



Master Cloud computing et calcul haute performance

Déchargement de calculs vers un serveur MEC : Minimiser le «temps globale de traitement» d'exécution des tâches

Réalisée Par :

Nouhaila MOUSSAMMI
Houssam CHAKIRI

Encadré Par :

Pr. Mohamed EL GHMARY

8 octobre 2020

REMERCIEMENTS

Nous tenons à remercier toutes les personnes qui ont contribué au succès de notre projet fin d'année et qui nous ont aidées lors de la rédaction de ce rapport.

Avant d'entamer ce rapport nous profitons de l'occasion pour remercier tout d'abord notre professeur Monsieur Mohammed el GHMARY qui n'a pas cessé de nous encourager pendant la durée du projet, ainsi pour sa générosité en matière de formation et d'encadrement. Nous le remercions également pour sa patience , sa disponibilité et surtout ses judicieux conseils qui ont contribué à alimenter notre réflexion.

nous remercions aussi notre cher coordonnateur et professeur à l'université mohamed V monsieur Mostapha ZBAKH de nous avoir incités à travailler en mettant à notre disposition leur expériences et leur compétences.

Nos parents pour leurs soutiens constants et leurs encouragements.

Abstract

On edge computing, mobile devices can offload some computing intensive tasks to the cloud so that the time delay and battery losses can be reduced. Different from cloud computing, an edge computing model is under the constraint of radio transmitting bandwidth, power and etc. With regard to most models in presence, each user is assigned to a single mission, transmitting power or local CPU frequency on mobile terminals is deemed to be a constant. Furthermore, energy consumption has a positive correlation with the above two parameters. In a context of multitask, such values could be increased or reduced according to workload to save energy. Additionally, the existing offloading methods are inappropriate if all the compute densities of multiple tasks are high. In this paper, a single-user multi-task with high computing density model is proposed and partial task is offloaded when use the different offload algorithm. orchestration algorithm is the best method to select offloading tasks, which can enhance the offloading ratio and save energy consumption.

Keywords—edge computing, offloading, orchestration

Résumé

Dans le domaine de l'edge Computing , les appareils mobiles peuvent décharger certaines tâches intensives de calcul vers le cloud afin de réduire le délai et les pertes de batterie. Différent du cloud computing, un modèle de Edge computing est sous la contrainte de la bande passante de transmission radio, de la puissance, etc. En ce qui concerne la plupart des modèles en présence, chaque utilisateur est affecté à une seule mission, la puissance d'émission ou la fréquence CPU locale sur les terminaux mobiles est réputée être une constante. De plus, la consommation d'énergie a une corrélation positive avec les deux paramètres ci-dessus. Dans un contexte de multitâche, ces valeurs pourraient être augmentées ou réduites en fonction de la charge de travail pour économiser de l'énergie. De plus, les méthodes de déchargement existantes sont inappropriées si toutes les densités de calcul de plusieurs tâches sont élevées. Dans cet article, un modèle multi-tâches mono-utilisateur à haute densité de calcul est proposé et la tâche partielle est déchargée lors de l'utilisation de l'algorithme de déchargement différent. L'algorithme de l'orchestration est la meilleure méthode pour sélectionner les tâches de déchargement, ce qui peut améliorer le taux de déchargement et économiser la consommation d'énergie.

Mots clés - edge computing, déchargement, orchestration.

Table des matières

1	Généralité	1
1.1	Introduction	1
1.2	Cloud Computing (CC)	2
1.2.1	Définition	2
1.2.2	Modèles de services	3
1.2.2.1	SaaS (Software as a Service)	3
1.2.2.2	PaaS (Platform as a Service)	3
1.2.2.3	IaaS (Infrastructure as a Service)	4
1.2.3	Modèles de Déploiement	4
1.2.3.1	Cloud Public	4
1.2.3.2	Cloud Privé	4
1.2.3.3	Cloud Hybride	5
1.2.4	Les Avantages et les Inconvénients	5
1.3	Mobile Cloud Computing (MCC)	6
1.3.1	Définition	6
1.3.2	Comment ça fonctionne?	7
1.3.3	les avantages du MCC	7
1.3.4	Pourquoi le cloud mobile?	8
1.3.5	les inconvénients du MCC	9
1.3.5.1	Sécurité, confidentialité et conformité	9
1.3.5.2	Contrôle du flux d'informations	9
1.3.6	Contrôle limité	9
1.3.7	La relation entre CC et MCC	10
1.4	Mobile Edge Computing	10

1.4.1	Définition	10
1.4.2	Les avantages	10
1.4.3	Les inconvénients	11
1.5	Fog Computing	11
1.5.1	Qu'est-ce que le fog computing?	11
1.5.2	L'edge computing VS le fog computing	12
1.5.3	Comment fonctionne le fog computing?	12
1.5.4	Le fog computing peut-il remplacer le cloud?	13
1.6	Réseaux 5G	13
1.6.1	Quels atouts?	13
1.6.2	Quels Services?	14
1.7	Computation Offloading	15
1.7.1	Définition	15
1.8	FDMA : Frequency division multiple access	16
1.9	TDMA : Time division multiple access	17
1.10	OFDMA : Orthogonal Frequency Division Multiple Access	18
1.10.1	les avantages et les inconvénients d'OFDMA	18
1.11	IOT	18
1.11.1	Qu'est ce que c'est IoT?	18
1.11.2	Les composantes de l'IoT	19

2 Problématique 20

2.1	Introduction	20
2.2	Les travaux connexes	21
2.3	Modèle du système	22
2.3.1	Modèle de consommation de temps et d'énergie pour l'exécution locale :	23
2.3.2	Modèles de consommation du temps et d'énergie pour l'exécution sur le serveur MEC :	24
2.4	Formulation du Problème	25
2.5	Résolution du problème	26
2.5.1	Architecture EdgeCloudSim	26
2.6	L'algorithme d'orchestration	28

2.7	simulation numérique	31
2.7.1	Le scénario de simulation	31
2.8	Résultat de la simulation et discussion	33
2.8.1	L'architecture proposée	34
2.8.2	L'algorithme d'orchestration des tâches	35
2.9	Conclusion	36

Table des figures

1.1	Les 3 modèles du Cloud Computing ([46])	3
2.1	Schéma fonctionnel EdgeCloudSim	27

1

1.1 Introduction

Les réseaux de communication mobile ont toujours connu une évolution continue et rapide depuis leur lancement en tant que réseaux téléphoniques uniquement. D'une génération à l'autre, les services se sont multipliés et diversifiés pour inclure les données dans un premier temps puis la vidéo et de nombreux autres services au fur et à mesure.

Ce progrès a été suivi par une croissance exponentielle du nombre d'utilisateurs et des dispositifs mobiles, et par conséquent, du volume de trafic de données. Grâce aux différentes générations de réseaux mobiles, l'architecture a été et est toujours en évolution et de nouveaux mécanismes sont encore proposés pour améliorer la qualité des services et les performances pour l'intégration de nouveaux services.

Dans ce chapitre, nous allons présenter la technologie du Cloud Computing avec ses modèles de services et de déploiement ainsi l'évolution de ces réseaux mobiles vers la 5G, Nous allons ensuite, décrire le déchargement du calculs et en fin on va présenter les composantes de l'internet des Objet

1.2 Cloud Computing (CC)

1.2.1 Définition

Le Cloud computing est un concept de déportation sur des serveurs distants des traitements informatiques traditionnellement localisés sur le poste utilisateur. L'idée est de déporter le traitement sur un serveur externalisé. Plus besoin d'installer le logiciel en local sur chaque poste, les fonctionnalités utiles pour l'entreprise se retrouvent toutes en ligne ou sur un serveur interne. Cette virtualisation des ressources permet à l'entreprise d'accéder à ses données sans avoir gérer une infrastructure informatique (. [42])

Cette technologie, permet aux utilisateurs de réduire leurs coûts d'exploitation des logiciels et du matériel directement en ligne, de gagner en performance et en extensibilité.

Le Cloud Computing permet de rendre une infrastructure matérielle et logicielle dynamique et flexible en exposant les capacités des Data-Centers comme étant un "réseau de services virtuels". Dans cette infrastructure, les utilisateurs peuvent accéder et déployer des applications à partir d'Internet suivant leurs demandes et la qualité de service exigée ([47]).

Le Cloud Computing offre à ses utilisateurs une grande capacité de calcul et de stockage. Les ressources offertes par le cloud sont disponibles sur le réseau et sont accessibles par n'importe quel appareil connecté à travers des mécanismes bien définis. Le Cloud peut servir plusieurs utilisateurs simultanément en affectant et en réattribuant les ressources. Des ressources de calcul, stockage, mémoire et d'autres ressources possibles sont regroupés dans des data centers et ne dépendent pas de l'emplacement de l'utilisateur. En d'autres termes, les utilisateurs du Cloud ne connaissent pas l'emplacement exact de la source des ressources fournies.

1.2.2 Modèles de services

IaaS PaaS et SaaS sont trois modèles de cloud computing populaires sur le marché. Selon les besoins, l'utilisateur peut choisir l'un des trois modèles.

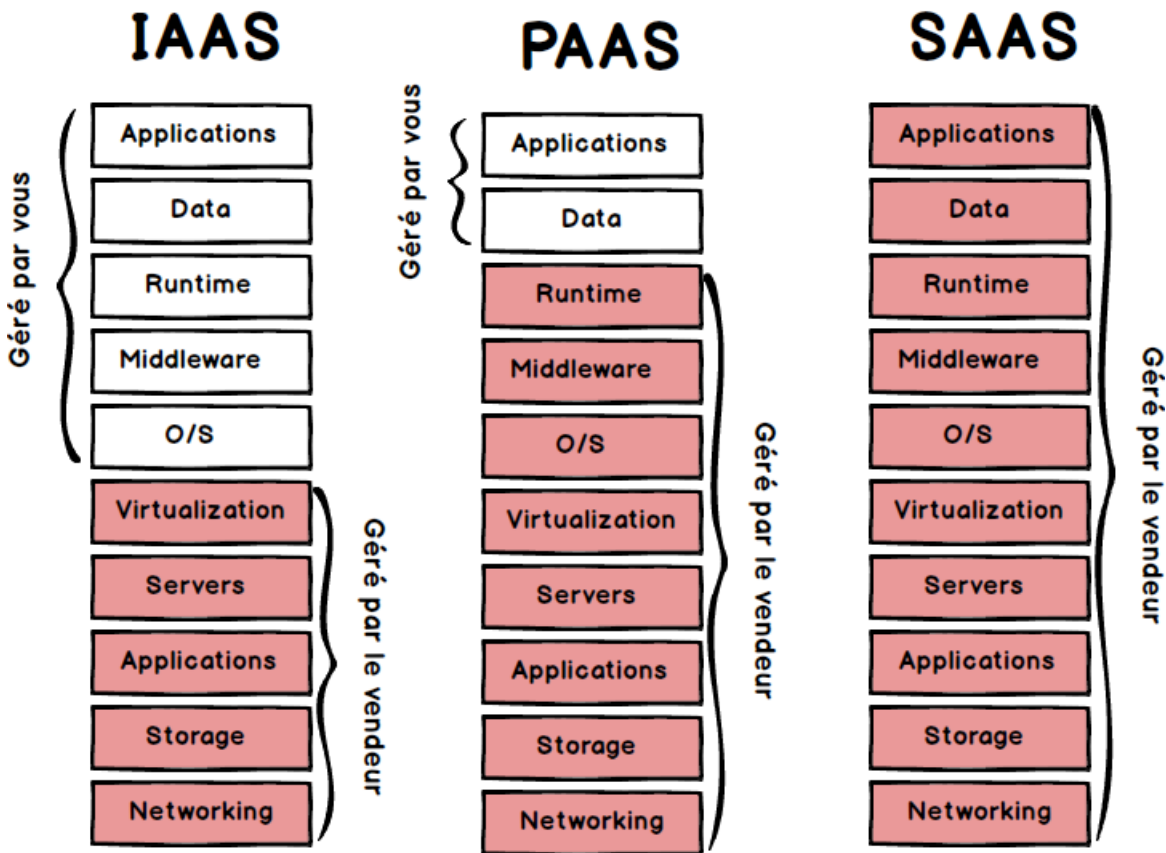


FIGURE 1.1 – Les 3 modèles du Cloud Computing ([46])

1.2.2.1 SaaS (Software as a Service)

Dans un SaaS, vous avez accès à des services d'application installés sur un serveur. Vous n'avez pas à vous soucier de l'installation, de la maintenance ou du codage de ce logiciel. Vous pouvez accéder au logiciel et l'utiliser avec votre navigateur uniquement. Vous n'avez pas à télécharger ni installer aucun type d'installation ou de système d'exploitation, le logiciel est simplement disponible pour que vous puissiez y accéder et le faire fonctionner. La maintenance, l'installation ou l'aide du logiciel sera fournie par le fournisseur de services SaaS et vous ne devrez payer que votre utilisation.([46])

1.2.2.2 PaaS (Platform as a Service)

PaaS ou plate-forme en tant que service vous fournit des plates-formes informatiques qui incluent généralement le système d'exploitation, l'environnement d'exécution du langage de programmation, la base de données et le serveur

Web. techniquement, c'est une couche au-dessus de l'IaaS, la deuxième chose que vous exigez après l'infrastructure est la plate-forme.

