

République algérienne démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique
Université Abbes Laghrour Khenchela
Faculté des Sciences et la Technologie
Département des Mathématiques et Informatique



Mémoire de fin d'étude pour l'obtention du
Diplôme de Master en Informatique option : Sécurité et Technologies Web

Vers un nouveau modèle pour l'équilibrage de charge dans le CloudIoT

Réalisé par :
Sofiane BENABBES
Abderrahim NECIB

Dirigé par :
Dr Sofiane HEMAM

Soutenu le **25 juin 2018**, devant le jury:

Dr Nawel TAKOUACHET	MCB	Université Abbes Laghrour Khenchela	Présidente
Dr Mohamed SOUIDI	MAB	Université Abbes Laghrour Khenchela	Examineur
Dr Sofiane HEMAM	MCA	Université Abbes Laghrour Khenchela	Encadreur

Année universitaire: 2017/2018

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Dédicace

A ma petite famille, mon grand Taha.

Au groupe d'encadrement 3H.

Sofiane BENABBES

Dédicace

A mes parents, ma famille et mes chers amis.

Abderrahim NECIB

Remerciements

Nous tenons tout d'abord à remercier ALLAH le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce Modeste travail.

En second lieu, nous tenons à remercier notre encadreur **Mr Sofiane HEMAM**, son précieux conseil et son aide durant toute la période du travail.

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre travail en acceptant de l'examiner et de l'enrichir par leurs remarques et propositions.

A Mr Sofiane **HEMAM**, Mme Ouassila **HIOUAL**, Mlle Ouided **HIOUAL** pour leurs soutiens et orientations surtout durant la période de la préparation de notre article pour la conférence ASD 2018, Maroc (02-03 Mai 2018).

A Mr Abderrahim **SIAM** et Mme Nawel **TAKOUACHET** pour leurs soutiens et encouragements.

A toute l'équipe pédagogique **STW**, pour tout effort fourni durant notre période d'étude.

Aux étudiants de notre promotion **STW 2018**, en précisant Ameer, Abderrezak et Youssef.

Sommaire

Sommaire

Liste des figures et tableaux	I
Introduction Générale	1
Chapitre 1 : Cloud Computing et Internet des objets.....	3
I- Introduction.....	4
II- Cloud Computing.....	4
II-1- Définition	4
II-2- Caractéristiques	4
II-3- Modèles de service.....	5
II-4- Modèles de déploiement.....	7
II-5- Avantages économiques et techniques.....	9
II-6- Inconvénients	9
III- Internet des objets.....	10
III-1- Définition	10
III-2- Applications de l'IdO.....	11
III-3- L'IdO comme système de systèmes	12
IV- Conclusion	14
Chapitre 2 : Le paradigme CloudIdO	15
I- Introduction	16
I-1- Motivation	16
II- Catégories du CloudIdO.....	18
II-1- La communication	18
II-2- L'espace de rangement	18
II-3- Le traitement	19
II-4- La portée	19
III- Nouvelles applications	20
III-1- Soins de santé	21
III-2- Villes intelligentes et communautés	21
III-3- Les Maisons intelligentes et Les compteurs intelligents	21
III-4- Vidéosurveillance	21
III-5- Automobile et mobilité intelligente	21
III-6- Énergie intelligente et Grille intelligente	21
III-7- Logistique intelligente	22
III-8- Surveillance de l'environnement	22
IV- Défis	22
IV-1- Sécurité et confidentialité	23
IV-2- Hétérogénéité	23
IV-3- Performance	23
IV-4- Fiabilité	24
IV-5- Grande échelle	24
IV-6- Aspects juridiques et sociaux	25
IV-7- Big Data	25

IV-8- Réseaux de capteurs	25
IV-9- Surveillance	25
IV-10- Brouillard informatique	26
V – Conclusion	26
Chapitre 3 : Equilibrage de charge de la CloudIdO	27
I- Introduction	28
II - Travaux voisins	28
III- Architecture du modelé proposé	30
III-1- Client	31
III-2- Machine virtuelle	31
III-3- File d’attente	31
III-4- Classifieur des MVs	31
III-5- Routeur	31
III-6- Equilibreur local	31
III-7- Equilibreur général	32
IV- Fonctionnement du modèle proposé	32
IV-1- Classifieur	33
IV-2- Routeur	35
IV-3- Equilibreur local	35
IV-4- Equilibreur général	37
V- Conclusion	39
Chapitre 4 : Implémentation	40
I- Introduction	41
II- Définition de la simulation informatique	41
III- Langage et environnement de développement	41
III-1- Langage de programmation java	41
III-2- CloudSim	42
III-3- L’architecture du CloudSim	42
III-3-1- Cloudlet	43
III-3-2- Virtual Machine	43
III-3-3- DataCenter	43
III-3-4- DataCenterBroker	44
IV- Description de l’application « SAS_Balancer »	44
V- Conclusion	50
Conclusion Générale et Perspectives	51
I- Conclusion Générale	52
II- Perspectives	52
Références Bibliographiques	53

Tables des figures et des tableaux

Tables des figures

Figure	Titre	Page
Figure 1	Caractéristiques de Cloud Computing	5
Figure 2	Modèles de service de Cloud Computing	6
Figure 3	Cloud privé	7
Figure 4	Cloud communautaire	7
Figure 5	Cloud public	8
Figure 6	Cloud hybride	8
Figure 7	L'Internet des objets	10
Figure 8	Domaines d'application IdO	11
Figure 9	Le Cloud comme intergiciel dans le paradigme IdO	17
Figure 10	Services rendus possibles grâce au paradigme CloudIdO	20
Figure 11	Scénarios d'application pilotés par le paradigme CloudIdO et les défis associés	22
Figure 12	Architecture du modèle proposée	30
Figure 13	Diagramme de Séquence du Modèle	32
Figure 14	Algorithme K-means pour classifier les MVs en cinq classes	34
Figure 15	Diagramme d'activité du Classifieur	34
Figure 16	Algorithme WCET pour calculer le pire temps d'exécution	35
Figure 17	Algorithme d'aide à la décision (TOPSIS)	36
Figure 18	Diagramme d'activité TOPSIS	36
Figure 19	Diagramme d'activité de l'équilibreur local	37
Figure 20	Diagramme d'activité de l'équilibreur général	38
Figure 21	Les couches de l'architecture du CloudSim	43
Figure 22	L'interface du SAS_Balancer	45

Figure 23	Création des machines virtuelles et les tâches	45
Figure 24	L'étape de la simulation pas-à-pas (Classification et Routage).	46
Figure 25	L'étape de la simulation pas-à-pas (Equilibrage local « TR»).	46
Figure 26	L'étape de la simulation pas-à-pas (Equilibrage général).	47
Figure 27	Les résultats graphiques (Equilibrage local « TR»).	47
Figure 28	Les résultats graphiques (Equilibrage général).	48
Figure 29	Le temps d'équilibrage par rapport aux Tâches	49
Figure 30	Le temps d'équilibrage par rapport aux MV	50

Tables des tableaux

Tableau	Titre	Page
Tableau 1	Les principaux systèmes technologiques nécessaires au fonctionnement de l'IdO	13
Tableau 2	Aspects complémentaires du Cloud et de l'IdO	17
Tableau 3	Nouveaux paradigmes activés par CloudIdO : tout en tant que service	19
Tableau 4	Liste des MV et Tâches du scénario N°1	48
Tableau 5	Liste des MV et Tâches du scénario N°2	49

Introduction

Générale

Introduction générale

Le Cloud Computing est un modèle informatique basé sur l'internet. Selon NIST « National Institute of Standards and Technology », le Cloud Computing est un modèle permettant d'établir un accès à la demande en réseau vers un bassin partagé de ressources informatiques configurable. Ces ressources peuvent être : des réseaux, des serveurs, de l'espace de stockage, des applications et des services. Elles peuvent être approvisionnées rapidement avec un minimum d'effort de gestion et d'interaction avec le fournisseur de services [48]. D'un autre côté, l'Internet des objets (IdO) repose sur des nœuds (objets) intelligents et auto-configurés, interconnectés dans une infrastructure de réseau dynamique et globale [5]. Il est généralement caractérisé par de petits objets du monde réel, largement distribués, avec une capacité de stockage et de traitement limitée, ce qui implique des préoccupations concernant la fiabilité, la performance, la sécurité et la confidentialité [5]. A l'opposé de l'Internet des objets, le Cloud Computing a des capacités pratiquement illimitées en termes de puissance de stockage et de traitement, c'est une technologie beaucoup plus mature.

A partir de ces deux technologies (Cloud Computing et Internet des Objets) est né, c'est un nouveau paradigme dans lequel le Cloud Computing et l'Internet des Objets sont fusionnés. Ce nouveau paradigme est appelé CloudIdO. Dans CloudIdO, la quantité de données à stocker et à traiter augmentent de jours en jours et de façon très rapide, ce qui nécessite un mécanisme qui permet d'équilibrer la charge de stockage et de traitement entre les différentes machines virtuelles du CloudIdO. L'équilibrage de charge est considéré comme l'un des problèmes clés, il est nécessaire pour répartir la charge de travail entre plusieurs nœuds afin de garantir qu'aucun nœud n'est surchargé [63]. Les algorithmes qui utilisent le principe du premier arrivé premier servi ont marqué leur limite dans l'équilibrage de charge, puisqu'ils ne permettent pas d'atteindre l'objectif de la répartition de charge entre les nœuds dynamiques ou entre les Machines Virtuelles du Cloud. Ainsi, certaines Machines Virtuelles seront surchargées tandis que d'autres seront sous-chargées [14].

Dans ce mémoire, nous proposons un modèle qui permet l'équilibrage de charge entre les différentes machines virtuelles d'un CloudIdO. Le modèle proposé se compose de quatre essentiels composants : Le Classifieur qui, est chargé de constituer cinq classes de Machines Virtuelles (MV) ; de la plus performante à la moins performante classe des machines virtuelles, en utilisant pour cela une des techniques de classification supervisée. Le routeur qui, en premier lieu, calcule les distances entre la tâche reçue et les centres de gravité des cinq classes, puis il envoie la tâche à la classe la plus proche par rapport aux distances précédemment calculées. L'équilibreur local dont le rôle est d'équilibrer la

charge entre les différentes Machines Virtuelles, appartenant à la même classe, toute en prenant en considération le pire temps d'exécution (PTE) d'une tâche et sa taille. Enfin, le composant équilibreur général qui, est chargé d'équilibrer la charge entre les classes en migrant les tâches d'une classe surchargée vers une classe la moins chargée.

Notre mémoire est structuré comme suit : après l'introduction générale, le premier chapitre sera consacré aux Cloud Computing et Internet des Objets, définition et domaine d'exploitation. Dans le deuxième chapitre nous allons présenter le paradigme CloudIdO, puis l'architecture générale du modèle proposé, le fonctionnement et le détail de chaque composant seront proposés et détaillés dans le troisième chapitre, le quatrième chapitre est consacré à l'implémentation et étude de cas. Et nous terminerons ce mémoire par une conclusion générale.

