurs.

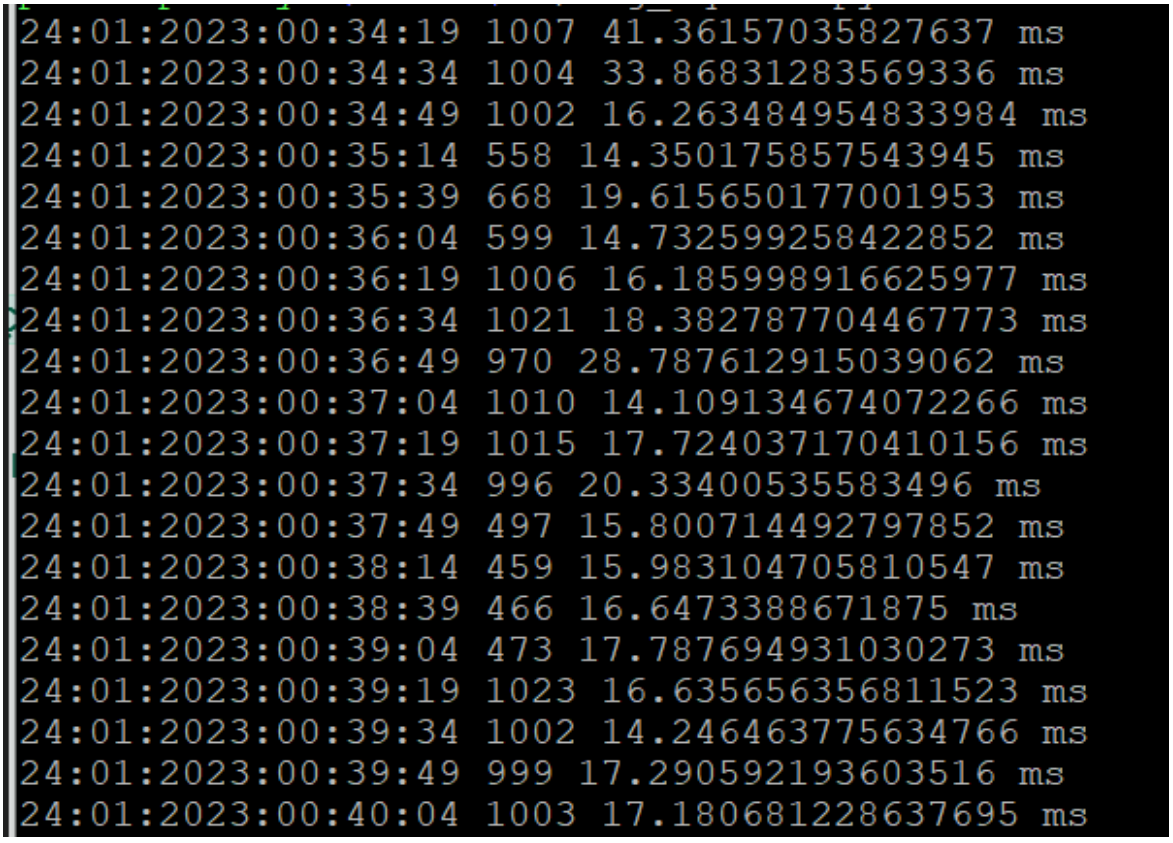


Figure IV. 27 : Résultats des données stockées sur le Fog en fonction du temps

```
23:01:2023:23:57:44 1004 106.77742958068848 ms
23:01:2023:23:57:59 1009 111.25922203063965 ms
23:01:2023:23:58:14 395 102.34403610229492 ms
23:01:2023:23:58:39 390 99.71356391906738 ms
23:01:2023:23:59:04 471 97.0768928527832 ms
23:01:2023:23:59:29 1021 100.74210166931152 ms
23:01:2023:23:59:44 1008 110.88705062866211 ms
23:01:2023:23:59:59 1002 109.21335220336914 ms
24:01:2023:00:00:14 447 128.72672080993652 ms
24:01:2023:00:00:39 492 161.97657585144043 ms
24:01:2023:00:01:04 466 119.32039260864258 ms
24:01:2023:00:01:29 443 98.54269027709961 ms
24:01:2023:00:01:44 1001 126.76334381103516 ms
24:01:2023:00:01:59 1008 449.42283630371094 ms
24:01:2023:00:02:14 1008 128.43632698059082 ms
24:01:2023:00:02:29 1008 113.48652839660645 ms
24:01:2023:00:02:44 1008 156.1446189880371 ms
24:01:2023:00:02:59 1003 122.69449234008789 ms
24:01:2023:00:03:24 498 106.4450740814209 ms
24:01:2023:00:03:49 537 119.37499046325684 ms
```