Les ordinateurs fournis dans une offre PaaS ont un système d'exploitation et des logiciels fixes. Vous pouvez exécuter votre logiciel en plus de cela.([46])

1.2.2.3 IaaS (Infrastructure as a Service)

IaaS fournit l'infrastructure telle que les machines virtuelles et d'autres ressources telles que la librairie d'images, disque de machines virtuelles, le stockage basé sur les blocs et les fichiers, les pare-feu, les adresses IP, les réseaux locaux virtuels, etc. L'infrastructure en tant que service ou IaaS est la couche de base du modèle en nuage.

De manière plus élaborée, IaaS est analogue à l'achat d'un meuble. On vous donne tous les matériaux, vous devez le construire vous-même. c'est-à-dire que vous recevez un ordinateur sans même un système d'exploitation et que vous pouvez installer le système d'exploitation et tous les logiciels qui s'y trouvent selon votre goût.([46])

1.2.3 Modèles de Déploiement

Nous distinguons deux formes principales de déploiements du Cloud Computing suivant le réseau dans lequel les services sont disponibles : Cloud public et Cloud privé, ainsi d'autre forme dérivée, à savoir Cloud hybride.

1.2.3.1 Cloud Public

Tout d'abord, les services du cloud peuvent être livrés par le biais de clouds publics auxquels l'accès est accordé publiquement pour tous types d'utilisateurs. Cela n'impose pas d'offrir la même qualité de service à tous les utilisateurs. Les fournisseurs peuvent toujours proposer différents plans cloud avec différents coûts en fonction de la quantité de ressources accessibles. Ces clouds ne peuvent être gérés que par les fournisseurs, ou par les institutions (entreprises, gouvernement, universités) qui les déploient dans les locaux des fournisseurs de services cloud. ([39])

Les fournisseurs du Cloud public les plus connus sont Google, Microsoft et Amazon.

1.2.3.2 Cloud Privé

Contrairement aux clouds publics, qui accordent un accès ouvert au grand public, les clouds privés n'autorisent l'accès qu'à un ensemble fermé d'utilisateurs. Les services cloud privés sont exclusivement dédiés à un ensemble spécifique d'utilisateurs définis par le propriétaire du cloud. Ce type de déploiement

est largement utilisé par les organisations professionnelles qui possèdent leurs propres clouds ou utilisent un cloud privé fourni par un fournisseur de services clouds. Les coûts de déploiement de ce modèle sont certes plus élevés mais il est plus sécurisé ([39])

Eucalyptus, OpenNebula et OpenStack sont des exemples de solutions proposées pour la mise en place d'un Cloud privé.

1.2.3.3 Cloud Hybride

Toute combinaison de deux modes d'accès privé, public ou communautaire conduit à un mode d'accès en cloud hybride. Dans les clouds hybrides, deux clouds ayant deux ou plusieurs modes de déploiement sont opérationnellement liés. Ce type de déploiement est parfait pour pouvoir répartir ses moyens en fonctions des avantages recherchés.([39])

1.2.4 Les Avantages et les Inconvénients

Avantages :

mise à jour et évolutivité : pas besoin de mettre à jour l'ensemble des postes pour ajouter de nouvelles fonctionnalités, il suffit de mettre à jour l'application réseau et tous les utilisateurs bénéficient des nouveautés et des corrections. Il en résulte une plus grande cohérence de la méthodologie de travail et des documents produits par l'ensemble des contributeurs de l'organisation.([42])

mise en commun des ressources : chaque utilisateur peut contribuer à l'enrichissement des données et des expériences de l'ensemble si des outils collaboratifs sont mis en place. Cet avantage facilite le knowledge management (gestion et transmission des connaissances) dans les entreprises([42]).

sécurité : si les documents ne sont plus présents en local (et que l'utilisateur ne sauvegarde pas ses identifiants de connexion sur son poste) on évite le problème de l'ordinateur perdu ou piraté et des documents confidentiels perdus dans la nature.([42])

puissance de calcul : le système déporté sur un réseau de serveurs offre une bien meilleure efficacité de calcul qu'un poste seul. Cette fonctionnalité est développée dans les domaines de la compression ou de l'application d'effets vidéo, plus largement dans le partage de vidéo (Youtube etc.) mais aussi dans le jeu en ligne (encore à l'état de test pour le grand public, le goulot d'étranglement étant la bande passante de l'utilisateur). Ce domaine est toutefois réellement intéressant dans le cadre d'application nécessitant une puissance de calcul importante pour une utilisation mobile.([42])

mobilité : l'utilisateur peut à tout moment et à partir de n'importe quel appa-

reil se connecter à ses applications et son workflow. Il peut y accéder à partir de n'importe quel type d'appareil à condition que celui-ci soit doté d'une navigateur

Inconvénients :

sécurité : la plateforme cloud, si elle est externe (non installée sur le réseau interne ou avec une ouverture extérieure) doit être suffisamment sécurisée pour éviter le risque d'intrusion, de vol des données par piratage. L'autre risque est qu'un utilisateur oublie de se déconnecter sur un appareil accessible par des éléments externes à l'organisation. Il faut dans ce cas prévoir une déconnexion automatique en cas de non-activité du compte et bien segmenter les droits utilisateurs afin que ces derniers ne puissent accéder qu'aux données des projets dans lesquels ils sont impliqués. Plus généralement, une clause de confidentialité et la confiance dans son personnel sont primordiales pour que les données ne fuitent pas de manière volontaire.([42])

connexion : c'est l'autre goulot d'étranglement. Si l'utilisateur n'a pas de connexion internet, ou une connexion insuffisante, il ne pourra accéder à sa plateforme de travail. L'idée dans ce cas est de permettre le travail sur un application locale qui synchronise ensuite les données avec le serveur dès que l'utilisateur a à nouveau accès au réseau. Le problème de la sécurité des données en local se pose donc à nouveau([42])

1.3 Mobile Cloud Computing (MCC)

1.3.1 Définition

Mobile Cloud Computing est introduit comme une intégration de Cloud Computing dans l'environnement mobile, pour apporter de nouvelles types de services et d'installations pour les utilisateurs mobiles.

Le terme "Mobile Cloud Computing" (MCC) a été introduit peu après que le concept de "Cloud Computing" a fait son apparition.

"Le Mobile Cloud Computing se réfère à une infrastructure où à la fois le stockage et le traitement des données se produisent à l'extérieur du dispositif mobile. Les applications cloud mobile déplacent la puissance des calculs et de stockage des données loin des dispositifs mobiles vers le cloud".([36])

D'une manière plus précise, le MCC implique un ensemble d'utilisateurs mobiles avec des traitements et des services de stockage des données se produisant dans le cloud. En suivant ce concept, les dispositifs mobiles n'auront pas besoin d'une configuration puissante (par exemple, vitesse du processeur et capacité

de stockage) puisque tous les modules de calculs complexes et intensifs peuvent être traités dans le cloud ([4]).

1.3.2 Comment ça fonctionne ?

Les applications sont exécutées sur un serveur distant. Ils vont être envoyés à l'utilisateur. Grâce à Apple et Google, les navigateurs mobiles ont été améliorés. Les développeurs peuvent contourner les restrictions créées par les systèmes d'exploitation mobiles.

Le traitement et le stockage des données sont déplacés du mobile à des plateformes informatiques puissantes et centralisées situées dans les nuages.

Ils sont accessibles par des appareils mobiles via une connexion sans fil. Connexion à l'aide d'une application ou d'un navigateur Web.

1.3.3 les avantages du MCC

Le cloud computing mobile présente plusieurs avantages parmi lesquels nous citons :

- Allongement de la durée de vie des batteries

La durée de vie de la batterie est l'une des plus grandes préoccupations des utilisateurs mobiles. Comme les téléphones mobiles deviennent des plateformes multi-services et multi-applications générant une quantité toujours croissante de trafic, une plus grande autonomie de ces périphériques est devenue indispensable. Malheureusement, les technologies des batteries ne se développent pas avec le même rythme que la capacité de traitement des téléphones mobiles ([31]). Le MCC, grâce au déchargement de calcul, est l'une des solutions les plus efficaces envisagées actuellement pour résoudre ce problème. Les appareils mobiles pourront décharger le calcul des lourdes tâches au MCC ce qui permettra de dépenser moins d'énergie et d'augmenter en conséquence la durée de vie de leurs batteries.

- Amélioration des performances des appareils mobiles

De nombreuses applications mobiles proposées aujourd'hui sont très gourmandes en termes de calcul ce qui rend problématique leur exécution dans un appareil mobile. En effet, les dispositifs mobiles sont limités en ressources en termes d'énergie, de calcul, de traitement et d'espace de stockage. Grâce au MCC, les périphériques mobiles seront capables d'exécuter des calculs complexes et de traiter de plus grandes quantités de données dans des délais plus courts. De plus, le MCC peut également donner accès aux périphériques mobiles à un espace de stockage quasi illimité. Cela permettra d'économiser l'espace mémoire

local limité de l'appareil mobile. Grâce à une connexion rapide au cloud computing, les utilisateurs mobiles peuvent accéder instantanément à leurs fichiers, photos, documents et vidéos

-Amélioration de la fiabilité

En déléguant l'exécution des applications et le traitement des données au cloud, la sauvegarde et le stockage des données seront fiabilisés. Ainsi, en cas de défaillance du dispositif mobile, les utilisateurs pourront facilement récupérer leurs données dans un autre appareil mobile. Les ressources accordées sont généralement fournies par des serveurs de haut standard qui sont moins exposés à des attaques et plus fiables.

-Evolutivité

MCC fournit aux utilisateurs mobiles un outil puissant pour effectuer le traitement des applications. Ainsi, les appareils mobiles n'ont plus à allouer des ressources ni à planifier les tâches si elles sont déchargées dans le cloud. Cela donne aux appareils mobiles la possibilité d'un approvisionnement dynamique et à la demande de ressources. Les appareils mobiles peuvent donc exécuter leurs applications sans réserve préalable de ressources en raison de l'évolutivité et de la haute disponibilité des ressources cloud.

1.3.4 Pourquoi le cloud mobile ?

Vitesse et souplesse

La création et la révision des applications de cloud mobile peuvent se faire rapidement à l'aide de services cloud. Les applications peuvent être distribuées à de nombreux appareils différents ayant des systèmes d'exploitation différents.

Ressources partagées

Les applis mobiles qui s'exécutent dans le cloud ne sont pas limitées par les ressources de stockage et de traitement des appareils. Les processus dont le fonctionnement exige un grand nombre de données peuvent s'exécuter dans le cloud.

Données intégrées

Le cloud computing mobile permet aux utilisateurs de collecter et d'intégrer rapidement et de manière sécurisée les données de diverses sources où qu'elles se trouvent.

1.3.5 les inconvénients du MCC

Comme cela ressort clairement de ce qui précède, le cloud computing mobile est une technologie qui offre différents avantages à ses utilisateurs. Cependant, il vient avec son ensemble de problèmes et d'inefficacités. Voici les inconvénients les plus importants :

1.3.5.1 Sécurité, confidentialité et conformité

Le problème le plus important est la sécurité en raison de la popularité croissante du cloud computing, la sécurité devient très question importante et critique. La sécurité est l'une des plus grandes préoccupations dans le cloud, en particulier en cas de gestion de données privées et confidentielles comme les informations client ou les informations de carte de crédit. La conformité dans le cloud peut également devenir un problème qui peut nécessiter le déploiement d'un cloud privé si vous devez sécuriser des données privées ([3])

1.3.5.2 Contrôle du flux d'informations

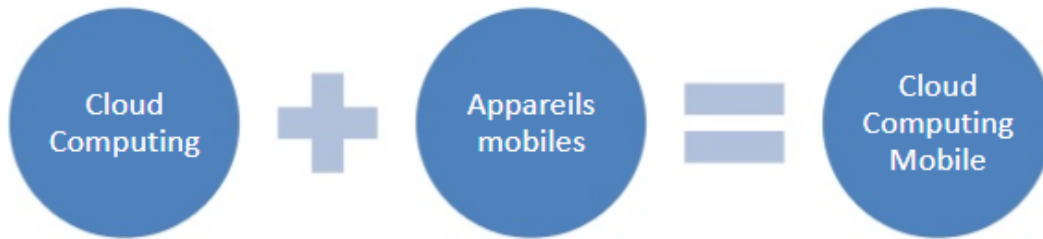
Le contrôle du flux d'informations depuis ou vers le cloud devient également un problème très important, car il est très important de garantir la confidentialité des informations qui circulent du cloud vers les utilisateurs locaux. La sécurisation du flux d'informations lors du calcul d'informations confidentielles dans le cloud est très importante. Lorsque, plus les informations, telles que les données de carte de crédit, les renseignements gouvernementaux ou les informations médicales personnelles traitées dans le cloud, sont sensibles, plus il est important de garantir la confidentialité de ces informations ([3]).

1.3.6 Contrôle limité

Les applications et les services s'exécutent sur des environnements distants donc tiers virtuels ; les utilisateurs et les entreprises ont un contrôle limité sur la fonction du matériel et des logiciels utilisés.

De plus, il manque généralement les fonctionnalités d'une application s'exécutant localement en raison du logiciel distant utilisé ([52]).

1.3.7 La relation entre CC et MCC



==> Le cloud computing mobile utilise le cloud computing pour distribuer des applications à des appareils mobiles. Les applications mobiles peuvent être déployées à distance rapidement, de façon souple et avec des outils de développement. La création et la révision des applications de Cloud mobile peuvent se faire rapidement à l'aide de services cloud. Les applications peuvent être distribuées à de nombreux appareils différents ayant des systèmes d'exploitation, des tâches et des mémoires différents. Les utilisateurs ont ainsi accès à des applications qui ne pourraient pas être prises en charge autrement.