Chapitre I

Cloud Computing et Internet des objets

I- Introduction

Dans ce chapitre nous allons mettre l'accent sur deux technologies très importantes. Ainsi, la première section sera réservée au Cloud Computing, où on va définir ses caractéristiques, ses modèles de service, ses modèle de déploiement et enfin ses avantages et inconvénients. Quant à la deuxième section, elle sera réservée pour la deuxième technologie à savoir l'internet des objets. Dans cette deuxième partie on va définir ce paradigme et ces domaines d'application.

II-Cloud Computing

II-1 : Définition

Le Cloud Computing est l'un des paradigmes computationnels les plus populaires et les plus intéressants, en raison de ses nombreux avantages, qui ont été explorés dans les dernières années. Bien que de nombreuses définitions aient été proposées dans les milieux universitaires et industriels, celle qui a été généralement acceptée par la communauté est celle qui a été formellement proposée par l'Institut national des normes et de la technologie (NIST) : [55]

« Le Cloud Computing est un modèle qui permet un accès réseau omniprésent, pratique et à la demande à un bassin partagé de ressources informatiques configurables (réseaux, serveurs, stockage, applications et services) pouvant être rapidement provisionnées et libérées avec un minimum d'effort de gestion ou interaction avec le fournisseur de services. »

II-2 : Caractéristiques du Cloud Computing

A partir de la définition précédente, il est possible d'extraire des caractéristiques (Figure 1) particulières qui émergent de ce paradigme.

Self-service à la demande

Les services Cloud donnent à un client un accès instantané, dans son créneau horaire voulu, aux ressources informatiques dont il a besoin (par exemple, temps CPU, espace de stockage, accès au réseau, etc.) de manière automatique et libre-service sans nécessiter d'interaction humaine et fournisseur de ressources. [66]

Large accès au réseau

Les ressources demandées par les clients sont fournies via le réseau (par exemple Internet) et sont utilisées par plusieurs applications client avec des plates-formes hétérogènes (telles que des appareils mobiles, ordinateurs portables, stations de travail, etc.) nécessitant des protocoles et mécanismes standard pour y accéder. [66]

Regroupement des ressources

Les ressources informatiques d'un fournisseur de services Cloud sont regroupées pour servir plusieurs clients d'une manière multi-locataire en utilisant différentes ressources physiques et virtuelles qui sont dynamiquement affectées pour satisfaire la demande de l'utilisateur. Ce modèle basé sur un bassin a pour but de rendre les ressources "invisibles" aux consommateurs qui n'ont pas de contrôle ou de connaissance sur l'emplacement, la formation et les originalités de ces ressources (par exemple CPU, base de données, etc.). [66]

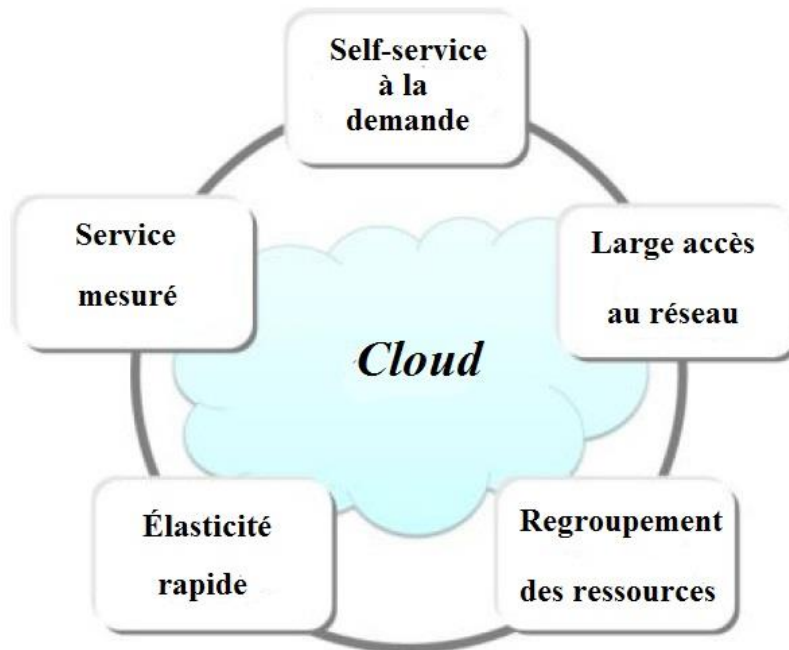


Figure 1: Caractéristiques de Cloud Computing [66]

Élasticité rapide

Les capacités d'un service fourni par un fournisseur de Cloud peuvent être déployées de manière élastique, attribuées et libérées pour être mises à l'échelle chaque fois que les consommateurs le souhaitent et le libèrent une fois qu'ils le souhaitent. De cette façon, les ressources semblent illimitées et peuvent être allouées en n'importe quelle quantité à tout moment. [66]

Service mesuré

Bien que les ressources de calcul soient partagées de plusieurs manières, l'infrastructure sous-jacente est capable de mesurer l'utilisation de ces ressources de chaque consommateur en utilisant des mécanismes appropriés afin de donner un modèle de paiement à l'utilisation à ses clients. [66]

II-3 : Modèles de service

On peut distinguer trois modèles (Figure 2) de services différents qui peuvent être offerts par un fournisseur Cloud dont chacun diffère des autres pour le contrôle donné à l'utilisateur sur les ressources qui lui sont allouées, les fonctionnalités générales et la couche architecturale proposée.

Software-as-a-Service (SaaS)

Dans le modèle **SaaS**, les consommateurs de Cloud libèrent leurs applications auprès du fournisseur choisi qui fournit un environnement d'hébergement auquel il est possible d'accéder via des réseaux. De cette manière, les clients n'ont aucun contrôle sur l'ensemble de l'infrastructure et sa gestion est entièrement prise en charge par le fournisseur.

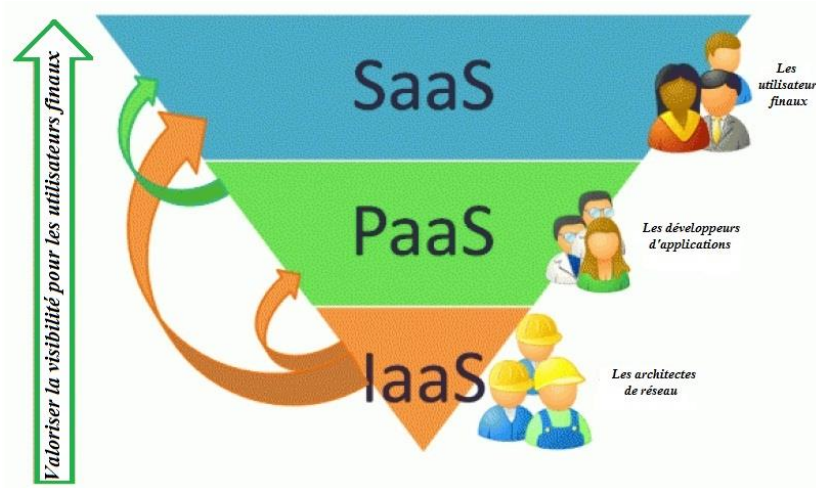


Figure 2 : Modèles de service de Cloud Computing [66].

Plate-forme-as-a-Service (PaaS)

Le modèle **PaaS** consiste en une plate-forme de développement qui supporte tout le cycle de vie du logiciel en fournissant aux clients un environnement complet pour développer ses propres services ou applications.

Infrastructure-as-a-Service (IaaS)

Dans le modèle **IaaS**, les clients Cloud peuvent utiliser directement l'infrastructure informatique pour gérer toute la pile logicielle. Grâce à la technologie de virtualisation, le fournisseur offre toutes les capacités de son infrastructure, du traitement au stockage et aux ressources réseau.

De plus, dans les dernières années, la nouvelle définition de Tout-en-un-Service (**XaaS**) devient encore plus pertinente et largement acceptée par la communauté. Cette définition, qui reprend l'objectif le plus important du Cloud Computing, met également en évidence le rôle fondamental de la fourniture de services à travers ce paradigme computationnel. [66]

II-4 : Modèles de déploiement

A partir de la définition NIST du Cloud Computing, nous distinguons trois catégories d'infrastructure Cloud à travers ses modèles de déploiement et qui sont:

Cloud privé

Une infrastructure de Cloud privé (Figure 3) est mise en service pour une entité ou une organisation unique qui fournit les services qu'elle exécute à ses clients ou employés. Un tel type d'infrastructure Cloud est géré par l'organisation elle-même ou par un tiers qui assure l'utilisation exclusive des ressources.

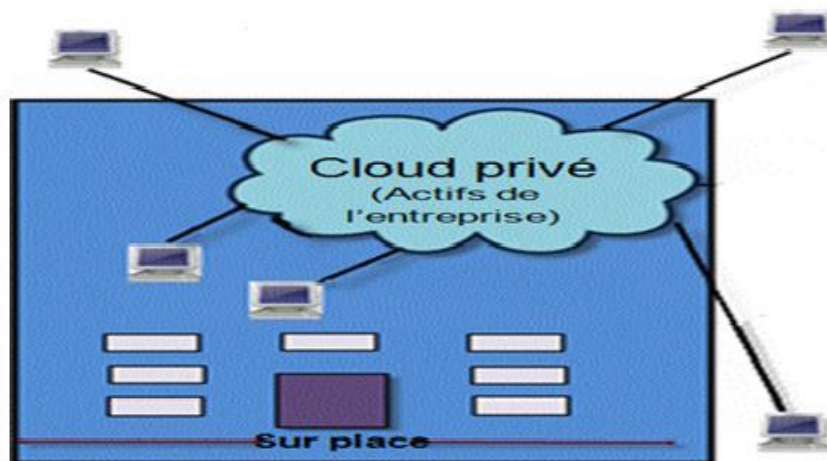


Figure 3 : Cloud privé. [34]

Cloud communautaire

Le modèle de communauté (Figure 4) implique des clients de différentes organisations partageant un objectif commun dans l'utilisation d'une infrastructure Cloud. Un Cloud communautaire peut être détenu et géré par une ou plusieurs organisations impliquées dans la communauté [2], ou comme un Cloud privé fourni par un tiers.



Figure 4 : Cloud communautaire. [34]

Cloud public

Un Cloud public (Figure 5) est une infrastructure de Cloud provisionnée pour une utilisation ouverte. Un fournisseur de Cloud public offre tous les services et les ressources à quiconque qui leur demande dans un modèle payant.