Figure IV. 28 : Résultats des données stockées sur le Cloud en fonction du temps

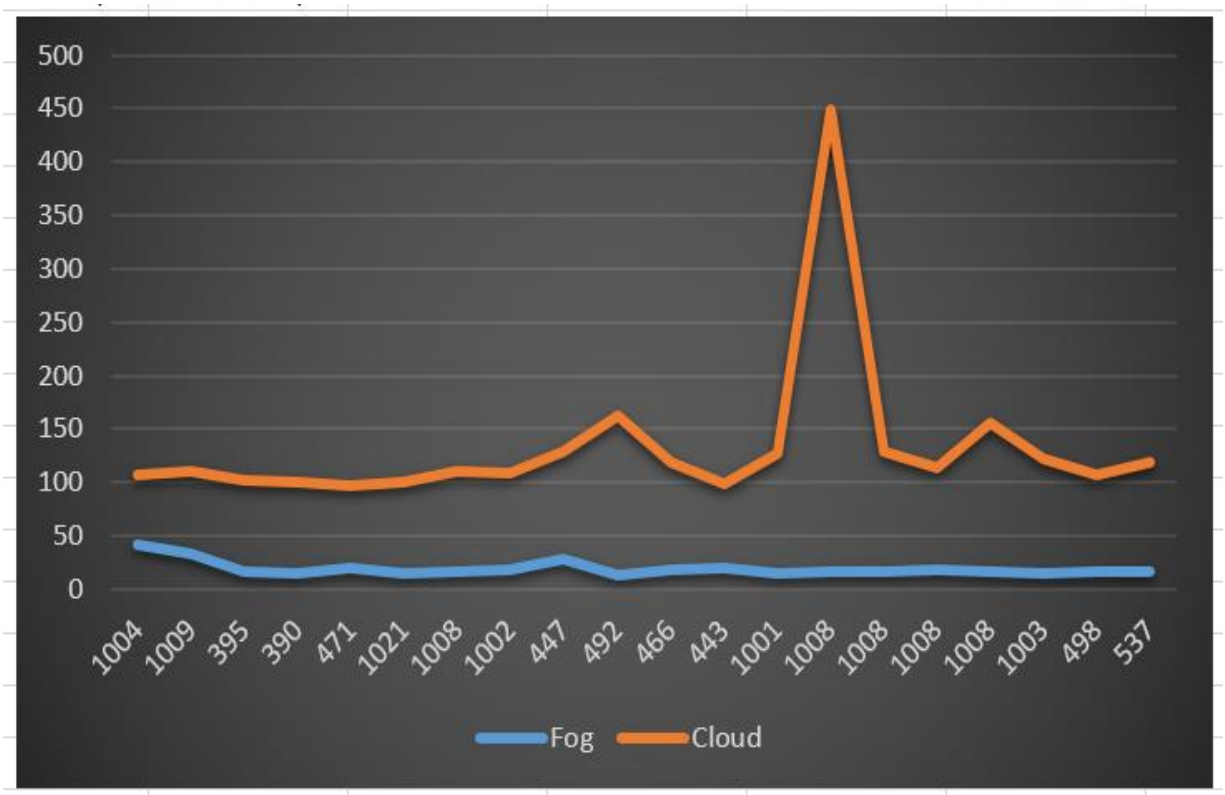


Figure IV. 29 *: Courbes Comparatifs des temps de stockage.*

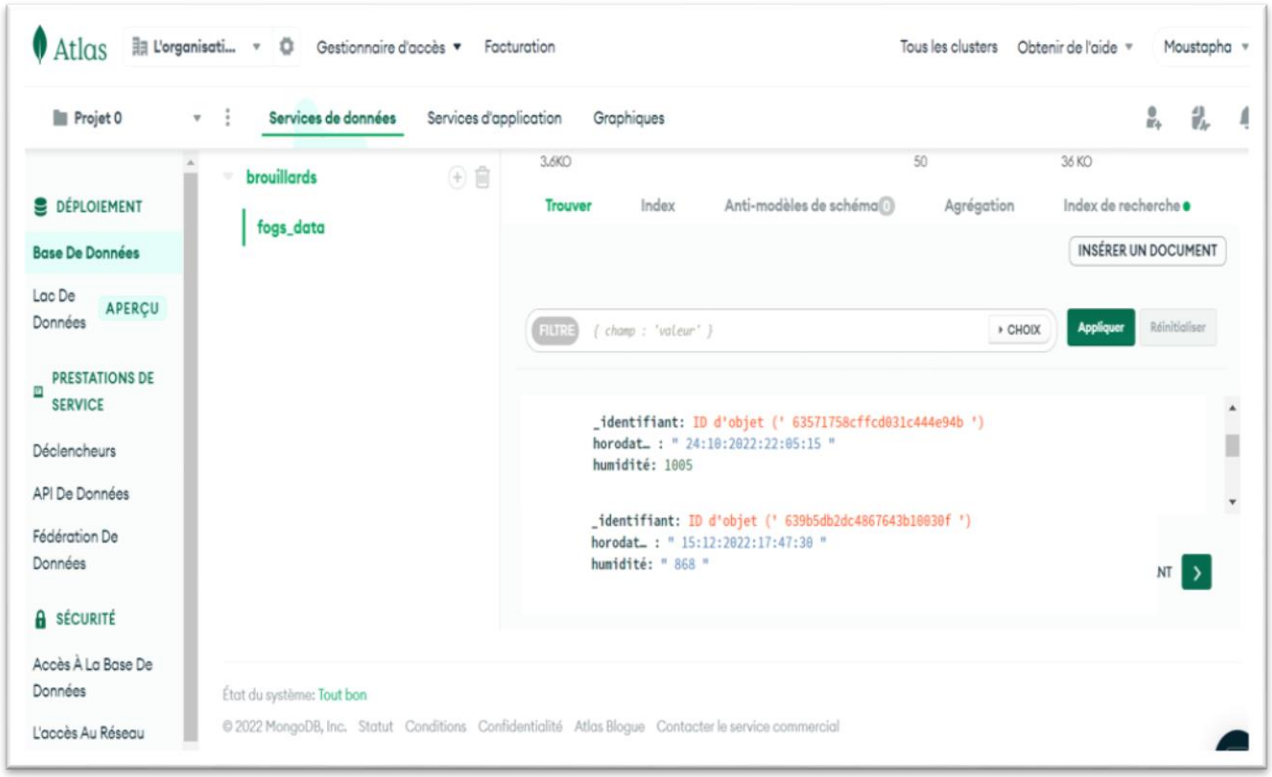


Figure IV. 30: Résultats des données stockées dans le MongoDB

CONCUSION

Dans ce chapitre, nous avons présenté les résultats de notre simulation sur l'ifogsim et de notre réalisation avec notre implémentation sur le Fog Computing. Nous avons montré avec notre simulation de l'arrosage automatique sur ifogsim que le Fog a eu un effet positif sur temps. Nous avons aussi montré comment faire les branchements pour l'arrosage automatique avec l'Arduino Mega 2560. Nous avons également mis en évidence des différences sur le temps de latence entre le Fog et le Cloud en stockant des données.

CONCLUSION GENERALE ET PERSPECTIVES

Dans ce mémoire nous avons porté notre travaille sur le Fog Computing. L'objectif était de montrer le temps de latence (stockage) pour accéder aux données est plus économique avec le Fog qu'avec le Cloud. Le Fog Computing, quant à lui, étend les capacités de calcul et de stockage du cloud vers le bord du réseau, permettant ainsi de traiter les données à proximité des objets connectés et de réduire les temps de latence. Cela permet d'améliorer les performances des systèmes IoT en réduisant le temps nécessaire pour traiter et analyser les données, ce qui peut être crucial dans des applications où la rapidité est essentielle, comme les systèmes de contrôle industriel ou les systèmes de transport intelligents.

Ensemble, ces deux technologies permettent de créer des systèmes de collecte et de traitement de données en temps réel qui peuvent être utilisés dans de nombreux domaines, tels que la santé, les transports, l'agriculture, l'énergie et bien d'autres encore. Ils offrent également de nouvelles possibilités pour améliorer l'efficacité et la résilience des systèmes existants en leur permettant de s'adapter en temps réel aux changements de leur environnement.