1.4 Mobile Edge Computing

1.4.1 Définition

Le multi-accès Edge Computing (MEC), également connu sous le nom de Mobile Edge Computing, est une architecture de réseau qui permet des capacités de cloud computing et un environnement de service informatique à la périphérie d'un réseau cellulaire.

La technologie MEC est conçue pour être mise en œuvre sur des stations de base cellulaires ou d'autres nœuds périphériques, et permet un déploiement flexible et rapide de nouvelles applications et de nouveaux services pour les clients. MEC est idéal pour être utilisé pour la prochaine génération de réseaux cellulaires 5G ([18]).

1.4.2 Les avantages

Traitement des données en temps réel : dans les architectures d'edge computing, les unités de calcul sont rapprochées au mieux des sources de données,

favorisant une communication en temps réel. On évite ainsi le problème récurrent de latence rencontré avec les solutions de Cloud plus classiques

Débit utile réduit : l'edge computing privilégie un traitement des données en local au niveau de passerelles Edge. Seules les données qui ne peuvent pas être traitées localement, ou qui doivent être mises en lignes, sont téléchargées dans le Cloud.

La sécurité des données : avec une solution d'edge computing, la majeure partie des données reste dans le réseau local. Dans une telle configuration, les entreprises auront plus de facilité à se conformer aux exigences de conformité.([19])

1.4.3 Les inconvénients

Des structures de réseau plus complexes : un système de répartition est bien plus compliqué qu'une architecture Cloud centralisée. Un environnement edge computing est un ensemble hétérogène de plusieurs composants de réseau, venant en partie de divers fabricants, et qui communiquent les uns avec les autres grâce à un grand nombre d'interfaces.

Les frais d'acquisition pour du matériel Edge : les architectures de Cloud se distinguent avant tout par le fait qu'il y a beaucoup moins d'équipement matériel à installer localement. On perd cet avantage si on opte pour des systèmes à répartition.

Un niveau de maintenance plus élevé : un système décentralisé, composé de plusieurs nœuds de calcul, nécessite plus d'entretien et d'administration qu'un centre de données([19]).

1.5 Fog Computing

1.5.1 Qu'est-ce que le fog computing ?

Le fog computing, autrement appelé informatique géodistribuée, informatique en brouillard, ou encore l'infonébulisation, paraît proche de l'Edge. Les deux technologies impliquent de traiter des données plus près de l'endroit où elles sont collectées. Avec le Fog Computing toutefois, les données sont transmises du point de collecte à une passerelle pour être traitées, puis renvoyées à la périphérie. L'informatique en brouillard utilise des solutions Edge et des passerelles avec un réseau local. Le Fog Computing informatique agit comme un pont, réunissant le cloud et la périphérie.([16])Le fog computing est considéré

comme une extension locale du cloud

1.5.2 L'edge computing VS le fog computing

L'edge computing est aujourd'hui considéré comme faisant partie du fog computing, où les ressources informatiques, comme la puissance de calcul et la capacité de stockage sont rapprochées au mieux des équipements IoT, en périphérie du réseau. Dans des architectures de fog computing, le traitement des données se fait d'abord au niveau de la couche fog, tandis que dans des concepts d'edge computing, il est exécuté au niveau de puissants routeurs IoT, et même parfois directement sur les appareils ou sur les capteurs. On peut parfaitement envisager une combinaison des deux concepts. Le graphique ci-dessous montre une telle architecture avec une couche Cloud, fog et edge.([18])

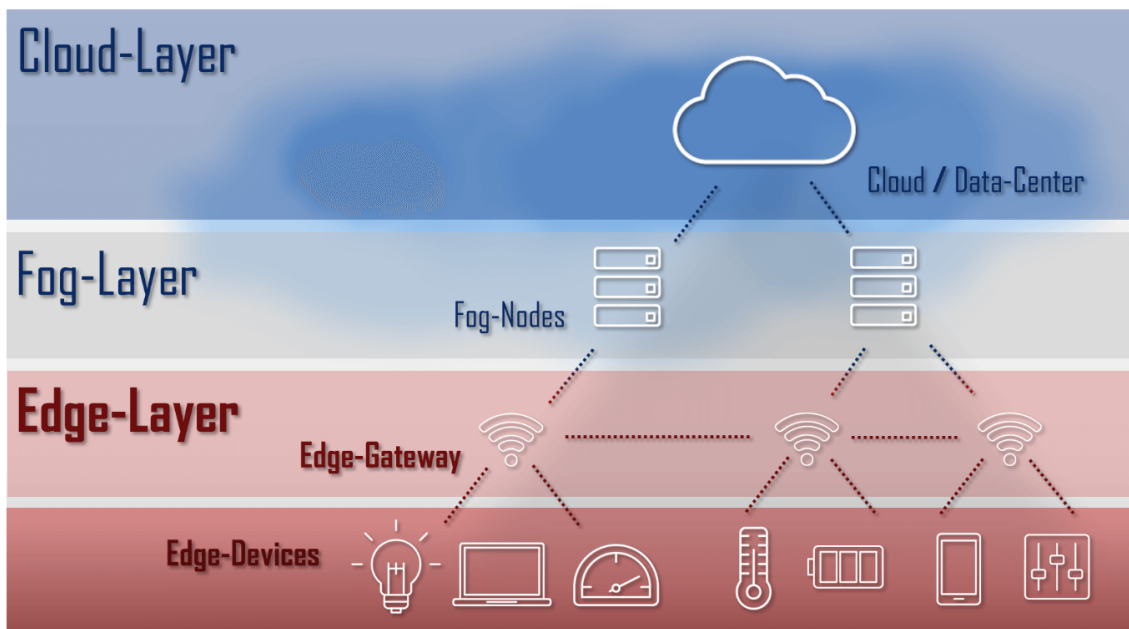


Schéma d'une architecture Cloud avec une couche Cloud, fog et edge : les données provenant d'appareils et de capteurs (répartis dans le monde entier) sont transmises à des passerelles Edge installées sur site, où elles font l'objet d'un premier traitement avant d'être transférées, au besoin, vers des nœuds régionaux au niveau de la couche fog. Ces données sont ensuite renvoyées, si nécessaire, vers un ordinateur central dans le Cloud chargé d'effectuer des analyses plus détaillées ou d'assurer un archivage des données.

1.5.3 Comment fonctionne le fog computing ?

L'OpenFog a défini une infrastructure pour mettre en œuvre une solution de FaaS (Fog as a Service). Celle-ci se décline en plusieurs couches calquées

sur la logique d'empilement du cloud (IaaS, PaaS et SaaS). Une architecture à laquelle s'ajoutent des éléments spécifiques au fog computing : des services d'interconnexion réseau, de collecte de données à partir des objets connectés, ou encore des applications de traitement plus spécifiques. Le principe de fonctionnement est ainsi le même que celui du cloud avec quelques particularités ([20]).

1.5.4 Le fog computing peut-il remplacer le cloud ?

Le fog computing n'est pas destiné à remplacer le cloud. C'est une couche applicative et matérielle complémentaire. Elle permet de répondre à des besoins applicatifs précis liés au traitement en temps réel. Certains cas d'usage, comme les véhicules autonomes ou les compteurs intelligents avec prise de décision immédiate, nécessitent des traitements instantanés qui pourront être pris en charge par cette architecture. D'autres applications sont plus adaptées au cloud qui garde toute sa pertinence dans beaucoup de type de traitements de données, par exemple en matière d'Analytics ([20])

1.6 Réseaux 5G

La 5G est présentée comme la nouvelle génération d'infrastructures de réseaux mobiles qui permettra un saut de performances et la possibilité d'usages très diversifiés. La 5G est, comme son sigle le laisse deviner, la cinquième génération des standards en matière de téléphonie mobile. Elle succédera à la 4G, qui est toujours en cours de déploiement en France en 2019. Pour le dire vite, cette norme apportera des débits plus importants encore, mais aussi un temps de latence bien plus faible qu'aujourd'hui, et pourra supporter énormément de connexions en simultané. Mais la 5G ne doit pas être vue comme une simple évolution de la 4G. C'est en réalité une technologie de rupture. Elle « se distingue des générations précédentes en ce qu'elle vise, dès sa conception, à intégrer un nombre de cas d'usages inédit », relève l'Agence nationale des fréquences. Dès lors, son employabilité promet d'être très étendue et pourra donc servir dans des secteurs variés, notamment industriels([23]).

1.6.1 Quels atouts ?

« Avec la 4G, un film de 800 Mo prend environ 40 secondes à télécharger ; avec la 5G ça serait réduit à une seule seconde », disait en 2014 l'ex-Premier ministre David Cameron. Voilà quel est son premier point fort : la capacité de téléchargement. Les débits en 5G seront jusqu'à 10 fois plus élevés que ceux de la 4G. Si l'on ose un parallèle, la 5G sera une sorte de fibre optique « sans fil » : elle pourrait même atteindre dans certaines situations jusqu'à 20 Gbit/s.

En pratique, il faut plutôt s'attendre à une expérience de navigation entre 100 Mbit/s et quelques Gbit/s.

Autre atout de la 5G : la latence. Cela désigne le délai de transit d'une donnée entre le moment où elle est envoyée et celui où elle est reçue. Celui-ci sera divisé par 10 par rapport à la 4G, avec un temps de réponse d'à peine une milliseconde. Cette réactivité est cruciale pour l'industrie, car des échanges constants et quasi-immédiats sont requis pour faire émerger des usages comme le transport autonome.

Troisième point fort de la 5G : la densité. Avec elle, la 5G supportera « un nombre très important de connexions mobiles simultanées », commente le régulateur des télécoms. Cela va « multiplier par 10 le nombre d'objets connectés au réseau simultanément », confirme l'agence nationale des fréquences. En clair, il s'agit d'éviter l'engorgement des réseaux à l'heure où tout devient connectable et que les capteurs pullulent.([23])

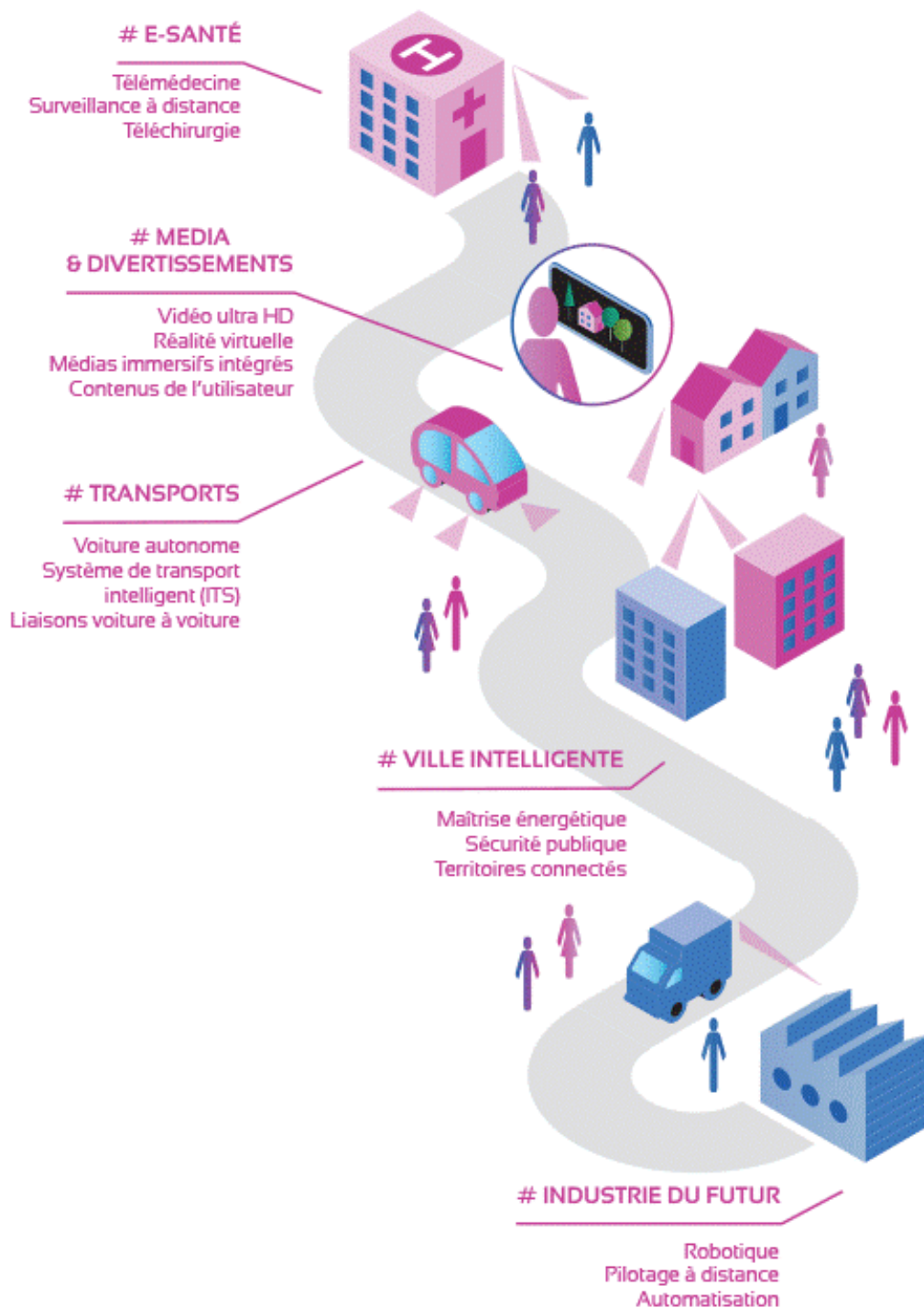
1.6.2 Quels Services ?