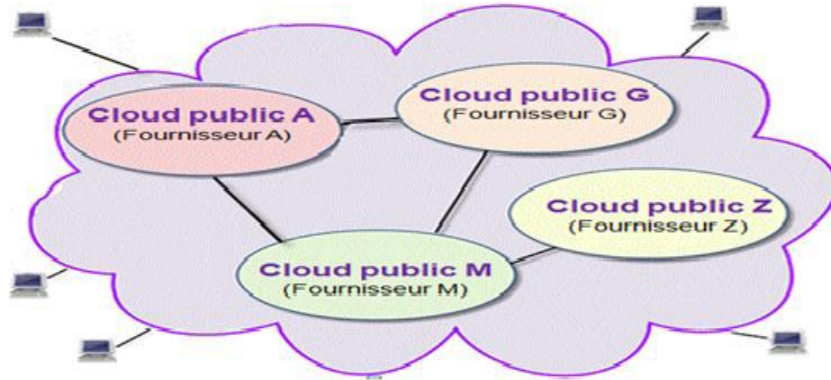


Figure 5 : Cloud public. [34]

Cloud hybride

Dans un Cloud hybride (Figure 6), l'infrastructure est une composition de deux ou plusieurs infrastructures distinctes avec différents modèles de déploiement. Les différentes infrastructures impliquées dans un Cloud hybride sont liées à une technologie standardisée ou propriétaire qui permet la portabilité des données et des applications.

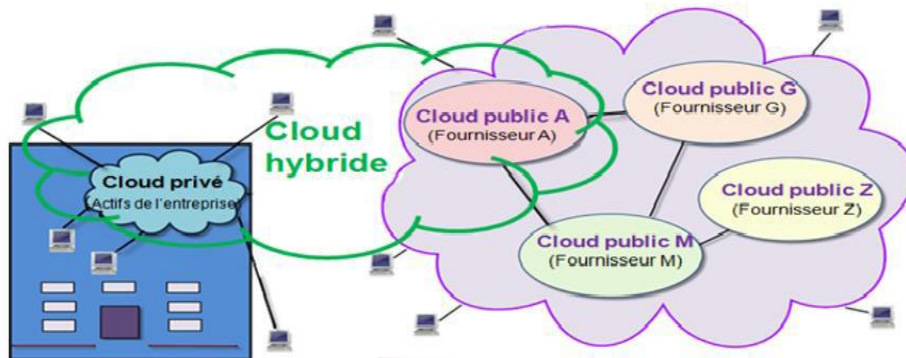


Figure 6 : Cloud hybride. [34]

Dans ce contexte, le paradigme du Cloud Computing offre un large éventail de choix à ceux qui voudraient exploiter les avantages de l'utilisation de ce type d'infrastructures. Cependant, dans les dernières années, de nombreux fournisseurs de services Cloud ont été confrontés au problème pour répondre aux attentes de tous leurs clients qui ont besoin de paramètres de qualité de service (QoS) spécifiques [9]. Pour faire face à ce problème, la communauté des chercheurs a également exploré un

nouveau modèle d'organisation fédérée des infrastructures Cloud. Les nouveaux concepts d'Inter-Cloud et de fédérations Cloud sont devenus très populaires et sont considérés comme le meilleur moyen d'offrir de nombreux avantages aux clients, comme l'atténuation du problème de blocage des fournisseurs (le client est souvent étroitement couplé à un seul fournisseur de Cloud), une fiabilité améliorée et une distribution géographique permettant de surmonter les problèmes de conformité aux lois et règlements. [59]

II-5 : Avantages économiques et techniques.

Avantages économiques

Le modèle de Cloud Computing est attractif car il libère le propriétaire de l'entreprise de la nécessité d'investir dans l'infrastructure (CapEx)¹, louant des ressources en fonction des besoins et ne payant que pour l'usage. De plus, cela permet de réduire les coûts d'exploitation (OpEx)², car les fournisseurs de services n'ont pas à provisionner les capacités en fonction de la charge maximale (en fait, les ressources sont libérées lorsque la demande de service est faible). Enfin, l'externalisation de l'infrastructure de service vers les Clouds déplace le risque métier vers le fournisseur d'infrastructure, généralement mieux équipé pour le gérer [47,55].

Avantages techniques

En plus de ces avantages économiques, le Cloud Computing garantit un certain nombre d'avantages techniques, notamment: l'efficacité énergétique, l'optimisation de l'utilisation des ressources matérielles et logicielles, l'élasticité, l'isolation des performances et la flexibilité [38,55].

II-6 : Inconvénients.

Le problème fondamental reste la sécurisation de l'accès à l'application entre le client et le serveur distant.

D'autre part les entreprises perdent la maîtrise de l'implantation de leurs données ainsi que du cycle de vie des applications, et il n'y aura par ailleurs plus la notion de confidentialité des données (financières, inventions, plans de prospection...). [55]

¹ CapEx : Capital Expenditures ou dépenses d'investissement de capital.

² OpEx : Operational Expenditures ou dépenses d'exploitation.

III- Internet des objets

III-1 : Définition

Le terme Internet of Things [Internet des objets] (IdO) (Figure 7), d'abord introduit par Kevin Ashton en 1998, fait référence à un paradigme émergent qui consiste en une nouvelle architecture de services d'information basée sur Internet. [62] Dans IdO, les « things » se réfèrent à des objets et des appareils connectés les uns aux autres pour former des systèmes beaucoup plus grands permettant de nouveaux scénarios informatiques omniprésents et envahissants.

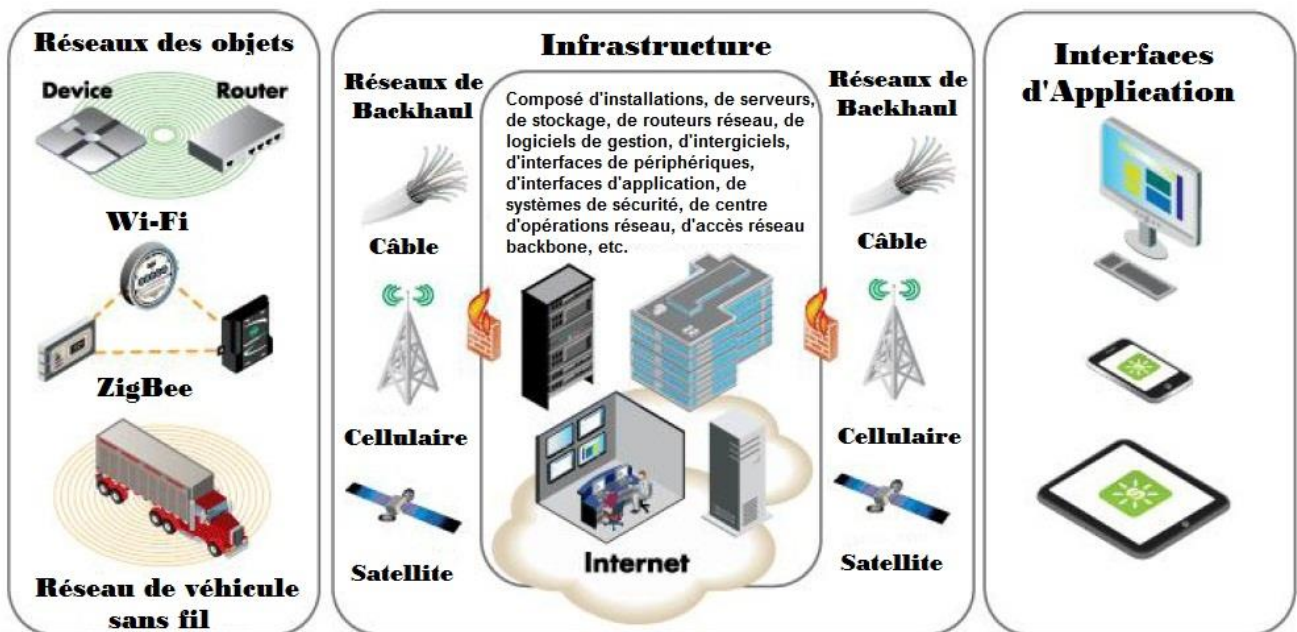


Figure 7 : L'Internet des objets. [62]

Selon le groupe de recherche européen sur l'Internet des objets¹, les objets et les dispositifs impliqués dans ce paradigme émergent sont généralement caractérisés comme le monde réel et les petits objets qui sont impliqués dans plusieurs processus commerciaux, d'information et sociaux. [31]. En effet, il n'y a pas de convergence réelle dans les définitions d'Internet des Objets dans la communauté de recherche, du fait qu'il existe différentes visions du même paradigme. En fait, il existe au moins trois perspectives différentes pour donner une définition du paradigme de l'IdO [46].

Le premier découle d'une perspective « axée sur les objets » dans laquelle l'accent est mis sur les dispositifs concernés qui peuvent être des dispositifs RFID², des capteurs sans fil et des actionneurs ou des objets intelligents de la vie quotidienne. Une autre perspective est une vision « orientée Internet » qui met l'accent sur les technologies habilitantes qui rendent possible la communication des objets impliqués dans une vision orientée réseau. Une troisième perspective dans la vision de l'IdO a une

¹ <http://www.internet-of-things-research.eu/> (17/06/2018).

² RFID : Radio Frequency Identification ou la radio-identification.

approche plus « sémantique ». Considérant que le nombre d'objets connectés ensemble augmente chaque jour de manière exponentielle est crucial pour faire face à des problèmes liés à la façon de stocker, gérer et représenter les données produites par les systèmes de l'Internet des objets.

III-2 : Applications de l'IdO

La collecte et la transmission de données à partir des appareils, des machines et des capteurs impliqués dans l'Internet des Objets et la communication entre eux conduisent à avoir plusieurs domaines d'application (Figure 8) et une flexibilité sans précédent dans leur variété. Une telle variété peut être classée dans au moins quatre domaines d'application : Personnel et Domicile, Entreprise, Utilitaires et Mobile. [36]

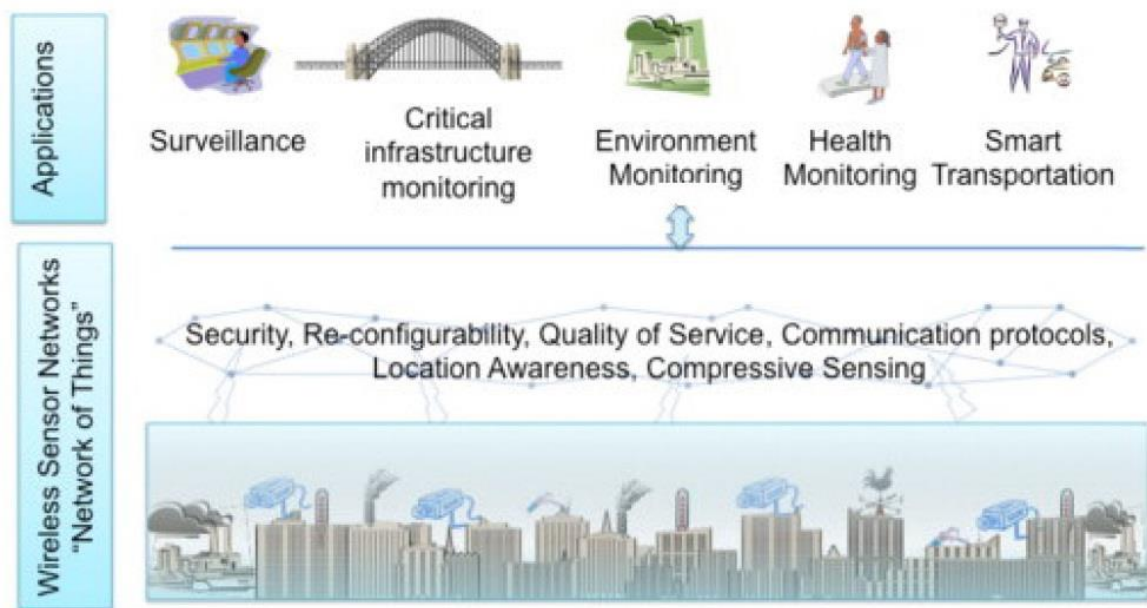


Figure 8 : Domaines d'application IdO [36]