Cependant, il convient de souligner que ces technologies comportent également des défis et des risques. Le IoT et le Fog Computing sont basés sur une grande quantité de données qui doivent être collectées, stockées et traitées, ce qui soulève des questions de confidentialité et de sécurité. De plus, l'intégration de ces technologies dans de nouveaux systèmes peut être coûteuse et complexe, et il est important de s'assurer que le bénéfice attendu justifie les coûts et les efforts de déploiement.

En perspectives, nous envisageons dans l'avenir dans nos recherches comment sécurisées les données stockées dans notre nœud de Fog, et d'utiliser des machines puis puissantes pour étudier l'ensemble du Fog sur tous les angles.

En conclusion, le IoT et le Fog Computing sont des technologies clés pour l'avenir de l'informatique et ont le potentiel de changer la manière dont nous vivons et travaillons. Toutefois, il est important de les aborder avec prudence et de tenir compte de leurs défis et de leurs risques potentiels afin de maximiser leurs avantages tout en minimisant leurs impacts négatifs.

REFERENCES

[1] Chikoye, D., Gondwe, T., & Nhamo, N. (Eds.). (2017). Technologies intelligentes pour une agriculture paysanne durable : mise à l'échelle dans les pays en développement. Presse académique.

[2] Nations, U. Growing at a Slower Pace, World Population is Expected to Reach 9.7 Billion in 2050 and Could Peak at Nearly 11 Billion around 2100. Available online : https://population.un.org/wpp/Publications/Files/WPP2019_PressRelease_EN.pdf

[3] Alexandratos, N. ; Bruinsma, J. World Agriculture towards 2030/2050 : The 2012 Revision. 2012. Available online : http : //www.fao.org/3/ap106e/ap106e.pdf

[4] Farooq, MS, Riaz, S., Abid, A., Abid, K. et Naeem, MA (2019). Une enquête sur le rôle de l'IoT dans l'agriculture pour la mise en œuvre de l'agriculture intelligente. Ieee Access , 7 , 156237-156271.

[5] Ayaz, M., Ammad-Uddin, M., Sharif, Z., Mansour, A., & Aggoune, EHM (2019). Agriculture intelligente basée sur l'Internet des objets (IdO) : pour faire parler les champs. Accès IEEE , 7 , 129551-129583.

[6] Pivoto, D., Waquil, PD, Talamini, E., Finocchio, CPS, Dalla Corte, VF, & de Vargas Mores, G. (2018). Développement scientifique de technologies agricoles intelligentes et leur application au Brésil. Traitement de l'information en agriculture, 5 (1), 21-32.

[7] Sethi, P., & Sarangi, SR (2017). Internet des objets : architectures, protocoles et applications. Journal de génie électrique et informatique, 2017.

[8] Yang, Z., Yue, Y., Yang, Y., Peng, Y., Wang, X. et Liu, W. (2011, juillet). Etude et application sur l'architecture et les technologies clés pour l'IOT. En 2011 Conférence internationale sur la technologie multimédia (pp. 747-751). IEEE.

[9] IDEMIA https://www.idemia.com/fr/actualite/cinq-predictions-pour-linternet-des-objets-en-2025-2016-09-28. Visité : 04/09/2022

[10] Turner, V., Gantz, JF, Reinsel, D. et Minton, S. (2014). L'univers numérique des opportunités : des données riches et la valeur croissante de l'Internet des objets. IDC Analyser l'avenir, 16 , 13-19.

[11] Zhang, B., Mor, N., Kolb, J., Chan, DS, Lutz, K., Allman, E., ... & Kubiatowicz, J. (2015, juillet). Le cloud ne suffit pas : économiser l'iot du cloud. Dans HotStorage.

[12] Gupta, H., Vahid Dastjerdi, A., Ghosh, SK, & Buyya, R. (2017). iFogSim : une boîte à outils pour la modélisation et la simulation des techniques de gestion des ressources dans les environnements informatiques de l'Internet des objets, Edge et Fog. Logiciel : pratique et expérience, 47 (9), 1275-1296.

[13] Sarkar, S., Chatterjee, S. et Misra, S. (2015). Évaluation de l'adéquation du fog computing dans le contexte de l'internet des objets. Transactions IEEE sur l'informatique en nuage, 6 (1), 46-59.