Les performances annoncées de la 5G sont telles que les domaines qui vont en profiter sont très nombreux. Pour le mobinaute, la 5G permettra de charger instantanément n'importe quel contenu audiovisuel en haute et en très haute définition (vidéo 4K, vidéo en 3D...) ou de profiter du jeu à la demande (cloud gaming), avec les parties qui sont diffusées directement en streaming entre le joueur et les serveurs du service.]

« La 5G continuera d'améliorer les services existants dans le domaine grand public en donnant par exemple l'accès à des contenus vidéo de meilleures définitions et en favorisant le développement d'applications de réalité augmentée ou virtuelle », anticipe l'Agence nationale des fréquence. Mais c'est surtout du côté de l'industrie que la 5G est intéressante.

Outre les débits accrus qui permettent de transférer plus rapidement des données en masse, la 5G, avec sa très faible latence, ouvre des perspectives dans les véhicules autonomes, l'automatisation industrielle ou le domaine de la santé. « Les sauts de performances permis par la 5G devraient également toucher de nombreux secteurs et permettre à de nouveaux usages d'émerger »([23]).

Quelques domaines qui profiteront de la 5G :



1.7 Computation Offloading

1.7.1 Définition

Le déchargement de calcul est une technique prometteuse qui peut promouvoir la durée de vie et les performances des appareils intelligents en déchargeant

les tâches de calcul locales sur serveurs Edge. le déchargement est l'une des principales caractéristiques du MCC afin d'améliorer la durée de vie de la batterie pour les appareils mobiles et augmenter les performances des applications. toutefois, il existe de nombreuses questions connexes, concernant l'efficacité et la dynamique du déchargement en fonction de changement d'environnement([21]). Le cloud computing mobile est basé sur une opération principale des dispositifs mobiles : le déchargement (l'offloading). Pour utiliser les ressources cloud, les appareils mobiles déchargent leurs tâches de calcul vers le cloud via le réseau mobile.

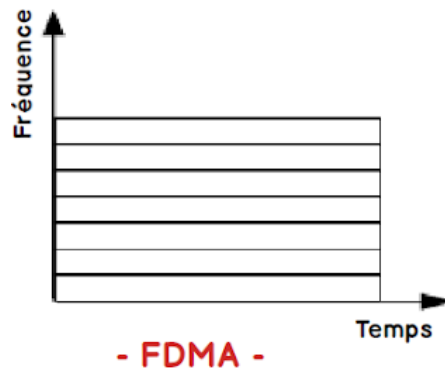
Le principe du computation offloading est de tirer parti d'infrastructures puissantes (par exemple, des serveurs distants) pour augmenter la capacité de calcul des appareils moins puissants (par exemple, les appareils mobiles). Ce paradigme a évolué au cours des 20 dernières années avec l'émergence de diverses technologies informatiques. Le déchargement des calculs est essentiellement un paradigme informatique distribué, dans lequel les appareils mobiles exploitent des télécommandes pour accélérer le calcul et économiser de l'énergie. Pour calculer le débit, les appareils mobiles doivent transmettre des données sur le réseau. Par conséquent, il s'agit d'un compromis entre l'avantage de l'exécution à distance et le coût de la transmission de données.

Computation offloading en informatique de bord fait face à de nombreux défis : le partitionnement des applications, l'allocation des tâches et l'exécution des tâches.

1.8 FDMA : Frequency division multiple access

FDMA, est une technique d'accès multiple par répartition en fréquence. C'est une technologie par laquelle la bande passante totale disponible pour le système est divisée en fréquences. Cette division est faite entre des fréquences non superposées qui sont ensuite assignées à chaque paire communicante (2 téléphones) FDMA est principalement utilisé pour la transmission analogique.

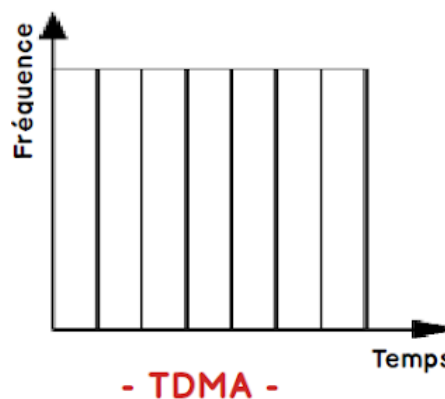
En FDMA, tous les utilisateurs partagent le satellite simultanément mais chaque utilisateur transmet à une fréquence unique. Pour mieux comprendre cette technologie, imaginez simplement comment fonctionne la radio FM. Toutes les radios ont leurs propres bandes de fréquences et envoient leurs signaux aux fréquences uniques soigneusement assignées dans les bandes disponibles([?]).



1.9 TDMA : Time division multiple access

TDMA, est une technique d'accès multiple par répartition dans le temps. À la différence de FDMA et CDMA, dans TDMA la division des appels se produit sur la base du temps. Le système numérise d'abord les appels, puis combine ces conversations en un flux numérique unifié sur un seul canal radio. Maintenant, il divise chaque canal cellulaire en trois tranches de temps, ce qui signifie que trois appels sont mis sur une seule fréquence, puis un intervalle de temps est attribué à chaque appel pendant la conversation, un espace régulier dans un flux numérique. Les utilisateurs émettent successivement l'un après l'autre, chacun utilisant son propre intervalle de temps. Ceci permet à plusieurs stations de partager le même support de transmission (par exemple un canal de fréquence radio) tout en n'utilisant qu'une partie de sa capacité de canal.([15])

Cette technologie permet à trois utilisateurs différents d'utiliser une fréquence en même temps. Ici, il n'y a pas besoin de trois fréquences distinctes comme en FDMA. Dans FDMA, au lieu de monopoliser un seul canal radio pour un seul appel, TDMA achemine efficacement trois appels en même temps. Cette technologie est celle utilisée dans notre système GSM



1.10 OFDMA : Orthogonal Frequency Division Multiple Access

L'accès multiple par répartition en fréquence orthogonale est une fonctionnalité du Wi-Fi 6 qui permet aux points d'accès de desservir plusieurs clients en même temps. L'OFDMA suit un ensemble de règles créées pour la transmission de données entre plusieurs terminaux (tout appareil à l'extrémité d'un canal de transmission, tel qu'un ordinateur ou un téléphone) sur un support de transmission (tel qu'un réseau sans fil).

Un exemple du fonctionnement d'OFDMA est lorsque deux téléphones envoient des données sur la même ligne téléphonique. Un intervalle de temps peut être attribué à chaque téléphone, et ils enverront à tour de rôle leur signal sur la ligne à chaque intervalle attribué. Cependant, ces délais sont imperceptiblement courts, ce qui fait que le transfert de données semble se produire simultanément et de manière transparente.

1.10.1 les avantages et les inconvénients d'OFDMA

La mise en œuvre d'OFDMA peut offrir les avantages suivants :

- * Plus grande diversité et efficacité de fréquence
- * Fournit moins d'interférences entre les cellules.
- * Plus de flexibilité car les canaux et sous-canaux peuvent être activés ou désactivés.
- * Meilleure couverture sur les réseaux.

Cependant, les inconvénients potentiels incluent :

La diversité des fréquences dépend de la façon dont les sous-porteuses sont attribuées aux utilisateurs et peut donc devenir très complexe. Nécessite une puissance supplémentaire car il est toujours allumé et prêt à envoyer une transmission. A une sensibilité plus élevée que les autres types de canaux.

1.11 IOT

1.11.1 Qu'est ce que c'est IoT ?

L'IoT est l'acronyme de Internet Of Things (Internet des Objets en français). Le terme IoT est apparu la première fois en 1999 dans un discours de Kevin ASHTON, un ingénieur britannique. Il servait à désigner un système où les objets physiques sont connectés à Internet. Il s'agit également de systèmes capables de créer et transmettre des données afin de créer de la valeur pour ses

utilisateurs à travers divers services (agrégation, analytique...)([17])

Selon l'UIT (Union Internationale des Télécommunications), l'Internet des Objets est défini comme « une infrastructure mondiale pour la société de l'information, qui permet de disposer de services évolués en interconnectant des objets (physique ou virtuels) grâce aux technologies de l'information et de la communication interopérables existantes ou en évolution »([17]).

Au fil du temps, le terme a évolué et il englobe maintenant tout l'écosystème des objets connectés. Cet écosystème englobe, des fabricants de capteurs, des éditeurs de logiciels, des opérateurs historiques ou nouveaux sur le marché, des intégrateurs... Cet éclectisme en fait sa richesse.

1.11.2 Les composantes de l'IoT

Une solution d'IoT s'articule autour de 5 composants essentiels que sont :

- Les objets (capteurs)
- Le réseau (connectivité)
- Les données
- Les informations
- Les applications d'exploitation.([17])



2.1 Introduction

Avec le développement rapide des services Internet mobiles tels que la réalité augmentée, la réalité virtuelle, les jeux en ligne et la vidéo ultra-claire, ainsi que des applications IoT telles que l'intelligence de groupe, la perception intelligente et la surveillance de l'environnement, le niveau de service des appareils mobiles est devenant de plus en plus exigeant. Ces applications nécessitent un calcul intensif pour prendre en charge l'analyse des données, le traitement vidéo en temps réel et la prise de décision pour optimiser l'expérience utilisateur. Cependant les appareils mobiles intelligents jouent un rôle important dans notre vie quotidienne et ils sont soumis à une puissance de calcul et à une durée de vie de la batterie . Néanmoins, ces appareils souffrent de ressources limitées telles que le processeur, la mémoire et l'énergie. Une technique prometteuse pour résoudre ce problème consiste à transférer des tâches à forte intensité de calcul vers des serveurs edge ce que l'on appelle le déchargement du calcul[27] ou des services nomades [45] , cette technique comprend six étapes :

connaissance de l'environnement de déchargement (découverte du serveur), segmentation des tâches, décision de déchargement, traitement des tâches, exécution à distance des tâches et retour des résultats. Les deux étapes les plus importantes sont la segmentation des tâches et la décision de déchargement. La segmentation des tâches se concentre sur les partitionnements à grains fins tels que MAUI dans [9], ThinkAir dans [26] et Phone2Cloud dans [49], qui est basé sur le code d'application. Il divise les applications en fonction des méthodes, des fonctions et d'autres critères. La décision de déchargement est la question centrale de la technique du déchargement du calcul , Les auteurs de [34][53] ont examiné la décision de déchargement sur le plan énergétique, tandis que l'auteur [11] l'a examiné sur les ressources du serveur et la bande passante.

Le déchargement du calcul a été d'abord appliqué au Mobile Cloud Compu-

ting (MCC)[13][43], qui transfère les tâches des appareils mobiles aux centres de données du cloud . Le principal obstacle au modèle de cloud computing pour le déchargement du calculs est que le retard rencontré en atteignant un serveur cloud (distant) sur un réseau étendu (WAN) peut compenser le temps gagné par le déchargement du calcul. Dans de nombreuses applications mobiles en temps réel telles que les jeux en ligne, la reconnaissance vocale et Facetime, la qualité de l'expérience utilisateur peut être gravement influencée [37][35]. En se concentrant sur cette lacune, les chercheurs ont avancé l'idée qu'un serveur cloud pourrait être rapproché des utilisateurs. Par conséquent, l'Institut européen des normes de télécommunications (ETSI) a proposé le concept de Mobile Edge Computing (MEC), qui combine MEC avec le déploiement de petites stations de base qui peuvent atteindre une proximité physique entre l'appareil mobile et le serveur cloud fournissant le service informatique, réduisant ainsi le délai d'accès aux applications [44][33].

La structure de ce chapitre est la suivante, la section 1 présente une introduction ,la section 2 présente des travaux connexes, La section 3 propose une définition formelle de modèle du système, dans la section 4, la formulation du problème. une résolution du problème est présentée dans la section 5. Pour la section 6, une simulation numérique . Le chapitre est conclu à la section 7.

2.2 Les travaux connexes

De nombreux travaux antérieurs ont proposé de faire évoluer les cadres et les algorithmes pour optimiser différentes mesures de performance ; par exemple, la consommation d'énergie sur l'appareil mobile [25], le débit des applications mobiles de transmission de données en continu [29] et le temps d'achèvement [5][12][32][40][48]. Parmi eux, Un travail dans [51] présentait un algorithme de déchargement qui permettait d'obtenir des décisions de déchargement presque optimales en temps linéaire.

-Dans [2] démontré par des expériences que l'énergie significative peut être économisée par le déchargement du calcul.

-Un auteur dans [14] a développé un algorithme adaptatif de déchargement basé à la fois sur l'historique d'exécution des applications et les conditions actuelles du système.

-Un auteur dans [6] a introduit un schéma de temporisation efficace pour le déchargement du calcul pour augmenter l'efficacité énergétique sur les appareils mobiles.

-Dans [41], un auteur a proposé un cadre de prise de décision basé sur la prédiction pour déterminer quand un calcul en surface surpassera l'exécution locale sur l'appareil mobile.