Dans le domaine personnel et domestique, par exemple, diverses applications pour le contrôle des habitations et des bâtiments tels que les climatiseurs, les réfrigérateurs, les lave-linges, etc., peuvent permettre une meilleure gestion de la maison et de l'énergie. De plus, des soins de santé omniprésents [46] ont été envisagés à travers des systèmes IdO qui offrent une plate-forme idéale pour permettre cette vision, par exemple, en utilisant des capteurs de surface corporelle et un back-office IdO pour télécharger des données sur des serveurs. Une extension des capteurs de surface corporelle a été développée pour créer un système de surveillance à domicile pour les soins aux personnes âgées qui permet le suivi des patients âgés par des médecins directement chez eux, réduisant ainsi les coûts d'hospitalisation. [27]

Dans le domaine de l'entreprise, les capteurs font partie intégrante de l'usine et sont principalement utilisés pour la sécurité, l'automatisation, le contrôle climatique, etc. L'avènement d'Internet Des Objets peut remplacer les anciens capteurs par des systèmes sans fil qui permettent de

modifier les paramètres. En outre, une attention croissante dans les dernières années est concentrée sur un environnement intelligent d'IdO dans lequel il est possible d'identifier de nombreuses applications et cas d'utilisation.

Ces applications peuvent être liées aux services d'urgence, à la gestion des traçages, à la surveillance des infrastructures, etc. [68]

Les informations provenant des réseaux de capteurs dans le domaine des utilitaires sont déjà utilisées par les sociétés de services publics pour la gestion des ressources afin d'optimiser les coûts et les protections. Le comptage intelligent et le réseau intelligent sont d'autres applications intéressantes de l'IdO et permettent, par exemple, de réaliser une consommation d'énergie efficace en surveillant les points d'électricité et en utilisant ces données pour changer la manière dont l'électricité est consommée. [36]

La surveillance du réseau d'eau est une autre application cruciale qui est traitée en utilisant l'IdO. La surveillance des paramètres de l'eau à travers les réseaux de capteurs permet d'assurer un approvisionnement de haute qualité et évite la contamination accidentelle entre les égouts pluviaux, l'eau potable et l'évacuation des eaux usées.

Une autre application importante dans le domaine des utilitaires est l'IdO basé sur la vidéo qui prend en charge les systèmes de surveillance. Ce type d'applications intègre la vidéo, l'infrarouge, les microphones et les technologies réseau pour aider à suivre les cibles, identifier les activités suspectes et surveiller les accès non autorisés.

Un autre domaine de l'application IdO implique le déplacement des objets. Le domaine mobile peut inclure le transport intelligent et la logistique intelligente. L'IdO peut permettre d'utiliser le réseau de capteurs existant pour fournir des informations de trafic dynamiques qui peuvent par exemple permettre une meilleure planification et améliorer l'ordonnement des mouvements de marchandises ou collecter des données pour le contrôle du trafic urbain.

III-3 : L'IdO comme système de systèmes [57]

L'IdO ne se résume certainement pas à une technologie spécifique. Il désigne plutôt diverses solutions techniques (RFID, TCP/IP, technologies mobiles etc.) qui permettent d'identifier des objets, de capter, stocker, traiter, et transférer des données dans les environnements physiques mais aussi entre des contextes physiques et des univers virtuels. Actuellement, l'enjeu majeur n'est pas tant d'inventer de nouvelles technologies que de perfectionner celles qui existent déjà, de les connecter, et de les intégrer. Le Tableau 1 indique les différents systèmes technologiques nécessaires au fonctionnement de l'IdO.

Tableau 1 : Les principaux systèmes technologiques nécessaires au fonctionnement de l'IdO. [57]

Type de systèmes	Identification	Capteurs	Connexion	Intégration	Traitement de données	Réseaux
Enjeux	Reconnaître chaque objet de façon unique et recueillir les données stockées au niveau de l'objet.	Recueillir des informations présentes dans l'environnement pour enrichir les fonctionnalités du dispositif.	Connecter les systèmes entre eux.	Intégrer les systèmes pour que les données soient transmises d'une couche à l'autre.	Stocker et analyser les données pour lancer des actions ou pour aider à la prise de décisions.	Transférer les données dans les mondes physiques et virtuels.
Technologies anciennes	Radio-identification simple Code-barres URI GPS	Luxmètre Capteur de proximité Thermomètre Hydromètre	Câbles Radio	Middleware simples	Base de données Tableur Progiciel de gestion intégré Gestion de la relation client	Internet
Technologies récentes	Radio-identification complexe Onde acoustique de surface ADN	Accéléromètre Gyroscope Capteurs miniaturisés Nanotechnologies	Bluetooth Wi-Fi ZigBee Z-Wave Communication en champ proche	Middleware complexes Analyse décisionnelle des systèmes complexes	Entrepôt de données 3D (compatible avec les puces RFID) Web sémantique	EPC Global

Lier un objet ou un lieu à Internet est un processus plus complexe que la liaison de deux pages Web. L'Internet des objets exige sept composants : [57]

1. Une étiquette physique ou virtuelle pour identifier les objets et les lieux. Quelques systèmes d'étiquetage sont décrits ci-dessous. Pour permettre aux étiquettes physiques plus petites d'être localisées elles doivent être embarquées dans des marqueurs visuels.
2. Un moyen de lire les étiquettes physiques, ou de localiser les étiquettes virtuelles.
3. Un dispositif mobile tel qu'un téléphone cellulaire, un assistant personnel ou un ordinateur portable.
4. Un logiciel additionnel pour le dispositif mobile.
5. Un réseau sans fil de type 2G, 3G ou 4G afin de permettre la communication entre le dispositif portable et le serveur contenant l'information liée à l'objet étiqueté.
6. L'information sur chaque objet lié. Cette information peut être contenue dans les pages existantes du Web, les bases de données comportant des informations de type prix, etc.
7. Un affichage pour regarder l'information sur l'objet lié. À l'heure actuelle, il est probable que ce soit l'écran d'un téléphone mobile.

IV- Conclusion

Dans ce chapitre nous avons abordé les deux technologies d'actualité (Cloud Computing et Internet des Objets) de façon séparée. Ces deux technologies ont fait l'objet de plusieurs recherches et études ces dernières années, et leur combinaison a permis la naissance d'un nouveau paradigme appelé : CloudIdO, qui sera l'objet du prochain chapitre.

Chapitre 2

Le paradigme CloudIdO

I- Introduction

Comme nous l'avons déjà expliqué dans le premier chapitre les deux concepts de Cloud Computing et Internet des objets qui ont vu une évolution indépendante dans les dernières années. Cependant, de nombreux acteurs du monde de la recherche et de l'industrie ont vu une convergence intéressante dans ces paradigmes. Plusieurs avantages découlant de l'intégration des systèmes IdO et des services et infrastructures Cloud ont été identifiés. Les systèmes IdO peuvent tirer de nombreux avantages des capacités pratiquement illimitées que le Cloud Computing peut offrir, par exemple, pour compenser les contraintes technologiques (par exemple, le stockage, le traitement, l'énergie) des dispositifs IdO [5].

Cette vision met en évidence que Cloud et IdO présentent de nombreuses caractéristiques complémentaires issues des différentes propositions en littérature. Le Cloud peut jouer le rôle de couche intermédiaire entre les objets et les applications en cachant toute la complexité de la gestion des ressources et les fonctionnalités nécessaires à la mise en œuvre de ces dernières [20]. Cette couche intermédiaire aura un impact sur le développement futur des applications, où la collecte, le traitement et la transmission de l'information engendreront de nouveaux défis à relever.

I-1 Motivation

La convergence entre Internet des Objets et Cloud Computing révèle un nouveau paradigme prometteur nommé dans la littérature sous divers aspects tels que Cloud Of Things ou CloudIoT (CloudIdO) [5]. Le nombre d'appareils connectés à Internet augmente de façon exponentielle et, depuis 2011, a déjà dépassé le nombre de personnes sur Terre. Ils ont déjà atteint les 9 milliards et devraient croître plus rapidement dans les prochaines années pour atteindre les 24 milliards en 2020 [6]. Outre le nombre croissant de systèmes IdO, d'autres aspects révèlent plusieurs motivations pour son intégration entre les infrastructures d'IdO et du Cloud Computing qui seront montrées dans ce chapitre.

Généralement, les services IdO sont fournis en tant que solution verticale isolée dans laquelle tous les composants des applications sont étroitement couplés au contexte d'application spécifique. Apporter des services IdO dans le Cloud peut en faciliter la livraison et le déploiement en tirant parti de toute la flexibilité des modèles Cloud. Dans ce contexte, le Cloud Computing facilite le développement d'applications, rend possible une vision abstraite des systèmes IdO et offre ses services pour répondre au besoin de découpler les applications du contexte spécifique [7].

Les deux mondes: Cloud et de l'IdO sont très différents les uns des autres et, mieux encore, leurs caractéristiques sont souvent complémentaires, comme le montre le Tableau 2. Cette complémentarité est la principale raison pour laquelle de nombreux chercheurs ont proposé et

proposent leur intégration, généralement pour obtenir des avantages dans des scénarios d'application spécifiques.

Tableau 2: Aspects complémentaires du Cloud et de l'IdO [7].

	IdO	Cloud
Déplacement	Omniprésent	Centralisé
Accessibilité	Limitée	Omniprésente
Composants	Choses du monde réel	Ressources virtuelles
Capacités de calcul	Limitées	Pratiquement illimitées
Stockage	Limité ou nul	Pratiquement illimité
Rôle de l'Internet	Point de convergence	Des moyens pour fournir des services

En général, l'IdO peut bénéficier des capacités et des ressources virtuellement illimitées de Cloud pour compenser ses contraintes technologiques (par exemple, stockage, traitement, communication) [43]. Pour ne citer que quelques exemples, Cloud peut offrir une solution efficace pour la gestion et la composition des services IdO ainsi que pour la mise en œuvre d'applications et de services qui exploitent les choses ou les données qu'ils produisent. D'un autre côté, Cloud peut bénéficier de l'IdO en élargissant son champ d'application pour traiter les objets du monde réel de manière plus distribuée et dynamique, et pour fournir de nouveaux services dans un grand nombre de scénarios réels. Dans de nombreux cas, le Cloud peut fournir la couche intermédiaire entre les objets et les applications (Figure 9), cachant toute la complexité et les fonctionnalités nécessaires à la mise en œuvre de ces dernières. Cela aura un impact sur le développement futur des applications, où la collecte, le traitement et la transmission d'informations généreront de nouveaux défis, en particulier dans un environnement multi-Cloud [20, 43].