[14] Bonomi, F., Milito, R., Zhu, J. et Addepalli, S. (2012, août). Fog computing et son rôle dans l'internet des objets. Dans Actes de la première édition de l'atelier du MCC sur l'informatique en nuage mobile (pp. 13-16).

[15] Byers, CC, & Wetterwald, P. (2015). Fog computing distribuant des données et de l'intelligence pour la résilience et l'échelle nécessaires à l'IdO : l'Internet des objets (symposium sur l'ubiquité). Ubiquité, 2015 (novembre), 1-12.

[16] Byers, CC, & Wetterwald, P. (2015). Fog computing distribuant des données et de l'intelligence pour la résilience et l'échelle nécessaires à l'IdO : l'Internet des objets (symposium sur l'ubiquité). Ubiquité, 2015 (novembre), 1-12.

[17] https://fr.acervolima.com/capteurs-dans-l-internet-des-objets-iot/ visité : 12/09/2022

[18] https://www.robot-maker.com/shop/composants/21-boite-fils-monobrins-21.html visité: 12/09/2022

[19] https://www.robot-maker.com/shop/accessoires-robotiques/43-nappe-40-fils-male-male-43.html visité: 12/09/2022

[20]https://www.techno-science.net/glossaire-definition/Arduino.html visité: 12/09/2022

[21] https://www.fnac.com/Raspberry-Pi-qu-est-ce-que-c-est/cp31961/w-4 visité : 12/09/2022

[22]https://www.clubic.com/raspberry-pi/article-849782-1-raspberry-pi-introduction-nano-ordinateur.html visité : 12/09/2022

[23] https://wikimemoires.net/2019/09/domaines-d-applications-de-l-iot/ visité : 12/09/2022

[24] https://wikimemoires.net/2019/09/domaines-d-applications-de-l-iot/ visité : 13/09/2022

[25] https://fr.digi.com/blog/post/iot-solutions-for-transportation visité : 13/09/2022

[26] https://praedictia.com/page/internet-des-objets/que-peut-faire-linternet-des-objets.html visité : 13/09/2022

[27] https://www.natural-solutions.eu/blog/elevage-connecte visité : 13/09/2022

[28] https://praedictia.com/page/internet-des-objets/que-peut-faire-linternet-des-objets.html visité : 13/09/2022

[29] https://www.mutualia.fr/agriculteur/infos/economie-et-societe/news/internet-des-objets-iot-pour-une-agriculture-plus intelligente#:~:text=Dans%20le%20monde%20agricole%2C%20les,'%C3%A9levage%20%C3%A0%20l'agriculture. Visité : 13/09/2022

[30] MAHI, S., & MEDJAHDI, N. Etude et implémentation des codes LDPC pour la technologie WiMAX IEEE 802.16 (Doctoral dissertation).

[31] https://web.maths.unsw.edu.au/~ lafaye/ CCM/ wireless/wwan.html visité : 15/09/2022

[32] https://www.trendmicro.com/vinfo/us/security/definition/lorawan visité : 13/09/2022

[33] https://www.frugalprototype.com/technologie-lora-reseau-lorawan/ visité : 13/09/2022

[34] https://www.frugalprototype.com/technologie-lora-reseau-lorawan/] visité : 13/09/2022

[35] Hammi, M. T. (2018). *Sécurisation de l'Internet des objets* (Doctoral dissertation, Université Paris-Saclay (ComUE)).

[36] BOUCHENTOUF, H., & BOUDGHENE STAMBOULI, R. (2013). Etude des performances des réseaux 4G (LTE) (Doctoral dissertation).

[37] http://www.marche-public.fr/Terminologie/Entrees/2G.htm visité : 15/09/2022

[38] https://selectra.info/telecom/guides/technologies/reseaux-mobile#9/48.8/2.3 visité: 15/09/2022

[38] https://selectra.info/telecom/guides/technologies/reseaux-mobile#9/48.8/2.3 visité: 15/09/2022

[39] https://web.maths.unsw.edu.au/~lafaye/CCM/wireless/wman.htm visité: 15/09/2022

[40] Roy-Chowdhury, A., Baras, JS, Hadjitheodosiou, M., & Papademetriou, S. (2005). Problèmes de sécurité dans les réseaux hybrides avec une composante satellite. Communications sans fil IEEE, 12 (6), 50-61.