-Dans [50] un auteur a présenté un efficace de politique de déchargement en

configurant conjointement la fréquence d'horloge dans l'appareil mobile et en programmant la transmission de données pour minimiser la consommation d'énergie.

- Autre auteur dans [10] a proposé un algorithme basé sur l'optimisation de Lyapunov basé sur l'optimisation dynamique pour améliorer les performances du cloud computing mobile tout en respectant le temps d'exécution de l'application. Une approche puissante consiste à transférer des tâches à forte intensité de calcul vers le cloud pour exécution [38][50][24][10][2]

2.3 Modèle du système

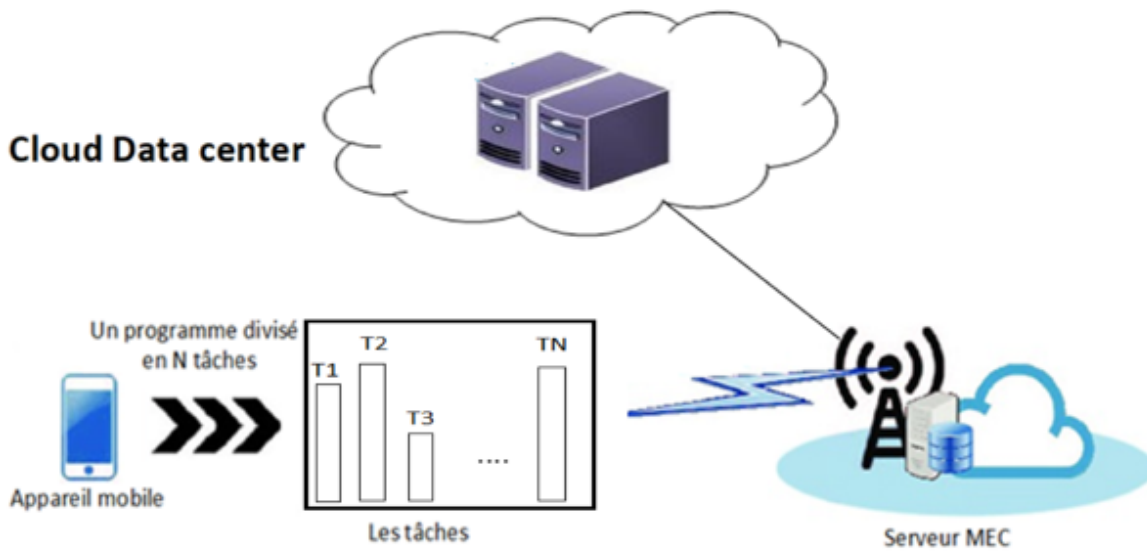


FIGURE 2.1 - Modèle du système d'Edge computing

Le terme data center ou centre de données est une infrastructure peut être utilisée par les entreprises pour organiser, traiter, stocker et entreposer de grandes quantités de données, cependant le terme Edge computing fait allusion au fait que le traitement des données ne se fait plus dans le Cloud, mais il est décentralisé, en serveur Edge, comme le montre la figure 2.1 un Modèle de système d'Edge computing, Dans ce modèle, un morceau de code est divisé en plusieurs tâches mutuellement indépendantes qui sont exécutées localement ou déchargées sur des serveurs Edge. Sur l'appareil mobile et le serveur edge toutes les tâches doivent être terminées dans le temps imparti T^d . L'exécution des tâches locales et celles en Edge sont représentée comme un modèle mathématique qui définit le calcul du temps respectivement la consommation d'énergie, afin de formuler une stratégie d'implémentation de minimisation du temps d'exécution.

Ainsi on va appliquer l'algorithme recuit simulé à ces tâches et on va obtenir comme résultat un sous ensemble des tâches l'une est exécuté localement et l'autre est exécuté sur le serveur . Un appareil mobile était supposé contenir N indépendant tâches, $N=N_0 + N_1$ telque N_0 indique le nombre des tâches qui va s'exécuter localement et N_1 le nombre des tâches qui va s'exécuter sur le serveur, ce qui est désigné par $T \triangleq \{T_1...T_N\}$. Chaque tâche peut être exprimée dans un tuple à deux éléments (D_i, C_i), où D_i (bits) est la taille des données d'entrée, qui se compose des paramètres d'environnement, du code du programme et des paramètres initiaux d'exécution de la tâche, C_i représente les cycles CPU de ces tâches. les valeurs de D_i et C_i dépendent de la nature des tâches et sont obtenus en analysant des situations d'exécution de tâches concrètes [28] [1]. Pour un utilisateur, si la tâche est sélectionnée pour être exécutée localement, la fréquence CPU du l'appareil est désignée par F^{loc} , si elle est sélectionnée pour être exécutée sur le serveur, la fréquence de serveur est désignée par F^o .

Conformément à l'équation de Shannon, la vitesse de transmission (bits / s) peut être exprimée dans la formule suivante comme l'équation (1).

$$R(p) = W \log(1 + kp) \quad (1)$$

Dans l'équation (1), W représente la bande passante en amont, et $k=\frac{h^2}{\sigma}$ est une constante avec h pour un gain de canal déterminé conjointement par la distance d'un utilisateur à une station de base sans fil et le facteur de perte de chemin, p pour le rapport du transmission requis par un utilisateur pour traiter les données d'entrée sur un serveur Edge à proximité, et σ pour le bruit de fond d'une tâche, y compris l'interférence de la transmission d'autres utilisateurs et le bruit de l'environnement naturel.

2.3.1 Modèle de consommation de temps et d'énergie pour l'exécution locale :

Lorsque la tâche i s'exécute localement, le temps requis pour la traiter est :

$$t_i^{loc} = \frac{C_i}{F^{loc}} \quad (2)$$

on considère T^{loc} est le temps total des tâches qui sont exécutées localement N_0 , il peut être exprimé comme :

$$T^{loc} = \sum_{i=1}^{N_0} t_i^{loc} = \sum_{i=1}^{N_0} \frac{C_i}{F^{loc}}$$

L'énergie consommée E_i^{loc} par l'appareil mobile lorsqu'une tâche i s'exécute

localement est :

$$E_i^{loc} = t_i^{loc} * P^{loc} = \frac{C_i}{F^{loc}} * P^{loc}$$

où P^{loc} est la puissance de calcul de l'appareil mobile.

Ainsi la consommation d'énergie total pour les tâches qui sont exécutées localement , il peut être exprimé comme :

$$E_{tot}^{loc} = \sum_{i=1}^{N_0} E_i^{loc} = \sum_{i=1}^{N_0} t_i^{loc} * P^{loc}$$

2.3.2 Modèles de consommation du temps et d'énergie pour l'exécution sur le serveur MEC :

Lorsqu'une tâche i s'exécute sur le serveur MEC, son temps de déchargement du calcul t_i^{off} doit être composé de trois parties, à savoir t_i^{up} est le temps de transmission des données entre l'appareil mobile et le serveur MEC , t_i^{exe} désigne le temps d'exécution des tâches sur le serveur, et t_i^{rep} est temps de renvoyer les résultats du calcul à un utilisateur. Généralement, le temps de retour des résultats de calcul à un utilisateur est ignoré, car la taille des données obtenues par le calcul est bien inférieure à la taille des données d'entrée. Il est conforme aux considérations mentionnées dans [7][30][22] .

Par conséquent, le temps de déchargement du calcul est exprimé comme suit :

$$t_i^{off} = t_i^{up} + t_i^{exe} = \frac{D_i}{R(p)} + \frac{C_i}{F^o} \quad (3)$$

tel que :

$$t_i^{up} = \frac{D_i}{R(p)} \text{ et } t_i^{exe} = \frac{C_i}{F^o}.$$

Par conséquent le temps total de déchargement du calcul peut être exprimée comme :

$$T^{off} = \sum_{i=1}^{N_1} t_i^{off} = \sum_{i=1}^{N_1} (t_i^{up} + t_i^{exe})$$

avec N_1 est le nombre des tâches qui sont exécutées sur le serveur Edge.

De plus , l'énergie consommée E_i^{up} par l'appareil mobile lorsqu'une tâche i s'exécute sur le serveur Edge, est exprimé comme suit :

$$E_i^{up} = t_i^{up} * p = \frac{D_i}{R(p)} * p$$

Ainsi la consommation d'énergie total pour les tâches qui sont exécutées sur le serveur Edge , il peut être exprimé comme :

$$E_{tot}^{up} = \sum_{i=1}^{N_1} E_i^{up} = \sum_{i=1}^{N_1} t_i^{up} * p = \sum_{i=1}^{N_1} \frac{D_i}{R(p)} * p$$

2.4 Formulation du Problème

On pose $x_i \in \{0, 1\}$, si $x_i = 0$ alors la tâche i va s'exécuter localement, et si $x_i = 1$ alors la tâche i va se décharger vers le serveur MEC, dans cette partie, β_1 et β_2 représentent respectivement un ensemble des tâches exécutées localement et l'autre ensemble des tâches implémentées sur le edge. Finalement, le modèle de calcul de cette étude est écrit dans les contraintes suivantes :

$$\begin{aligned}
(P1) \quad & \min_{\beta_1, \beta_2, p, F^{loc}, F^o} \left(\sum_{i \in \beta_1} \frac{C_i}{F^{loc}} + \sum_{i \in \beta_2} \left(\frac{D_i}{R(p)} + \frac{C_i}{F^o} \right) \right) \\
& \text{s.t} \\
(C1) \quad & x_i \in \{0, 1\} \quad \forall i \in [1, N] \\
(C2) \quad & 0 < p < P^{max} \\
(C3) \quad & \sum_{i \in \beta_1} \frac{C_i}{F^{loc}} < T^d \\
(C4) \quad & \sum_{i \in \beta_2} \frac{D_i}{R(p)} + \frac{C_i}{F^o} < \sum_{i \in \beta_2} \frac{C_i}{F^{loc}} \\
(C5) \quad & \sum_{i \in \beta_2} \frac{D_i}{R(p)} * p + \sum_{i \in \beta_1} \frac{C_i}{F^{loc}} * P^{loc} < E^{max}
\end{aligned}$$

La fonction objectif signifie que la puissance de transmission et la fréquence de fonctionnement de la puce sur l'appareil mobile doivent être modifiées et que les tâches appropriées sont sélectionnées sur le Edge pour minimiser le temps d'exécution lorsque le retard pertinent a été limité. Par exemple, nous pourrions utiliser DVFS (Dynamic Voltage and Frequency Scaling) pour ajuster dynamiquement la fréquence du processeur. Bien que la contrainte (C1) indique qu'une tâche doit être exécutée soit localement soit sur un Edge, la contrainte (C2) indique que la puissance de transmission ne doit pas dépasser sa limite supérieure ou être négative, les contraintes (C3) et (C4) soulignent que les tâches traitées localement ou sur le serveur Edge doivent être entièrement implémentées et terminées avant la date limite ainsi que lorsque le temps de déchargement du calcul prend moins de temps que l'exécution locale d'une tâche i le déchargement du calcul peut améliorer la qualité de l'expérience utilisateur. Selon la contrainte (C5) la consommation d'énergie au niveau de l'appareil mobile et au niveau du serveur Edge sont inférieures à une consommation d'énergie maximale E^{max} .

P1 est un problème de programmation non linéaire à nombres entiers mixtes (MINLP : Mixed integer nonlinear programming) pour satisfaire les conditions de retard en modifiant la puissance p d'émission et la fréquence de l'appareil F^{loc} . De plus, une stratégie de déchargement $\beta \in (\beta_1 \cup \beta_2)$ devrait également

être formulé pour minimiser le temps d'exécution, qui a été considérée comme un problème difficile NP. Dans la section suivante...

2.5 Résolution du problème

Pour résoudre le problème P1, on va utiliser un simulateur appelé EdgeCloudSim c'est un outil open source, il fournit une architecture modulaire pour prendre en charge une variété de fonctionnalités cruciales telles que la modélisation de réseau spécifique au WLAN et au WAN, le modèle de mobilité d'appareil et le générateur de charge réaliste et réglable. EdgeCloudSim de par sa conception prend en charge la simulation de scénarios à plusieurs niveaux dans lesquels les serveurs edge s'exécutent en coordination avec les solutions cloud de couche supérieure. Spécifique à ce besoin, EdgeCloudSim fournit un module d'orchestrateur d'edge pour permettre à ses utilisateurs de modéliser les actions d'orchestration qui se produisent généralement dans les scénarios d'edge computing. Pour modéliser les tâches de calcul telles que la création de VM avec une capacité donnée, EdgeCloudSim s'appuie sur les capacités de CloudSim qui dispose d'une base de code connue et fiable depuis longtemps pour la simulation d'actions de calcul. EdgeCloudSim est accessible au public en tant que projet open source. La conception modulaire et la base de code open source d'EdgeCloudSim permettent à ses utilisateurs d'intégrer les besoins spécifiques de leurs scénarios dans leurs expériences de simulation.