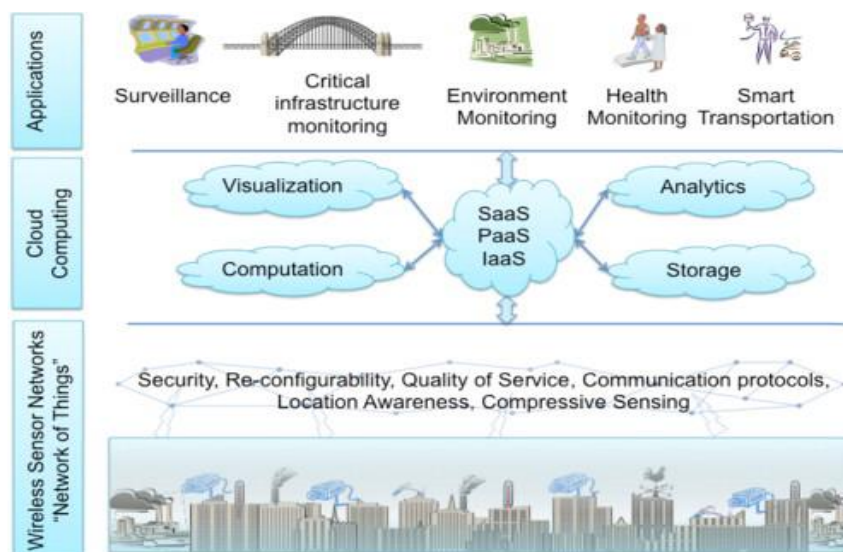


Figure 9 : Le Cloud comme intergiciel dans le paradigme IdO [6]

II- Catégories du CloudIdO

La plupart des études de la littérature considèrent en fait le Cloud comme la pièce manquante dans le scénario intégré, c'est-à-dire qu'ils croient que le Cloud comble certaines lacunes de l'IdO (par exemple le stockage limité). Quelques autres, au contraire, voient l'IdO combler les lacunes du Cloud (principalement la portée limitée). Dans ce mémoire, nous considérons tous les deux comme des pilotes CloudIdO.

La plupart de ces facteurs entrent dans trois catégories : la *communication*, le *stockage* et le *traitement*, tandis que quelques autres sont plus fondamentaux et ont des implications dans toutes ces catégories, c'est-à-dire qu'ils sont transversaux. Étant IdO caractérisé par une très grande hétérogénéité des dispositifs, des technologies et des protocoles, il lui manque différentes propriétés importantes telles que l'évolutivité, l'interopérabilité, la flexibilité, la fiabilité, l'efficacité, la disponibilité et la sécurité.

II-1 : La communication.

Grâce au paradigme CloudIdO, des applications omniprésentes personnalisées peuvent être fournies via l'IdO [24], tandis que l'automatisation peut être appliquée à la fois à la collecte de données et à la distribution à faible coût [53]. Le Cloud offre une solution efficace et peu coûteuse pour connecter, suivre et gérer n'importe quoi, n'importe où et n'importe quand, en utilisant des portails personnalisés et des applications intégrées. Il convient de noter que même si le Cloud peut considérablement améliorer et simplifier la communication de l'IdO [53], il peut néanmoins représenter un goulot d'étranglement dans certains scénarios : en effet, au cours des 20 dernières années, la densité de stockage et la puissance du processeur ont augmenté respectivement de 10^{18} et 10^{15} la capacité n'a augmenté que de 10^4 [37]. En conséquence, des limitations pratiques peuvent survenir lorsque vous essayez de transférer d'énormes quantités de données depuis le bord de l'Internet vers le Cloud.

II-2 : L'espace de rangement.

L'IdO implique par définition une grande quantité de sources d'information (les objets) [20], qui produisent une énorme quantité de données non structurées ou semi-structurées, qui ont aussi les trois caractéristiques typiques du Big Data [75] : volume (ie, taille des données), la variété (c.-à-d. les types de données) et la vitesse (c.-à-d. la fréquence de génération de données). Le stockage à grande échelle et à long terme, possible grâce à la capacité de stockage virtuellement illimitée, à faible coût et à la demande fournie par Cloud, représente un important pilote CloudIdO. Le Cloud est la solution la plus pratique et la plus rentable pour traiter les données produites par l'IdO [60]. Une fois dans le

Cloud, les données peuvent être traitées comme homogènes grâce à des API bien définies, peuvent être protégées en appliquant une sécurité de niveau supérieur, et peuvent être directement consultées et visualisées depuis n'importe quel endroit.

II-3 : Le traitement.

Les appareils IdO ont des ressources de traitement et d'énergie limitées qui ne permettent pas un traitement de données complexe sur site [75]. Les données collectées sont généralement transmises à des nœuds plus puissants où l'agrégation et le traitement sont possibles, mais l'évolutivité est difficile à réaliser sans infrastructure adéquate [60].

Cloud offre des capacités de traitement pratiquement illimitées et un modèle d'utilisation à la demande. Cela représente un autre important pilote CloudIdO :[60]

II-4 : La portée.

L'adoption du paradigme CloudIdO permet de nouveaux services et applications intelligents basés sur l'extension de Cloud à travers les objets qui permettent au Cloud de gérer un certain nombre de nouveaux scénarios réels [21,60], donnant naissance aux objets en tant que Le paradigme du service. C'est un autre moteur important pour CloudIdO [15, 17, 50]. Les recherches montrent comment un certain nombre de nouveaux paradigmes (Tableau 3) émergent de l'intégration de Cloud et IdO et liés à ce pilote particulier.

Tableau 3 : Nouveaux paradigmes activés par CloudIdO : tout en tant que service. [32]

XaaS (Acronyme)	X (Expansion)	Descriptions
	Les objets en tant que service	Agréger et abstraire des ressources hétérogènes selon une sémantique personnalisée
SaaSou S²aas	Le capteur en tant que service	Fournir un accès omniprésent aux données des capteurs
SAaaS	Le capteur et l'actionnement en tant que service	Activation des logiques de contrôle automatiques implémentées dans le Cloud
SEaaS	Événement de capture en tant que service	Envoi de services de messagerie déclenchés par des événements de capture
Senaas	Le capteur en tant que service	Permettant une gestion omniprésente des capteurs à distance

DBaaS	Les bases de données en tant que service	Permettant la gestion de base de données omniprésente
DaaS	Les données en tant que service	Fournir un accès omniprésent à tout type de données
EaaS	Ethernet en tant que service	Fournir une connectivité de couche 2 omniprésente aux périphériques distants
IPMaaS	Identité et gestion des politiques en tant que service	Permettant un accès omniprésent aux fonctionnalités de gestion des politiques et des identités
VSaaS	La vidéosurveillance en tant que service	Fournir un accès omniprésent à la vidéo enregistrée et mettre en œuvre des analyses complexes dans le Cloud

III- Nouvelles applications

CloudIdO a donné naissance à un nouvel ensemble de services et d'applications intelligentes, qui peuvent fortement influencer la vie quotidienne (figure 10). La plupart des applications décrites dans ce qui suit peuvent tirer parti des communications Machine-to-Machine (M2M) lorsque les objets doivent échanger des informations entre eux et non seulement les envoyer vers le Cloud [49]. Dans cette section, nous décrivons le large éventail d'applications rendues possibles ou significativement améliorées et leurs défis grâce au paradigme CloudIdO.

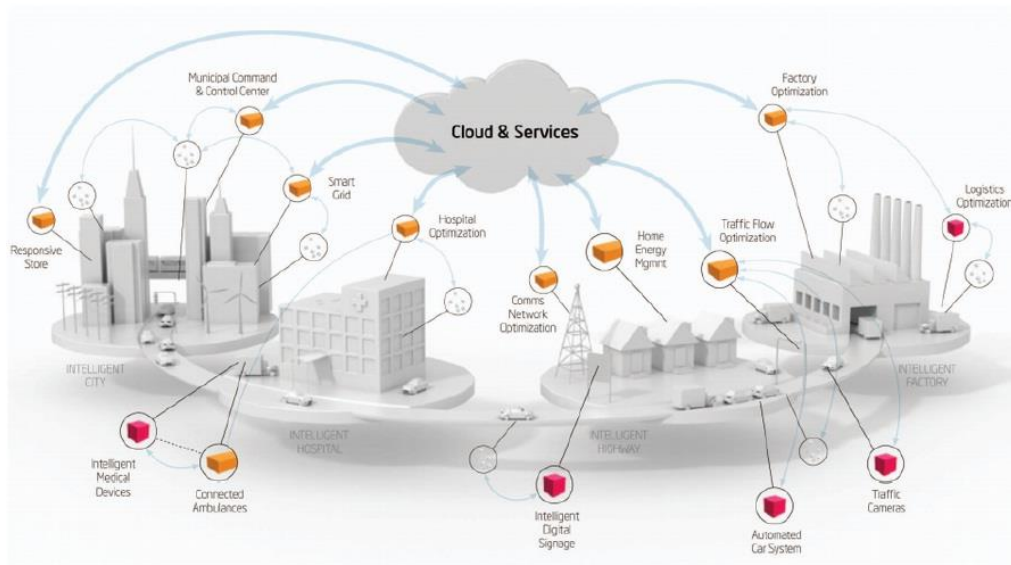


Figure 10 : Services rendus possibles grâce au paradigme CloudIdO.

Source : <https://siliconangle.com/>

III-1 : Soins de santé

L'adoption du paradigme CloudIdO dans le domaine de la santé peut apporter plusieurs opportunités à l'informatique médicale, et les experts estiment qu'elle peut améliorer de manière significative les services de santé et contribuer à son innovation continue et systématique. [25]

III-2 : Villes intelligentes et communautés

CloudIdO conduit à la génération de services qui interagissent avec l'environnement, créant ainsi de nouvelles opportunités pour la contextualisation et la géo-conscience. Le développement durable des zones urbaines est un défi d'une importance capitale et nécessite des technologies et des services nouveaux, efficaces et conviviaux [35].

III-3 : Les Maisons intelligentes et Les compteurs intelligents

Les réseaux domestiques ont été identifiés comme l'environnement où les utilisateurs agissent principalement: CloudIdO a une grande application dans les environnements domestiques, où l'adoption conjointe de dispositifs embarqués hétérogènes et de Cloud permet l'automatisation des activités internes communes [13].

III-4 : Vidéosurveillance

CloudIdO dans le contexte de la vidéosurveillance intelligente permet de stocker, de gérer et de traiter facilement et efficacement le contenu vidéo provenant de capteurs vidéo (caméras IP, par exemple) et d'extraire automatiquement les connaissances des scènes [26].

III-5 : Automobile et mobilité intelligente

En tant que technologie émergente, IdO devrait offrir des solutions prometteuses pour transformer les systèmes de transport et les services automobiles (par exemple, Intelligent Transportation Systems, ITS) [32]. L'intégration du Cloud aux technologies IdO (telles que les WSN et RFID) représente une opportunité prometteuse. [32]

III-6 : Énergie intelligente et Grille intelligente

L'IdO et le Cloud peuvent être fusionnés efficacement pour fournir une gestion intelligente de la distribution et de la consommation d'énergie dans des environnements hétérogènes locaux et étendus.