[41]https://eduscol.education.fr/?fbclid=IwAR3AbftSkHjWRtEpCxlSCOuir1nHl2CgppDc2pfJifZcBTFSLUxWHGPqI. Visité : 15/09/2022

[42] Melle Delel Abir. Zigbee alliance organization. Zigbee specification.

[43] Farahani, S. (2011). ZigBee wireless networks and transceivers. newnes.

[44] Vaquero, LM, & Rodero-Merino, L. (2014). Trouver son chemin dans le brouillard : vers une définition complète du calcul du brouillard. ACM SIGCOMM computer communication Review, 44 (5), 27-32.

[45] Iorga, M., Feldman, L., Barton, R., Martin, M., Goren, N. et Mahmoudi, C. (2017). La définition nist de l'informatique de brouillard (N° NIST Special Publication (SP) 800-191 (Draft)). Institut national des normes et de la technologie.

[46] En ligneBar-Magen, J. (2013). Fog computing : introduction à une nouvelle évolution du cloud. Dans Silenced Writings: Landscape as Historiography (pp. 111-126). Éditions Université d'Alcala.

[47] Hu, P., Dhelim, S., Ning, H. et Qiu, T. (2017). Enquête sur le fog computing : architecture, technologies clés, applications et problèmes ouverts. Journal des applications réseaux et informatiques, 98 , 27-42.

[48] Aazam, M., & Huh, EN (2015, mars). E-HAMC : Tirer parti du Fog computing pour le service d'alerte d'urgence. En 2015, conférence internationale ieee sur l'informatique omniprésente et les ateliers de communication (ateliers percom) (pp. 518-523). IEEE.

[49] OpenFog Consortium Architecture Working. OpenFog Architecture Overview. Rapport technique, OpenFog Consortium.

[50] Chen, M., Miao, Y., Hao, Y. et Hwang, K. (2017). Internet des objets à bande étroite. Accès IEEE, 5, 20557-20577.

[51] Atlam, HF, Walters, RJ et Wills, GB (2018). Fog computing et Internet des objets : un bilan. méga données et informatique cognitive , 2 (2), 10.

[52] Mayer, R., Graser, L., Gupta, H., Saurez, E., & Ramachandran, U. (2017, October). Emufog: Extensible and scalable emulation of large-scale fog computing infrastructures. In 2017 IEEE Fog World Congress (FWC) (pp. 1-6). IEEE.

[53] Cisco. Cisco iox router. Https:// community.cisco.com/t5/cisco-iox-documents/ cisco-iox-nodes-isr819-cgr1120-1240-ir829-809-hardware-and/ ta-p/3619076.

[54] Ahmadi, M., RAD, BB, THOMAS, MO et ONYIMBO, BA (2018). Un changement de paradigme technologique : du cloud computing au fog computing. Journal of Engineering Science and Technology, IcCSIt , 216-228.

[55] Yi, S., Hao, Z., Qin, Z. et Li, Q. (2015, novembre). Fog computing : plate-forme et applications. En 2015, troisième atelier IEEE sur des sujets d'actualité dans les systèmes et technologies Web (HotWeb) (pp. 73-78). IEEE.

[56] Computing, F. (2015). The Internet of Things: Extend the Cloud to Where the Things are. Cisco White Paper, 13.

[57] Iorga, M., Feldman, L., Barton, R., Martin, MJ, Goren, N.-É. et Mahmoudi, C. (2018). Modèle conceptuel de calcul de brouillard.

[58] Daneshfar, N., Pappas, N., Polishchuk, V. et Angelakis, V. (2018, décembre). Allocation de service dans une infrastructure de brouillard mobile sous contraintes de disponibilité et de qualité de service. En 2018 IEEE Global Communications Conference (GLOBECOM) (pp. 1-6). IEEE.

[59] https://www.futura-sciences.com/tech/definitions/informatique-cloud-computing-11573/ visité : 24/12/2022

[60]https://www.journaldunet.fr/web-tech/dictionnaire-de-l-iot/1440664-edge-computing-definition-et-cas-d-usage-de-la-technologie/ visité : 24/12/2022

[61]https://kde.mitre.org/blog/2021/03/01/mist-computing-everything-computing-everywhere/ visité : 24/12/2022 visité : 24/12/2022

[62] Dastjerdi, AV, Gupta, H., Calheiros, RN, Ghosh, SK, & Buyya, R. (2016). Fog computing : principes, architectures et applications. Dans Internet des objets (pp. 61-75). Morgan Kaufman.