2.5.1 Architecture EdgeCloudSim

EdgeCloudSim fournit une architecture modulaire où chaque module aborde un aspect spécifique de l'informatique de périphérie avec des interfaces clairement définies vers les autres modules. Pour faciliter les efforts de prototypage rapide, chaque module contient une implémentation par défaut qui peut être réglée via les paramètres de simulation. Comme expliqué dans les sections précédentes, EdgeCloudSim s'appuie sur les capacités de CloudSim pour modéliser les tâches de calcul. Les machines virtuelles servies par les hôtes sont essentiellement les machines virtuelles standard fournies par CloudSim avec une capacité inférieure et un modèle d'utilisation du processeur plus réaliste. Dans CloudSim, la bande passante, la RAM, le processeur et les ressources de stockage sont limités lors de la création d'une VM. En ce qui concerne les tâches, il n'y a aucune limitation pour le nombre de tâches exécutées sur les machines virtuelles. S'il y a beaucoup de tâches à exécuter, simplement, l'exécution des tâches prend plus de temps. cependant, cette modélisation du temps d'exécution n'est pas compatible avec l'approche Edge Computing, car nous attendons des serveurs

Edge qu'ils gèrent les tâches entrantes dans un court intervalle de temps. Par conséquent, nous implémentons un nouveau modèle d'utilisation du processeur pour les machines virtuelles où le nombre maximum de tâches s'exécutant en parallèle sur les machines virtuelles est limité. L'utilisation du processeur d'une seule tâche est statique et est définie dans le fichier de configuration, mais un modèle d'utilisation dynamique peut également être implémenté en réécrivant la classe d'utilisation du processeur associée. Un schéma fonctionnel détaillé des modules EdgeCloudSim est également illustré dans la figure nous implémentons un nouveau modèle d'utilisation du processeur pour les VM où le nombre maximum de tâches exécutées en parallèle sur les VM est limité. L'utilisation du processeur d'une seule tâche est statique et est définie dans le fichier de configuration, mais un modèle d'utilisation dynamique peut également être implémenté en réécrivant la classe d'utilisation du processeur associée. Un schéma fonctionnel détaillé des modules EdgeCloudSim est également illustré dans la figure nous implémentons un nouveau modèle d'utilisation du processeur pour les VM où le nombre maximum de tâches exécutées en parallèle sur les VM est limité. L'utilisation du processeur d'une seule tâche est statique et est définie dans le fichier de configuration, mais un modèle d'utilisation dynamique peut également être implémenté en réécrivant la classe d'utilisation du processeur associée. Un schéma fonctionnel détaillé des modules EdgeCloudSim est également illustré dans la figure2.1

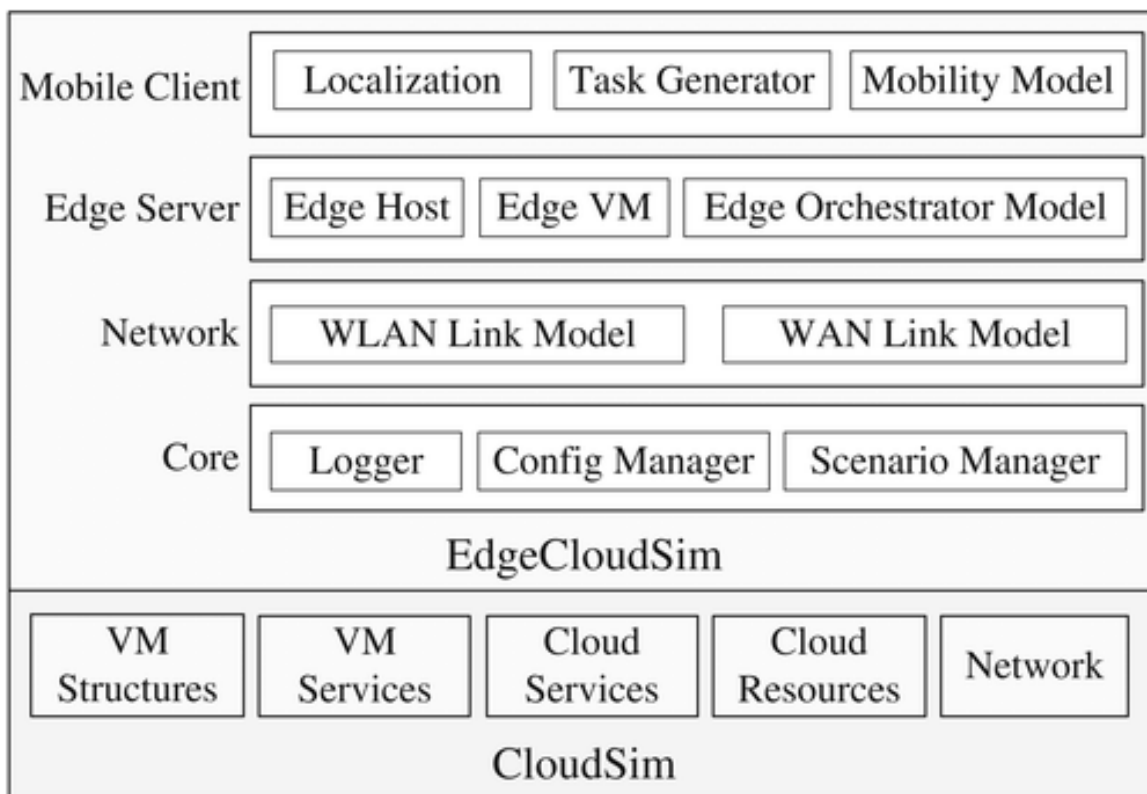


FIGURE 2.1 – Schéma fonctionnel EdgeCloudSim

2.6 L'algorithme d'orchestration

Le réseau IoT est susceptible de changer (par exemple, en raison de la croissance des appareils, de leur mobilité, de leur mort lorsqu'ils sont à court d'énergie, etc.). On s'attend à ce que l'orchestrateur s'adapte en permanence à ces changements afin de respecter la QoS et d'activer les capacités autonomes de l'IoT. Pour garantir cela, nous présentons un algorithme d'orchestration de tâches basé sur l'arbre de décision flou. Il se compose de trois phases principales : la classification, l'apprentissage par renforcement et l'apprentissage par transfert.

Classification des ressources : Il se compose de trois étapes principales (algorithme 1) :

a) La fuzzification : L'arbre de décision permet à l'orchestrateur de classer les ressources en fonction de leurs attributs (les critères) qui sont : la sensibilité à la latence de la tâche, sa taille, la mobilité des dispositifs, la distance à la ressource, son énergie restante, et sa capacité de calcul, qui permet de prendre en charge l'hétérogénéité des dispositifs. Cependant, leurs valeurs sont susceptibles de changer ; ce qui était auparavant considéré comme élevé (par exemple, la capacité de calcul) peut être reconsidéré comme faible lors du traitement d'une nouvelle demande de déchargement et vice versa, en fonction de la tâche, de la ressource qui l'exécute et de l'appareil qui l'a déchargée. Pour prendre en charge de tels changements, au lieu de représenter les ressources par leurs valeurs précises, chaque valeur sera floue et décrite par un terme linguistique tel que «faible», «moyen» et «élevé».

Algorithm 1. Resources classification.

```
1. procedure findDestination(request, inventory)
2.   minCost  $\leftarrow$  -1
3.   selectedResource  $\leftarrow$  null
4.   highestClass  $\leftarrow$  low
5.   for each resource  $\in$  inventory do
6.     fuzzyOutput  $\leftarrow$  Fuzzification(request, resource)
7.     class  $\leftarrow$  decisionTree.classify(fuzzyOutput)
8.     if (class > highestClass) //find the optimal location
9.       highestClass  $\leftarrow$  class
10.      minCost  $\leftarrow$  getCost(getResourceID(resource))
11.      selectedResource  $\leftarrow$  resource
12.    else if (class = highestClass)
13.      newCost  $\leftarrow$  getCost(getResourceID(resource))
14.      if (minCost = -1 or newCost < minCost)
15.        minCost  $\leftarrow$  newCost
16.        selectedResource  $\leftarrow$  resource
17.      end_if
18.    end_if
19.  end_for
20.  offload(request, getResourceID(selectedResource))
21. end
```

b) La classification : La sortie de fuzzification (c'est-à-dire la ressource) sera classée dans l'une des trois classes : «faible», «moyenne» et «élevée», en fonction de la confiance que l'orchestrateur met sur cette ressource (le succès attendu taux), où les ressources classées «faibles» ne sont pas fiables et doivent être évitées.

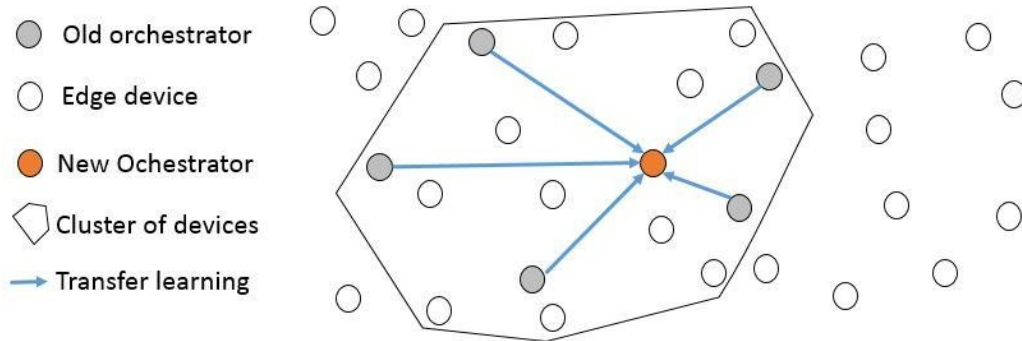
c) Déchargement de la tâche : La classification aboutit à au plus trois classes de ressources. L'orchestrateur choisira alors la destination optimale de la classe avec le plus haut degré de confiance et se décharge de la tâche.

L'apprentissage par renforcement : Le déploiement des orchestrateurs sur les têtes de cluster réduira la latence et l'utilisation du réseau. Cependant, les clusters peuvent évoluer (la sélection de nouveaux orchestrateurs, la mobilité des appareils, etc.), ce qui oblige les orchestrateurs à revoir et à mettre à jour leur politique de décision en réponse à ces changements environnementaux. Pour ce faire, après avoir reçu les résultats de l'exécution, l'appareil récompensera l'orchestrateur si sa tâche a été exécutée avec succès ou le punira autrement. L'orchestrateur mettra alors à jour le niveau de confiance (c'est-à-dire la classe) de la règle de décision correspondante (qui était responsable du déchargement de cette tâche) en fonction du taux de récompense. Il ajoutera également les nouvelles règles de décision qui manquaient si nécessaire (Algorithme 2).

Algorithm 2. Reinforcement learning.

```
1. procedure updateTrust(taskID, status)
2.   rule  $\leftarrow$  getDecisionRuleFromHistory(taskID)
3.   if (status = punishment) incrementPunishment(rule)
4.   else incrementReward(rule)
5.   end_if
6.   if (rewardRate(rule)  $\geq$  0.98 & mobility = low & energy = high) setTrust(rule, high)
7.   else if (rewardRate(rule)  $\geq$  0.96 & mobility = medium & energy = medium) setTrust(rule,
      medium)
8.   else setTrust(rule, low)
9.   end_if
10.  updateOrAdd(rule)
11. end
```

L'apprentissage par transfert : Au fil du temps, en raison de la mobilité et de la consommation d'énergie, le déploiement des orchestrateurs basés sur le cluster peut conduire à la sélection de nouveaux orchestrateurs. Grâce à l'apprentissage par renforcement, ces orchestrateurs peuvent s'adapter à leur environnement. Cependant, cette adaptation prendra un certain temps (quelques minutes), ce qui entraîne l'échec de nombreuses tâches, d'autant plus que les orchestrateurs changent constamment. Par conséquent, le nouvel orchestrateur doit bénéficier des connaissances des proches (y compris les anciens). Bien que ces orchestrateurs exécutent le même algorithme, leurs clusters sont différents, ce qui se traduit par une sorte de forêt aléatoire où chacun d'eux représente une arborescence différente. Cet orchestrateur nouvellement sélectionné demandera et fusionnera ses arborescences pour générer son arborescence initiale. En conséquence, le taux de réussite des tâches sera plus élevé.



L'approche d'apprentissage par transfert.

Algorithm 3. The transfer learning.

```
1. procedure getDecisionTrees()
2.   Forest  $\leftarrow$  RequestDecisionTrees()
3.   DecisionTree  $\leftarrow$  Forest[0] //initialize the decision tree
4.   for i  $\leftarrow$  1 to Length(Forest) do //browse all the trees
5.     for j  $\leftarrow$  0 to Length(Forest[i]) do //browse all rules
6.       if (HasRule(DecisionTree, Forest[i][j]) = false)
7.         addRule(DecisionTree, Forest[i][j])
8.       end_if
9.     end_for
10.  end_for
11.  for i  $\leftarrow$  0 to Length(DecisionTree) do
```

```

12.    high,medium,low ← 0
13.    for k ← 0 to Length(Forest) do
14.        for l ← 0 to Length(Forest[k]) do
15.            if (DecisionTree[i] = Forest[k][l])
16.                if (getTrust(Forest[k][l]) = "high") high ← high + 1
17.                else if (getTrust(Forest[k][l]) = "medium") medium ← medium + 1
18.                else low ← low + 1
19.            end_if
20.        end_for
21.    end_for
22.    if (high > medium & high > low) Trust ← "high"
23.    else if (medium > low) Trust ← "medium"
24.    else Trust ← "low"
25.    end_if
26.    updateTrust(DecisionTree[i],Trust)
27.    save(DecisionTree) //save the obtained tree
28. end_for
29.
30. end

```

2.7 simulation numérique

cette section est responsable du lancement et de la gestion de l'environnement de simulation, de la planification des événements et de la génération des fichiers de sortie. Il contient deux classes importantes, la Gestionnaire de simulation, qui initialise l'environnement de simulation, démarre la simulation et planifie sa fin. Il planifie également la génération des tâches et l'impression des résultats. La deuxième classe est le Enregistreur de simulation. Il est responsable de la génération de la sortie de simulation ; il calcule les résultats, les affiche à la fin de chaque itération et les enregistre au format CSV (valeurs séparées par des virgules) pour les exploiter facilement plus tard.