Les nœuds IdO généralement impliqués dans ce type de processus ont des capacités de détection, de traitement et de mise en réseau, mais des ressources limitées. [71]

III-7 : Logistique intelligente

L'adoption de CloudIdO en logistique favorise un nouveau mode de service qui change radicalement les paradigmes d'affaires. Il permet de nouveaux scénarios intéressants et permet une gestion facile et automatisée des flux de marchandises entre le point d'origine et le point de consommation, afin de répondre à des besoins spécifiques exprimés en termes de temps, de coût ou de moyen de transport [44]. De plus, grâce aux technologies de géolocalisation, il permet de suivre automatiquement les marchandises en transit [67].

III-8 : Surveillance de l'environnement

L'utilisation combinée du Cloud et de l'IdO peut contribuer au déploiement d'un système d'information à haut débit entre l'entité en charge de la surveillance des environnements à grande surface et les capteurs / actionneurs correctement déployés dans la zone [42].

IV- Défis

Le CloudIdO impose plusieurs défis à chaque application qui reçoit actuellement l'attention de la communauté de la recherche. Cette section est consacrée à l'analyse de ces défis. Dans la suite, nous traitons d'abord les défis typiques soulevés par les scénarios d'application rapportés ci-dessus (Figure 11).

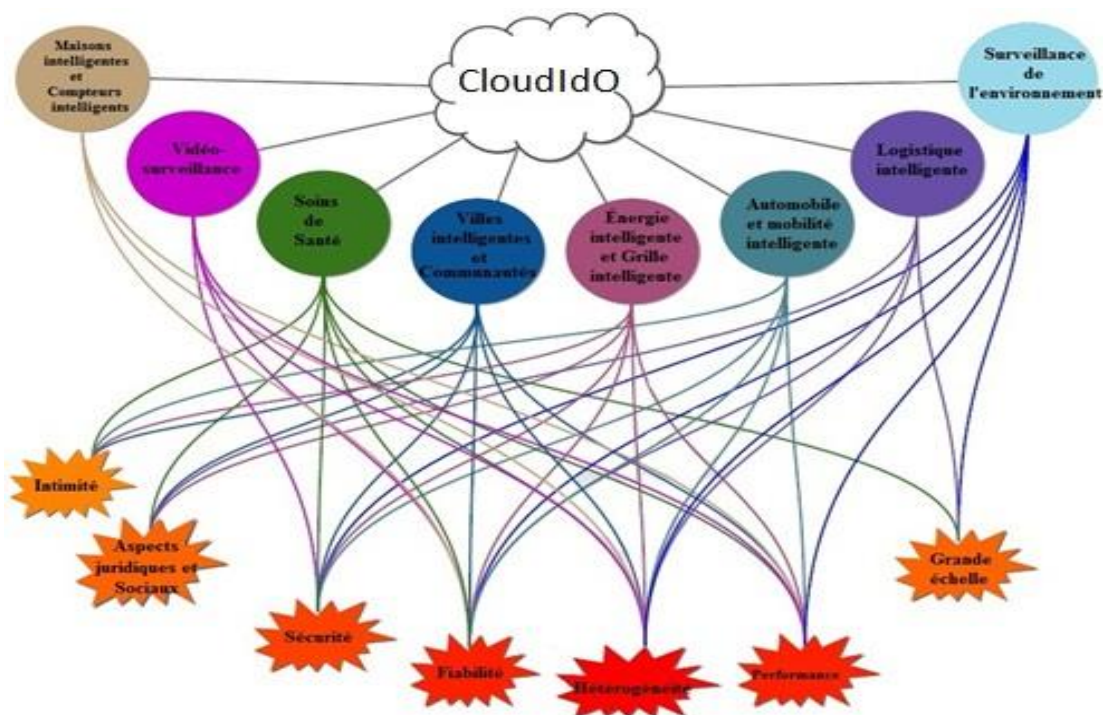


Figure 11 : Scénarios d'application pilotés par le paradigme CloudIdO et les défis associés [64].

IV-1 : Sécurité et confidentialité

Lorsque les applications critiques de l'IdO se déplacent vers le Cloud, des problèmes surviennent en raison du manque de confiance dans le fournisseur de service, de connaissance des accords de niveau de service (SLA : Service-Level-Agreement) et de connaissance de l'emplacement physique des données [64]. En conséquence, les nouveaux défis nécessitent une attention particulière. Un tel système réparti est exposé à plusieurs attaques possibles (par exemple, équitation de session, injection SQL, script intersite et sidechannel) et à des vulnérabilités importantes (par exemple, piratage de session et évocation de machine virtuelle) [64]. La multi-location peut également compromettre la sécurité et conduire à des fuites d'informations sensibles [64]. De plus, la cryptographie à clé publique ne peut pas être appliquée à toutes les couches en raison des contraintes de puissance de calcul imposées par les objets. Ce sont des exemples de sujets actuellement à l'étude afin de relever le grand défi de la sécurité et de la confidentialité dans CloudIdO.

IV-2 : Hétérogénéité

Un grand défi dans CloudIdO est lié à la grande hétérogénéité des périphériques, des systèmes d'exploitation, des plates-formes et des services disponibles et éventuellement utilisés pour des applications nouvelles ou améliorées [64].

L'hétérogénéité des plateformes Cloud est également une préoccupation non négligeable. Les services Cloud sont généralement dotés d'interfaces propriétaires, ce qui permet d'adapter correctement l'intégration des ressources et le mash-up en fonction des fournisseurs spécifiques [64].

Les services et les applications d'IdO ont généralement été conçus comme des solutions verticales isolées, dans lesquelles tous les composants du système sont étroitement liés au contexte d'application spécifique. Pour chaque application / service possible, les fournisseurs doivent étudier les scénarios cibles, analyser les besoins, sélectionner les environnements matériels et logiciels, intégrer des sous-systèmes hétérogènes, développer, fournir une infrastructure informatique et assurer la maintenance des services [10]. D'un autre côté, également grâce aux modèles de prestation de services Cloud, CloudIdO devrait faciliter la fourniture de services IdO. Cependant, bien que les modèles de type PaaS représentent une solution générique pour faciliter le déploiement des applications IdO, leur mise en œuvre nécessite de relever le grand défi de l'hétérogénéité [64].

IV-3 : Performance

Souvent, les applications CloudIdO introduisent des exigences de performance et de QoS spécifiques à plusieurs niveaux [60] (par exemple pour les aspects de communication, de calcul et de stockage) et, dans certains cas particuliers, les exigences de réunion peuvent ne pas être facilement

réalisables. En particulier, l'obtention de performances réseau stables acceptables pour atteindre le Cloud constitue un défi majeur, étant donné que l'augmentation du haut débit n'a pas suivi l'évolution du stockage et du calcul [65]. En fait, dans plusieurs scénarios (par exemple, lorsque la mobilité est requise), l'approvisionnement de données et de services doit être effectué avec une réactivité élevée. Comme la rapidité d'exécution peut être fortement affectée par des problèmes d'imprévisibilité, les applications en temps réel sont principalement sensibles aux problèmes de performance [65, 64]. La facilité d'utilisation et l'expérience utilisateur peuvent également être affectées par une mauvaise qualité de service (par exemple, lorsque le streaming multimédia est nécessaire) [65, 64].

IV-4 : Fiabilité

Lorsque CloudIdO est adopté pour les applications critiques, les problèmes de fiabilité surviennent généralement, par exemple, dans le contexte de la mobilité intelligente, les véhicules sont souvent en mouvement et la mise en réseau et la communication sont souvent intermittentes ou peu fiables [32]. Lorsque les applications sont déployées dans des environnements soumis à des contraintes de ressources, un certain nombre de problèmes liés à la défaillance d'un périphérique ou à des périphériques qui ne sont pas toujours joignables existent [71].

D'une part, les capacités Cloud aident à surmonter certains de ces défis [64] (par exemple, le Cloud améliore la fiabilité des périphériques en permettant de décharger des tâches lourdes et donc d'augmenter la durée de vie des batteries ou de construire une architecture modulaire); D'un autre côté, il introduit des incertitudes liées à la virtualisation des centres de données ou à l'épuisement des ressources. Le manque d'analyses de fiabilité et de développement d'études de cas spécifiques exacerbe le défi [32, 71, 60].

IV-5 : Grande échelle

CloudIdO permet de concevoir de nouvelles applications visant à intégrer et analyser des informations provenant de dispositifs réels (embarqués) [56, 42, 60]. Certains des scénarios représentés nécessitent implicitement l'interaction avec un très grand nombre de ces périphériques, généralement distribués dans des environnements étendus. La grande échelle des systèmes résultants rend les défis typiques plus difficiles à surmonter [56, 42, 60] (par exemple, les exigences concernant la capacité de stockage et la capacité de calcul pour un traitement ultérieur deviennent difficiles à satisfaire quand on fait face à des données de longue durée collectées à haut débit). De plus, la distribution des périphériques IdO rend les tâches de surveillance plus difficiles car ils doivent faire face à des problèmes de dynamique de latence et de connectivité.

IV-6 : Aspects juridiques et sociaux

Ce sont deux défis importants, en partie liés. Les aspects juridiques sont extrêmement importants et réels dans la recherche actuelle pour des scénarios d'application spécifiques [16]. Pensez, par exemple, à un service CloudIdO basé sur des données fournies par l'utilisateur. Dans ce cas, d'une part, le fournisseur de services doit se conformer à différentes lois internationales. D'autre part, les utilisateurs doivent recevoir des incitations afin de contribuer à la collecte de données. En termes plus généraux, les aspects sociaux présentent un intérêt pour la recherche et sont actuellement considérés comme un défi intéressant [16].

IV-7 : Big Data

Avec un nombre estimé de 50 milliards d'appareils qui seront mis en réseau d'ici 2020, une attention particulière doit être accordée au transport, au stockage, à l'accès et au traitement de l'énorme quantité de données qu'ils produiront.

Grâce au développement récent des technologies, l'IdO sera l'une des principales sources de Big Data, et le Cloud permettra de le stocker longtemps et de réaliser des analyses complexes. L'omniprésence des appareils mobiles et l'omniprésence des capteurs font en effet appel à des plateformes informatiques évolutives (chaque jour, 2,5 milliards d'octets de données sont créés) [18].

IV-8 : Réseaux de capteurs

Les réseaux de capteurs ont été définis comme le principal moteur de l'IdO [72] et l'une des cinq technologies qui façonneront le monde, offrant la capacité de mesurer, déduire et comprendre les indicateurs environnementaux, des écologies délicates et des ressources naturelles aux environnements urbains [28]. Les progrès technologiques récents ont rendu disponibles des dispositifs miniaturisés efficaces, à faible coût et de faible puissance pour une utilisation dans des applications de télédétection à grande échelle [54]. Le Cloud offre de nouvelles possibilités d'agrégation des données de capteurs et d'exploitation des agrégats pour une couverture et une pertinence plus grande, [73, 54] mais affecte en même temps la confidentialité et la sécurité. De plus, le manque de mobilité étant un aspect typique des appareils IdO courants, la mobilité des capteurs introduits par les Smartphones ainsi que l'électronique portable représentent un nouveau défi.