[63] En ligneSimmhan, Y. (2017). Big data et calcul du brouillard. préimpression arXiv arXiv:1712.09552 .

[64] Adjih, C., Baccelli, E., Fleury, E., Harter, G., Mitton, N., Noel, T., ... & Watteyne, T. (2015, December). FIT IoT-LAB: A large scale open experimental IoT testbed. In 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT) (pp. 459-464). IEEE.

[65] https://inetlab.icube.unistra.fr/index.php/FIT_IoT-LAB visité: 24/12/2022

[66] Le Mouël, F. (2008). Bibliography on Mobile Computing, Adaptive Systems and Distributed Computing. mainly included in PhD Thesis Adaptive Distribution Environment for Applications in a Mobile Context.

[67] Mayer, R., Graser, L., Gupta, H., Saurez, E., & Ramachandran, U. (2017, octobre). Emufog : émulation extensible et évolutive d'infrastructures de calcul de brouillard à grande échelle. En 2017 IEEE Fog World Congress (FWC) (pp. 1-6). IEEE.

[68] Keti, F., & Askar, S. (2015, février). Émulation de réseaux définis par logiciel à l'aide de mininet dans différents environnements de simulation. En 2015, 6e Conférence internationale sur les systèmes intelligents, la modélisation et la simulation (pp. 205-210). IEEE.

[69] https://fogbed.readthedocs.io/en/latest/intro.html visité: 24/12/2022

[70] Sulistio, A., Cibej, U., Venugopal, S., Robic, B. et Buyya, R. (2008). Une boîte à outils pour modéliser et simuler des grilles de données : une extension de GridSim. Concurrence et calcul : pratique et expérience, 20 (13), 1591-1609.

[71] Brogi, A., & Forti, S. (2017). Déploiement d'applications IoT compatible QoS à travers le brouillard. IEEE Internet of Things Journal, 4 (5), 1185-1192.

[72] Lera, I., Guerrero, C., & Juiz, C. (2019). YAFS : un simulateur pour les scénarios IoT dans le calcul du brouillard. Accès IEEE, 7, 91745-91758.

[73] Qayyum, T., Malik, A. W., Khattak, M. A. K., Khalid, O., & Khan, S. U. (2018). FogNetSim++: A toolkit for modeling and simulation of distributed fog environment. IEEE Access, 6, 63570-63583.

[74] Sonmez, C., Ozgovde, A. et Ersoy, C. (2018). Edgecloudsim : un environnement pour l'évaluation des performances des systèmes informatiques de pointe. Transactions sur les technologies de télécommunications émergentes, 29 (11), e3493.

[75] Lopes, MM, Higashino, WA, Capretz, MA et Bittencourt, LF (2017, décembre). Myifogsim : un simulateur de migration de machines virtuelles dans le fog computing. Dans Companion Proceedings of the 10th International Conference on Utility and Cloud Computing (pp. 47-52).

[76] Bangui, H., Rakrak, S., Raghay, S., & Buhnova, B. (2018). Moving to the edge-cloud-of-things: recent advances and future research directions. Electronics, 7(11), 309.

[77] Skarlat, O., Nardelli, M., Schulte, S., & Dustdar, S. (2017, May). Towards qos-aware fog service placement. In 2017 IEEE 1st international conference on Fog and Edge Computing (ICFEC) (pp. 89-96). IEEE.

[78] Calheiros, RN, Ranjan, R., Beloglazov, A., De Rose, CA, & Buyya, R. (2011). CloudSim : une boîte à outils pour la modélisation et la simulation d'environnements de cloud computing et l'évaluation d'algorithmes de provisionnement de ressources. Logiciels : pratique et expérience, 41 (1), 23-50.

[79]https://www.techno-science.net/glossaire-definition/Eclipse-logiciel.html visité: 25/12/2022

[80] Awaisi, KS, Abbas, A., Zareei, M., Khattak, HA, Khan, MUS, Ali, M., ... & Shah, S. (2019). Vers une architecture de parking efficace permise par le brouillard. Accès IEEE, 7 , 159100-159111.