2.7.1 Le scénario de simulation

Dans ce scénario, les utilisateurs qui possèdent des appareils mobiles (par exemple, des smartphones) déménagent après un laps de temps aléatoire. il doit comprendre également d'autres appareils : des capteurs simples (par exemple, des caméras de vidéosurveillance, des appareils portables, etc.), des appareils non mobiles (par exemple, des ordinateurs portables et des serveurs). Chaque utilisateur utilisera un type d'application spécifique avec des exigences spécifiques. Pour prolonger leur durée de vie, les appareils alimentés par batterie devraient, si possible, décharger leurs tâches sur d'autres appareils, tandis que les appareils avec des tâches intensives en calcul devraient les décharger pour minimiser le temps d'exécution.

Les types d'appareils suivants seront utilisés (Tableau 1), où le quatrième type se réfère à des capteurs simples qui n'ont pas une capacité de calcul suf-

fisante (ne génèrent que des données / tâches). Les valeurs CPU sont définies selon les benchmarks donnés dans [8]. Cependant, les valeurs de consommation électrique sont approximatives, calculées à l'aide d'un vrai smartphone. Dans ce cas, ces valeurs ne sont utilisées que pour comparer les algorithmes et n'affectent pas beaucoup les résultats de la simulation.

Types d'appareils Edge	ordinateurs Portable	Téléphone intelligent	Serveur	Capteur
Mobilité	Non	Oui	Non	Non
batterie-Alimenté	Oui	Oui	Non	Non
Générer des tâches	Non	Oui	Non	Oui
Pourcentage d'appareils (%)	20	40	dix	30
Capacité de la batterie (Wh)	15,75	20	-	-
Taux de consommation d'énergie en veille (Wh / s)	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
consommation d'énergie maximal (Wh / s)	11×10^{-4}	11×10^{-4}	11×10^{-4}	11×10^{-4}
CPU (GIPS)	130	70	200	-
RAM (Gbyte)	8	4	32	-
Stockage (Gbyte)	1024	64	2048	-

Tableau 1. Les types d'appareils.

On suppose que gain de canal $h = 10^{-10}$, bruit $\sigma = 10^{-9}$ et la puissance de transmission maximale $P_{max} = 0.2W$. De plus, la ressource CPU F^0 sur le cloud est de 5 GHz. Dans ce scénario, trois types d'applications sont utilisés (Tableau 2) avec des exigences de latence différentes : (i) Une application de reconnaissance faciale requise par le système de vidéosurveillance. Cette application est sensible à la latence et nécessite une capacité de calcul importante. Les données téléchargées dans ce cas sont une image dont la taille est généralement d'environ 1 Moctets. (ii) Ce scénario peut également inclure d'autres tâches informatiques lourdes, telles que la reconnaissance d'objets et le traitement vidéo, qui nécessitent une puissance de calcul supplémentaire. Enfin, (iii) les applications sensibles à la latence telles que les systèmes de surveillance à distance et l'e-santé. Dans ce cas, les données générées par les capteurs ne dépassent pas quelques kilo-octets et leur traitement ne nécessite pas une grande puissance de traitement.

Types d'applications	Visage reconnaissance	Calcul-intensif	Latence-sensible
Sensibilité de latence	Oui	Non	Oui
Nombre de cœurs de processeur	1	1	1
Longueur de la tâche Instructions Giga (GI)	60	150	30
Taille de la tâche (Ko)	1500	10 000	100
Taille des résultats (Mo)	100	10 000	100

Tableau 2. Les types d'applications.

Lors du déchargement de la tâche, l'orchestrateur choisira la destination de déchargement à l'aide de l'un des algorithmes d'orchestration suivants :

1. L'algorithme de pointe qui minimise la consommation d'énergie et les délais [54].
 2. L'algorithme proposé basé sur l'arbre de décision flou.
- Les paramètres de simulation pour ce scénario sont repris dans le tableau 3.

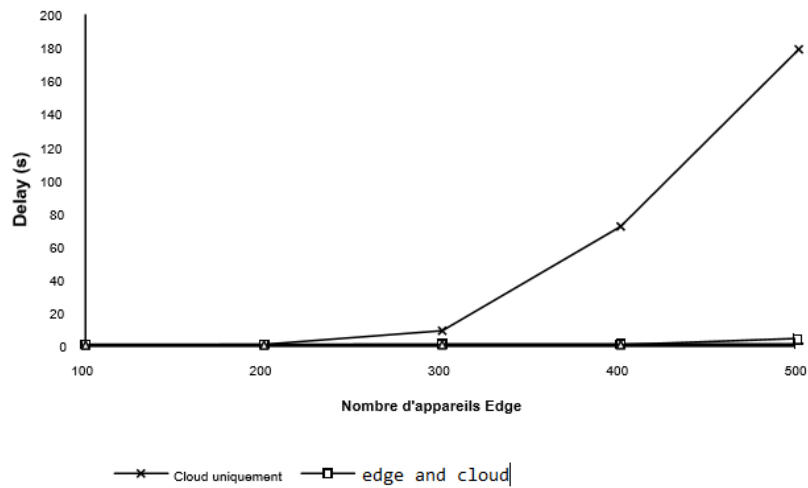
Paramètre	Valeur
Durée de la simulation	60 (min)
Intervalle entre les événements	0,01 (s)
Nombre minimal d'appareils Edge	100
Nombre maximal d'appareils Edge	500
Taille du pas de compteur d'appareils Edge	100
Vitesse des appareils mobiles	1,4 (m / s)
Gamme d'appareils Edge	10 (mètres)
Taille de la zone de simulation	200 x 200 (mètres)
Taux de génération des tâches	3 (Tâches / appareil / minute)
Intervalle de mise à jour du réseau	0,1 (s)
Bande passante WLAN	300 (Mbits / s)
Bande passante WAN	300 (Mbits / s)
Délai de propagation WAN	0,2 (s)
Déploiement d'orchestrateurs	Basé sur les clusters.
Algorithme d'orchestration	Proposé, à la pointe de la technologie.
Architectures	Edge.

Tableau 3. Les paramètres de simulation

2.8 Résultat de la simulation et discussion

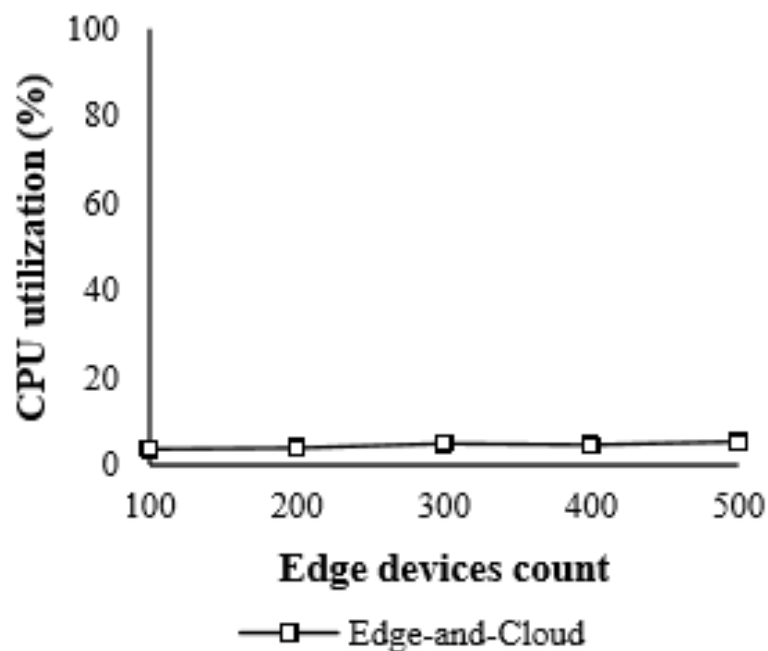
Dans cette section, la plateforme proposée sera évaluée. Au cours de cette évaluation, nous nous concentrerons sur des mesures importantes qui déterminent la qualité de service et reflètent l'évolutivité de la solution proposée, telles que le taux d'échec des tâches, les retards, le réseau et l'utilisation du processeur. Tout d'abord, l'architecture proposée sera évaluée. Ensuite, la stratégie d'orchestration des tâches proposée. Après cela, la solution de déploiement d'application proposée, et enfin, l'algorithme d'orchestration des tâches proposé.

2.8.1 L'architecture proposée



Délai moyen d'achèvement des tâche

Lorsque le nombre d'appareils Edge augmente, le délai moyen d'achèvement des tâches augmente, en particulier lors de l'utilisation d'architecture Cloud uniquement où toutes les tâches sont déchargées vers le Cloud distant. Cependant, il passe toujours d'une moyenne de 0,5 seconde (100 appareils) à 4,5 secondes (500 appareils). En revanche, le délai d'achèvement des tâches est resté presque stable lors de l'utilisation du Edge et Cloud architecture. Malgré leur faible capacité de calcul, l'utilisation des appareils Edge a réussi à réduire le délai moyen quel que soit le nombre d'appareils avec une moyenne de 0,9 seconde (100 appareils) à 1,1 seconde (à 500 appareils).

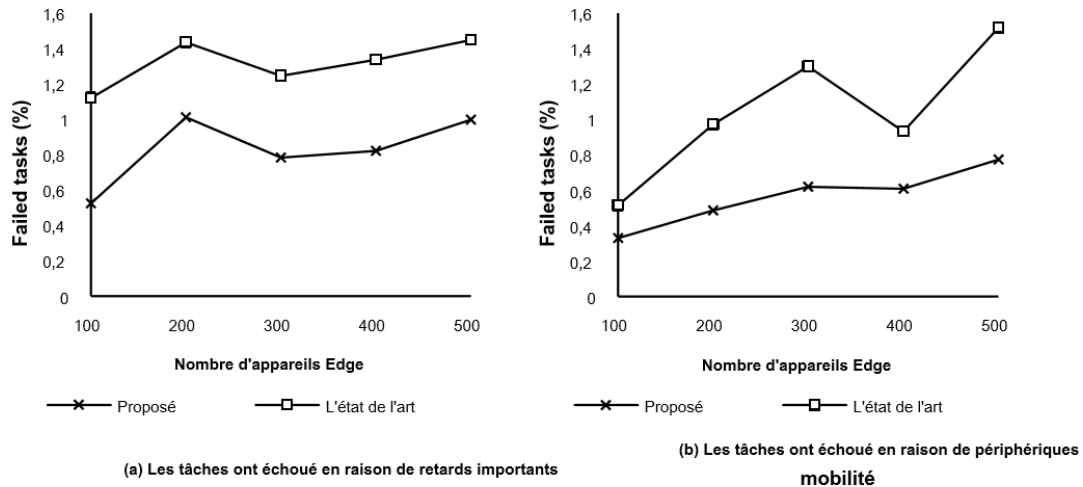


Utilisation moyenne du processeur CPU Edge

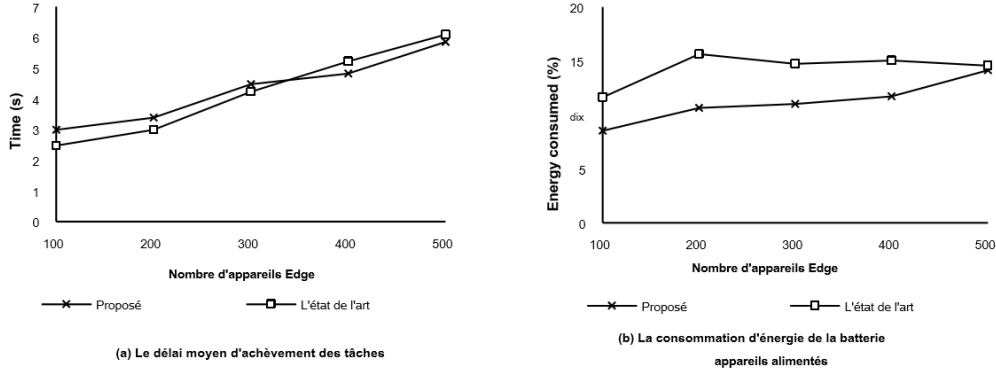
La croissance des appareils dans ce cas signifie la disponibilité de plus de ressources. Ainsi, la charge de travail reste stable quel que soit le nombre de d'appareils, ce qui est confirmé par l'image au dessus, où l'utilisation moyenne du processeur d'appareils d'edge est restée stable autour de 5%.

2.8.2 L'algorithme d'orchestration des tâches

Les taux des tâches qui ont échoué en raison de la mobilité et des retards élevés sont illustrés dans la Figure 16. L'échec d'une tâche due à la mobilité se produit lorsqu'un appareil (soit celui qui a déchargé la tâche, soit celui qui l'exécute) quitte son cluster avant de terminer la tâche déchargée. Contrairement à l'algorithme de pointe, l'algorithme d'orchestration proposé prend en compte l'hétérogénéité des appareils et les exigences de leurs applications. Par conséquent, si une tâche n'est pas sensible à la latence, elle sera souvent déchargée vers le Cloud si disponible (c'est la raison pour laquelle l'utilisation du Cloud était relativement élevée, bien que l'utilisation d'Edge soit faible, dans la figure 16). Sinon, il sera déchargé sur un autre appareil Edge. Cependant, les appareils mobiles et alimentés par batterie seront évités autant que possible.