IV-9 : Surveillance

Comme largement documenté dans la littérature, la surveillance est une activité essentielle dans les environnements Cloud pour la planification de la capacité, la gestion des ressources, les SLA, les performances et la sécurité [3], ainsi que pour le dépannage. En conséquence, CloudIdO hérite des

mêmes exigences de surveillance de la part du Cloud, mais les problèmes liés au volume, à la variété et à la vitesse de l'IdO affectent également les défis associés [3].

IV-10 : Brouillard informatique

Le brouillard informatique est une extension du Cloud Computing classique à la périphérie du réseau (le brouillard étant un nuage proche du sol). Il a été conçu pour prendre en charge des applications IdO caractérisées par des contraintes de latence et des exigences de mobilité et de géo-distribution [1, 74].

V- Conclusion

L'intégration du Cloud Computing et de l'Internet des Objets représente le prochain grand pas en avant dans le futur Internet. Les nouvelles applications issues de cette intégration - appelé CloudIdO - ouvrent de nouvelles directions intéressantes pour les affaires et la recherche.

Les questions ouvertes du paradigme CloudIdO concernent principalement l'efficacité énergétique, l'application des SLA, la tarification et la facturation, la sécurité et la confidentialité et l'équilibrage de charge.

Dans le prochain chapitre, nous proposons une approche dans laquelle nous équilibrons la charge entre les différentes machines virtuelles du CloudIoT

Chapitre 3

Equilibrage de charge dans le CloudIdO

I- Introduction

Dans ce chapitre, nous proposons un modèle qui permet l'équilibrage de charge entre les différentes machines virtuelles d'un CloudIdO. Le modèle proposé se compose de quatre essentiels composants : Le Classifieur qui, est chargé de constituer cinq classes de Machines Virtuelles (MV) ; de la plus performante à la moins performante classe des machines virtuelles, en utilisant pour cela une des techniques de classification supervisée. Le routeur qui, en premier lieu, calcule les distances entre la tâche reçue et les centres de gravité des cinq classes, puis il envoie la tâche à la classe la plus proche par rapport aux distances précédemment calculées. L'équilibreur local dont le rôle est d'équilibrer la charge entre les différentes Machines Virtuelles, appartenant à la même classe, tout en prenant en considération le pire temps d'exécution (PTE) d'une tâche et sa taille. Enfin, le composant équilibreur général qui, est chargé d'équilibrer la charge entre les classes en migrant les tâches d'une classe surchargée vers une classe la moins chargée.

Ce chapitre sera structuré comme suit : après l'introduction, la section 2 sera consacrée aux travaux voisins. Dans la section 3 nous allons présenter d'abord l'architecture générale du modèle proposé, puis le fonctionnement et le détail de chaque composant seront expliqués dans la section 4. Et nous terminerons ce chapitre par une conclusion dans la section 5.

II- Travaux voisins

L'équilibrage de charge est une technique importante pour améliorer les performances des systèmes. Les demandes des utilisateurs doivent être réparties de façon égale entre les machines ou nœuds afin d'obtenir une réponse rapide [12]. L'objectif de l'équilibrage de charge est d'accroître l'utilisation des ressources, de réduire le temps de réponse et la surcharge de certains nœuds.

L'équilibrage de charge peut être réalisé par différentes approches avec différents degrés d'efficacité. Nous pouvons distinguer entre deux types d'équilibrage de charge : statique et dynamique. Ces approches dépendent de la manière dont un algorithme d'équilibrage de charge est basé pour prendre sa décision, c'est-à-dire qu'il est basé sur l'état actuel du système ou non [4]. L'équilibrage de charge statique peut être défini par une situation dans laquelle une charge de calcul est partitionnée entre des nœuds par un algorithme exécuté avant l'exécution du programme, c'est-à-dire l'assignation d'un ensemble de requêtes à un ensemble de ressources; alors que dans l'approche dynamique, l'algorithme d'équilibrage de charge est basé sur l'état actuel du système, de sorte que les demandes sont déplacées dynamiquement d'un nœud surchargé à un nœud sous-chargé, suivant la disponibilité des ressources système [52]. Il existe deux catégories d'algorithmes d'équilibrage de charge dynamique

: les algorithmes centralisés et distribués. Dans l'algorithme d'équilibrage de charge dynamique distribué, tous les nœuds du système doivent exécuter l'algorithme et interagir entre eux pour réaliser l'équilibrage de charge. Cette interaction peut être coopérative ou non coopérative. L'interaction entre les nœuds dans les algorithmes d'équilibrage de charge dynamique distribués génère plus de messages par rapport aux algorithmes dynamiques centralisés. Cependant, même avec la plus grande complexité d'exécution ; les algorithmes dynamiques ont le potentiel d'offrir de meilleures performances que les algorithmes statiques [52].

Plusieurs travaux relatifs à l'équilibrage de charge dans le Cloud Computing ont été proposés ces dernières années. Dans ce papier, nous citons les travaux les plus récents. Ainsi, dans [14], les auteurs ont proposé un algorithme d'équilibrage de charge qui, permet d'améliorer les performances de l'environnement Cloud en fonction de la méthode d'estimation du temps de fin de service. Ils ont réussi à améliorer le temps de service et le temps de réponse de l'utilisateur.

Le problème de consommation d'énergie a été abordé par les auteurs de [69]. Ainsi, ils expliquent que la consommation d'énergie est devenue un défi majeur dans les infrastructures de Cloud Computing. Pour cela, Ils ont proposé une nouvelle méthode d'équilibrage de la charge, appelée ICAMMT. Cette dernière, permet de gérer la consommation d'énergie dans les centres de données du Cloud Computing.

Pour satisfaire les besoins des utilisateurs, les auteurs de [39] ont proposé un algorithme qui vise à maximiser la satisfaction des utilisateurs, en minimisant le temps de réponse des tâches et en améliorant l'utilisation des ressources grâce à une allocation équitable des ressources Cloud.

Les auteurs de [22] ont proposé une nouvelle conception d'équilibrage de charge. Leur modèle est basé sur deux phases principales : la première, est la pré-estimation du temps d'exécution des instructions d'une tâche, en se basant sur les méthodes qui calculent le pire temps d'exécution. Quant à la deuxième phase, elle consiste à ordonnancer les tâches selon des niveaux de classification afin d'allouer des tâches à des machines virtuelles avec la garantie de l'égalité au niveau de la répartition de la charge et de diminuer la dégré de déséquilibrage de charge en prenant en considération les caractéristiques des tâches avant leurs allocations.

Les auteurs de [8] expliquent que, les équilibreurs de charge peuvent être déployés en fonction de plusieurs architectures différentes. L'architecture d'équilibrage de charge centralisée, qui se base sur seul un équilibreur de charge central, permet de maintenir l'ensemble du système à un état équilibré, en sélectionnant la ressource de Cloud qui, devrait prendre en charge le travail reçu. Dans l'architecture d'équilibrage de charge hiérarchique, un équilibreur de charge principal (parent) reçoit toutes les demandes de travail, puis les répartit vers d'autres équilibreurs de charge connectés (enfants) où chaque équilibreur de charge de l'arborescence peut utiliser un algorithme différent.

Les auteurs de [33] ont proposé une nouvelle approche basée sur l'idée de cloner un service Cloud sur un ou plusieurs nœuds, lorsque le nombre de requêtes utilisateur sera important à un instant donné. Ils ont proposé un algorithme qui calcul la moyenne de la charge des nœuds afin de sélectionner les nœuds les moins chargés.

A la différence de ces travaux, notre travail consiste à équilibrer la charge au niveau de chaque classe de façon décentralisée, puisque les équilibreurs locaux sont indépendants entre eux, ce qui permet un gain en temps de service et de réponse d'un côté, et ils permettent de sélectionner la machine virtuelle qui correspond le mieux à la tâche reçu d'un autre côté. Quant à l'équilibreur central, il permet de mettre en contact les équilibreurs locaux des classes les surchargées et les moins-chargées afin que ces dernières collaborent entre eux pour migrer les tâches d'une classe à une autre.

III- Architecture proposée

Cette section décrit l'architecture proposée (Cf Figure 12). Cette dernière est constituée de deux types d'équilibreur de charge : les équilibrages locaux qui se trouvent au niveau de chaque classe de machines virtuelles, et un équilibreur central.

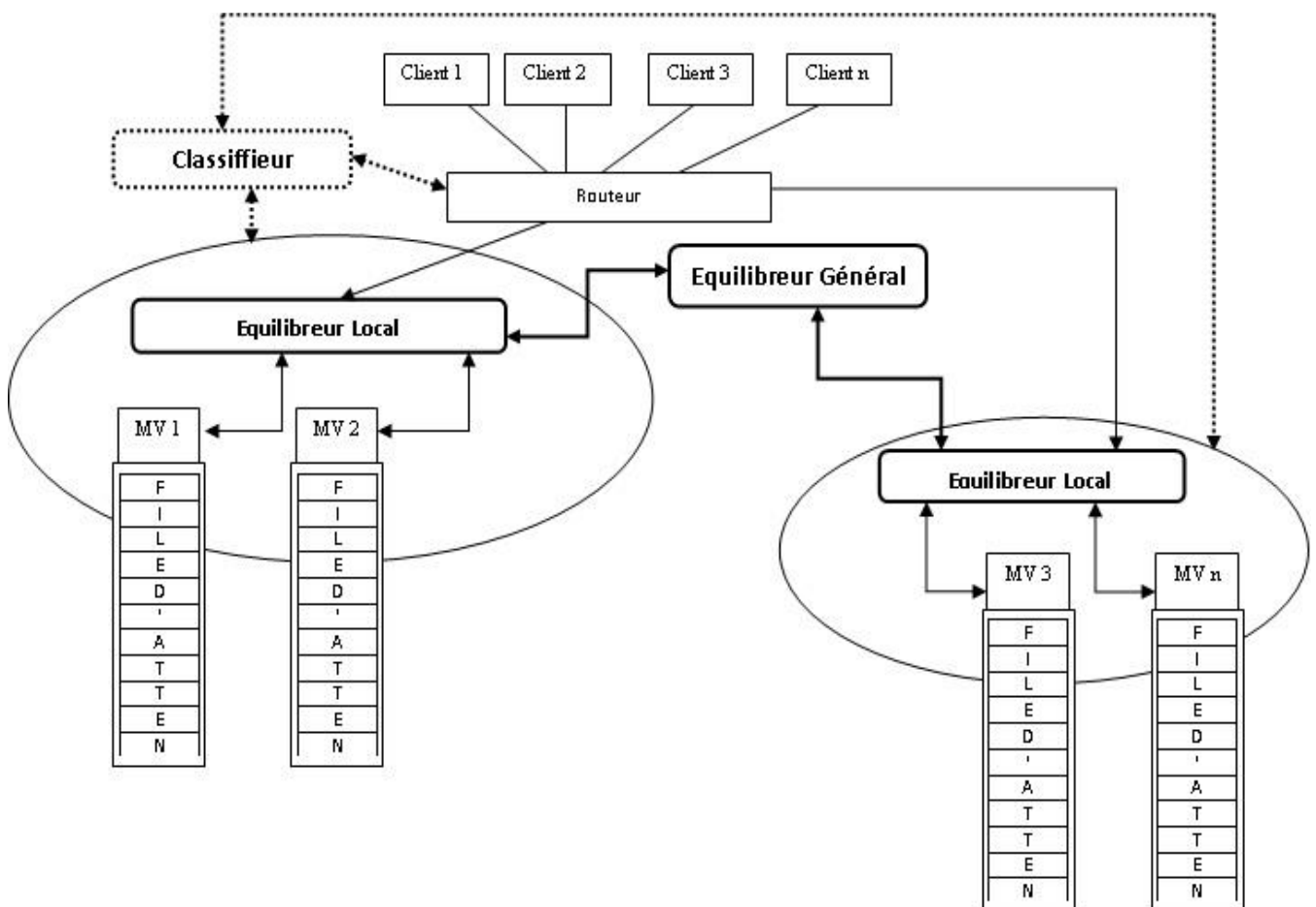


Figure 12 : Architecture du modèle proposée.[11]