De même, l'image suivante représente le délai moyen d'achèvement des tâches et la consommation d'énergie des appareils alimentés par batterie. De plus, en évitant les appareils mobiles, l'algorithme proposé évite autant que possible les appareils alimentés par batterie. En conséquence, la consommation d'énergie moyenne de ces appareils a diminué de 61% par rapport à l'algorithme de pointe. D'un autre côté, même s'il s'appuie davantage sur le Cloud lorsqu'il s'agit de tâches non sensibles à la latence, l'algorithme proposé a réussi à conserver le même délai que celui de l'état de l'art.



2.9 Conclusion

Le modèle à tâches multiples à l'utilisateur présenté dans ce rapport a été adopté pour attribuer les tâches à décharger en vertu d'un algorithme de d'orchestration. L'objectif de réduire le temps d'exécution et la consommation totale d'énergie. Comme démontré par l'expérience d'émulation basée sur la simulation numérique, dans le cas de tâches multiples ont des densités de calcul élevées qui les décharger en fonction des densités de calcul peuvent entraîner l'échec du déchargement de celles de grande taille mais une lourde charge de calcul; en outre, continuer à sélectionner les tâches qui semblent être les plus économes en énergie à décharger dans le cloud peut éventuellement conduire à un faible taux de déchargement et tomber dans la solution optimale locale car une telle opération a occupé trop Puissance d'émission. cependant, le L'algorithme d'orchestration proposé dans ce rapport est capable de décharger des tâches consommant beaucoup d'énergie exécutées localement et d'améliorer le taux de déchargement simultanément pour finalement réduire la consommation totale d'énergie.

2

Bibliographie

- [1] A. p. miettinen and j. k. nurminen, “energy efficiency of mobile clients in cloud computing,” in *proc. unix conf. hot topics cloud comput. (hot-cloud)*, june 2010.
- [2] A. rudenko, p. reiher, g. j. popek, and g. h. kuenning, “saving portable computer battery power through remote process execution,” *acm sigmobile mobile computing and communications review*, vol. 2, no. 1, pp. 19–26, 1998.
- [3] Aepona 2010, “mobile cloud computing solution brief,” aepona, white paper (november. 2010).
- [4] Amit k sharma and priyanka soni. mobile cloud computing (mcc) : Open research issues. *international journal of innovations in engineering and technology (ijiet)*, pages 24–27, 2013.
- [5] B. g. chun, p. maniatis. “clonecloud : Elastic execution between mobile device and cloud,” - *proc. eurosys’ 2010*.
- [6] C. xian, y. lu, and z. li, “adaptive computation offloading for energy conservation on battery-powered systems,” in *international conference on parallel and distributed systems*, vol. 2. *ieee*, 2007, pp. 1–8.
- [7] Cheng, k .; teng, y .; sun, w .; liu, a .; wang, x. stratégie conjointe de répartition des ressources sans fil et éconergétique dans les systèmes de serveurs multi-mec. dans les actes de l’*ieee icc*, kansas city, mo, usa, 20-24 mai 2018.
- [8] Cpu-world. test : Sandra dhrystone (mips). [http ://www.cpu-world.com/benchmarks/ browse/103596,116104,76063,76061,263806,96563,96516](http://www.cpu-world.com/benchmarks/browse/103596,116104,76063,76061,263806,96563,96516).
- [9] Cuervo, e .; balasubramanian, a .; cho, d .; wolman, a .; saroiu, s .; chandra, r. maui : Making smartphones last longer with code offload. in *proceedings of the 8th international conference on mobile systems, applications, and services*, san francisco, ca, usa, 15–18 june 2010.

- [10] D. huang, p. wang, and d. niyato, "a dynamic offloading algorithm for mobile computing," *ieee transactions on wireless communications*, vol. 11, no. 6, pp. 1991–1995, 2012.
- [11] Deng, m. ; tian, h. ; lyu, x. adaptive sequential offloading game for multi-cell mobile edge computing. in *proceed ings of the 2016 23 rd international conference on telecommunications(ict),thessaloniki,greece, 16–18 may 2016 ; pp. 1–5.*
- [12] E. cuervo, a. balasubramanian, d cho. "maui : Making smartphones last longer with code offload," - *mobisys' 2010.*
- [13] Fernando, n. ; loke, s.w. ; rahayu, w. mobile cloud computing : A survey. *future gener. comput. syst.* 2013, 29, 84–106.
- [14] G. huertacanepa and d. lee, "an adaptable application offloading scheme based on application behavior," in *22nd international conference on advanced information networking and applicationsworkshops*, 2008.
- [15] "[https ://waytolearnx.com/2018/07/difference-entre-fdma-tdma-et-cdma.html](https://waytolearnx.com/2018/07/difference-entre-fdma-tdma-et-cdma.html) " created juillet 23, 2018.
- [16] "[https ://www.dcloudnews.eu/stratus-veut-fiabiliser-linternet-industriel-des-objets/](https://www.dcloudnews.eu/stratus-veut-fiabiliser-linternet-industriel-des-objets/).
- [17] "[https ://www.digora.com/fr/blog/definition-iot-et-strategie-iot](https://www.digora.com/fr/blog/definition-iot-et-strategie-iot)".
- [18] "[https ://www.gigabyte.com/glossary/mobile-edge-computing](https://www.gigabyte.com/glossary/mobile-edge-computing)".
- [19] "[https ://www.ionos.fr/digitalguide/serveur/know-how/edge-computing/](https://www.ionos.fr/digitalguide/serveur/know-how/edge-computing/)".
- [20] "[https ://www.journaldunet.com/solutions/cloud-computing/1193318-comprendre-le-fog-computing-en-sept-questions/](https://www.journaldunet.com/solutions/cloud-computing/1193318-comprendre-le-fog-computing-en-sept-questions/)".
- [21] "[http ://www.univ-tebessa.dz/fichiers/theses_dissertations/docs/thesesdoc/informatique](http://www.univ-tebessa.dz/fichiers/theses_dissertations/docs/thesesdoc/informatique)".
- [22] Huang, d . ; wang, p . ; niyato, d. un algorithme de floading dynamique pour l’informatique mobile. *ieee trans.* 38. huang, d . ; wang, p . ; niyato, d. un algorithme de floading dynamique pour l’informatique mobile. *ieee trans. wirel. commun.* 2012, 11, 1991–1995.
- [23] Julien lausson "[https ://www.numerama.com/tech/147723-5g-tout-savoir-sur-le-reseau-mobile-du-futur.html](https://www.numerama.com/tech/147723-5g-tout-savoir-sur-le-reseau-mobile-du-futur.html)" , 28 novembre 2019.
- [24] K. kumar, j. liu, y. lu, and b. bhargava, "a survey of computation offloading for mobile systems," *mobile networks and applications*, vol. 18, no. 1, pp. 129–140, 2013.
- [25] K. kumar, y. lu. "cloud computing for mobile users : Can offloading computation save energy," - *ieee computer society 2010.*
- [26] Kosta, s. ; aucinas, a. ; hui, p. ; mortier, r. ; zhang, x. thinkair : Dynamic resource allocation and parallel execution in the cloud for mobile code of [U+FB02] oading. in *proceedings of the t2012 proceedings ieee infocom, orlando, fl, usa, 25–30 march 2012 ; pp. 945–953.*

- [27] Kumar, k. ; liu, j. ; lu, y.h. ; bhargava, b. a survey of computation offloading for mobile systems. *mob. netw. appl.* 2013, 18, 129–140.
- [28] L. yang, j. cao, h. cheng, and y. ji, “multi-user computation partitioning for latency sensitive mobile cloud applications,” *ieee trans. comput.*, vol. 64, no. 8, pp. 2253–2266, 2015.
- [29] L. yang, j. cao, y.yuan, t. li, a. han, a. chan. “a framework for partitioning and execution of data stream applications in mobile cloud computing,” - *acm sigmetrics performance evaluation review* (per) 2013.
- [30] Lin, x. ; zhang, h. ; ji, h. ; leung, vcm calcul conjoint et allocation des ressources de communication dans les réseaux de cloud computing de périphérie mobile. dans les actes de la conférence internationale de l’ieee 2016 sur l’infrastructure de réseau et le contenu numérique (ic-nidc), beijing, chine, 23-25 septembre 2016 ; pp. 166-171.
- [31] M. r. palacín, “recent advances in rechargeable battery materials : a chemist’s perspective,” *chem. soc. rev.*, vol. 38, no. 9, pp. 2565–2575, aug. 2009.
- [32] M. ra, a. sheth, l. mummert, p. pillai, d. wetherall, r. govindan. “odessa : Enabling interactive perception applications on mobile devices,” - *acm mobisys’* 2011.
- [33] Mach,p. ;becvar,z.edge computing : A survey on architecture and computation offloading. *ieee commun. surv. tutor.* 2017, 19, 1628–1656.
- [34] Mao, y. ; zhang, j. ; letaief, k.b. dynamic computation offloading for mobile-edge computing with energy harvesting devices. *ieee j. sel. areas commun.* 2016, 34, 3590–3605.
- [35] Min, c. ; hao, y. ; yong, l. ; lai, c.f. ; di, w. on the computation offloading at ad hoc cloudlet : architecture and service modes. *ieee commun. mag.* 2015, 53, 18–24.
- [36] Modèles de conception pour des applications collaboratives dans le cloud "https://tel.archives-ouvertes.fr/tel-01430151/document".
- [37] Orsini,g. bade,d.lamersdorf, computing at the mobile edge : Design ing elastic android applications for computation offloading. in *proceedings of the i[U+FB01]p wireless mobile networking conference*, colmar, france, 11–13 july 2016.
- [38] P. bahl, r. y. han, l. e. li, and m. satyanarayanan, “advancing the state of mobile cloud computing,” in the third acm workshop on mobile cloud computing and services, 2012.
- [39] P. mell and t. grance, “the nist definition of cloud computing,” sep-2011. [online]. available : <http://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-145.pdf>. [accessed : 16-mar-2017].

- [40] R. newton, s. toledo, l. girod, h. balakrishnan, s. madden. "wishbone : Profile-based partitioning for sensornet applications," - nsdi 2009.
- [41] R. wolski, s. gurun, c. krintz, and d. nurmi, "using bandwidth data to make computation offloading decisions," in *ieee international symposium on parallel and distributed processing*. ieee, 2008, pp. 1–8.
- [42] Renaud venet, url = "[https ://www.renaudvenet.com/cloud-computing-avantages-et-inconvenients-2011-01-26.html](https://www.renaudvenet.com/cloud-computing-avantages-et-inconvenients-2011-01-26.html)", 26/jan/2011.
- [43] Sanaei, z. ; abolfazli, s. ; gani, a. ; buyya, r. heterogeneity in mobile cloud computing : taxonomy and open challenges. *ieee commun. surv. tutor.* 2014, 16, 369–392.
- [44] Sardellitti,s. ;scutari,g. ;barbarossa,s.join to optimization of radio and computation al resources formultice ll mobile-edge computing. *ieee trans. signal inf. process. over netw.* 2015, 1, 89–103.
- [45] Shari[U+FB01],m. ;kafaie,s. ;kashe[U+FB01],o.asurvey and taxonomy of cyber forag ing of mobile devices. *ieee commun. surv. tutor.* 2012, 14, 1232–1243.
- [46] url "[https ://waytolearnx.com/2019/03/difference-entre-iaas-saas-et-paas.html](https://waytolearnx.com/2019/03/difference-entre-iaas-saas-et-paas.html)" mars 18, 2019.
- [47] "vaquero, l, rodero-merino, l, caceres, j and lindner m (2009). a break in the clouds : towards a cloud definition. *acm sigcomm computer communications review*. volume 39, issue 1, january 2009, pp 50-55 ".
- [48] X. zhang, a. kunjithapatham, s. jeong, s. gibbs. "towards an elastic application model for augmenting the computing capabilities of mobile device.
- [49] Xia, f. ; ding, f. ; li, j. ; kong, x. ; yang, l.t. ; ma, j. phone2cloud : Exploiting computation offloading for energy saving on smartphones in mobile cloud computing. *inf. syst. front.* 2014, 16, 95–111.
- [50] Y. wen, w. zhang, and h. luo, "energy-optimal mobile application execution : Taming resource-poor mobile devices with cloud clones," in *ieee infocom*. ieee, 2012, pp. 2716–2720.
- [51] Y. zhang, h. liu, l. jiao, x. fu. "to offload or not to offload : an efficient code partition algorithm for mobile cloud computing," - *ieee 1st international conference on cloud networking* 2012.
- [52] Yang, x. pan, t. and shen, j. 2010. on 3g mobile ecommerce platform based on cloud computing. in *proceedings of the 3rd ieee international conference on ubi-media computing*. (august. 2010), 198-201.
- [53] You, c. ; huang, k. ; chae, h. ; kim, b. energy-ef[U+FB01]cient resource allocation for mobile-edge computation offloading. *ieee trans. wirel. commun.* 2017, 16, 1397–1411.

- [54] Zhao, x., zhao, l. et liang, k. (2016). un algorithme de déchargement orienté consommation d'énergie pour le calcul du brouillard. dans international conference on heterogeneous networking for quality, reliability, security and robustness (pp. 293-301). springer, cham.