En plus de ces deux composants, notre architecture contient le composant Classifieur, dont son rôle est de classer les machines virtuelles en utilisant la méthode K-means, et le composant routeur qui permet d'envoyer les tâches des clients vers les classes qui leur conviennent. Les classes d'appartenance des machines virtuelles sont : Très-Rapide (TR), Rapide (R), Moyenne (M), Lente (L), Très-Lente (TL). Notre architecture est composée, aussi, des clients, des machines virtuelles et leurs files d'attente.

III-1 : Client

Les clients (les objets) envoient leurs tâches au routeur. Les tâches se composent d'un ensemble d'instruction, et se caractérisent par leurs pires temps d'exécution qui est exprimé en Millions d'Instructions Par Seconde (MIPS) et leurs tailles.

III-2 : Machine virtuelle

Une machine virtuelle se compose d'un ensemble de logiciels et matériels géographiquement distant, elle permet d'exécuter les tâches émissent par les clients.

III-3 : File d'attente

La file d'attente est une structure qui permet d'ordonner les tâches, en attente d'exécution, selon leur ordre d'arrivé. Sa structure permet un accès direct aux tâches.

III-4 : Classifieur des MVs

Le composant Classifieur permet de classer les MVs en cinq classes en fonction de la vitesse du CPU et la taille de la RAM en utilisant la méthode K-means[23]. Ce composant est déclenché à chaque fois que le nombre de machines virtuelles change.

III-5 : Routeur

Ce composant est responsable de router les tâches des clients vers les classes. Pour cela, en fonction des caractéristiques (pire temps d'exécution et taille) d'une tâche, et en fonction des caractéristiques des classes. Le routeur sélectionne la classe adéquate pour l'exécution d'une tâche donnée.

III-6 : Equilibreur local

Chaque classe possède son propre équilibreur de charge appelé équilibreur local. Son premier rôle est d'affecter les tâches, reçues de la part du routeur, aux MVs correspondantes. Il permet aussi d'équilibrer la charge à l'intérieur de la classe à chaque fin d'exécution d'une tâche.

III-7 : Equilibreur général

De la même façon que l'équilibreur local, ce composant assure l'équilibrage de charge entre les classes. Il est déclenché périodiquement, ou à la demande d'un équilibreur de charge local.

IV- Fonctionnement du modèle proposé

Comme le montre la figure 13, les composants du modèle proposé interagissent entre eux afin de garantir une charge équitable entre les différentes machines virtuelles, ce qui permet de réduire le temps de réponse et de service.

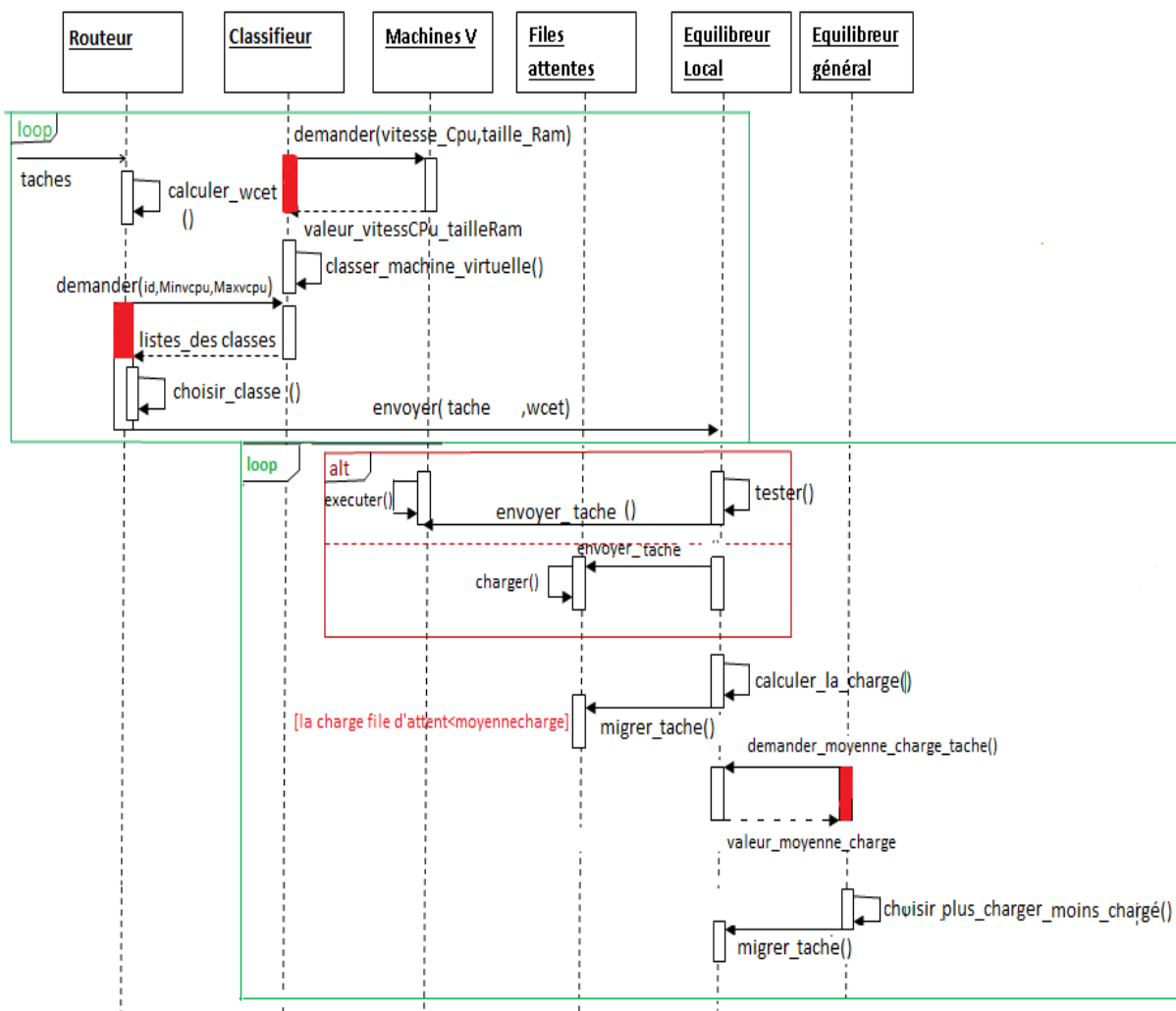


Figure 13 : Diagramme de Séquence du Modèle.[11]

Ainsi, le fonctionnement de notre modèle est devisé en trois niveaux. Le premier niveau permet de classer les machines virtuelles en 5 classes par rapport à la vitesse des processeurs (très rapide, rapide, moyen, lent et très lent) et à la taille de la mémoire de ces MVs. Ce niveau est réalisé en appliquant la méthode de classement K-means, cet algorithme est exécuté la première fois, et à chaque

fois où le nombre de machines virtuelles change. Le routeur permet d'affecter une tâche à une classe des MVs qui lui corresponde.

Le deuxième niveau se situe à l'intérieur de la classe. Une fois que le pire temps d'exécution de la tâche est calculé par le routeur, le composant équilibreur de charge local affecte cette tâche à la MV qui lui correspond. A un moment donné, il va y avoir un déséquilibre de charge entre à l'intérieur de la classe car le choix de MV est basé sur le pire temps d'exécution d'une tâche par rapport à la vitesse du processeur de la MV, c'est-à-dire que probablement certaines tâches vont terminer avant leur temps estimé, ce qui provoque un déséquilibre entre les MVs de la même classe. C'est dans cette situation que l'équilibreur local se déclenche afin d'équilibrer la charge entre les MVs de la même classe.

Le troisième niveau est l'équilibrage de charge entre les classes. Ainsi, l'équilibreur général est responsable de l'équilibrage de charge entre les classes. Pour cela, il sélectionne les classes surchargées afin de migrer les tâches de ces dernières vers les classes les moins chargées.

IV-1 : Classifieur

Le Classifieur applique la méthode de K-means. Ainsi, il permet de classer toutes les MVs du système en cinq classes (TR, R, M, L, TL). Les MVs sont caractérisées par deux critères (Vitesse CPU et Taille RAM). La méthode commence par affecter les cinq premières MVs aux cinq classes respectivement puis, à chaque itération elle calcule la distance entre la MV et le centre de gravité de la classe.

$$D_{xy} = \sqrt{(Xcpu - Ycpu)^2 + (Xram - Yram)^2} \quad (1)$$

Où D_{xy} est la distance entre deux nœuds.

$$\left. \begin{aligned} C_i: CPU_i &= \frac{\sum_{j=1}^n MV_j(cpu)}{n} \\ C_i: RAM_i &= \frac{\sum_{j=1}^n MV_j(ram)}{n} \end{aligned} \right\} \quad (2)$$

Où C_i (CPU, RAM) est le centre.

Le Classifieur affecte la MV à la classe où la distance est la minimale, jusqu'à ce que la structure des classes soit la même.

Les classes sont nommées TR, R, M, L, TL, et chacune d'elle est caractérisée par son centre de gravité C_i .

1. Créer la liste des classes (TR, R, M, L, TL)
2. Récupérer les machines virtuelles (CPU, RAM)
3. Affecter les cinq première MV aux cinq classes respectivement.
4. Calculer le centre de chaque classe (Formule 2)
5. Calculer la distance entre chaque MV et les cinq centres (Formule 1)
6. Affecter chaque MV à la classe la plus proche.
7. Tant que l'état des cinq classes est différent aller vers 8 sinon aller vers 12
8. Calculer le centre de chaque classe (Formule 2)
9. Calculer la distance entre chaque MV et les cinq centres (Formule 1)
10. Affecter la MV à la classe où la distance est la plus petite.
11. Aller vers 7
12. Fin

Figure 14 : Algorithme K-means pour classifier les MVs en cinq classes.

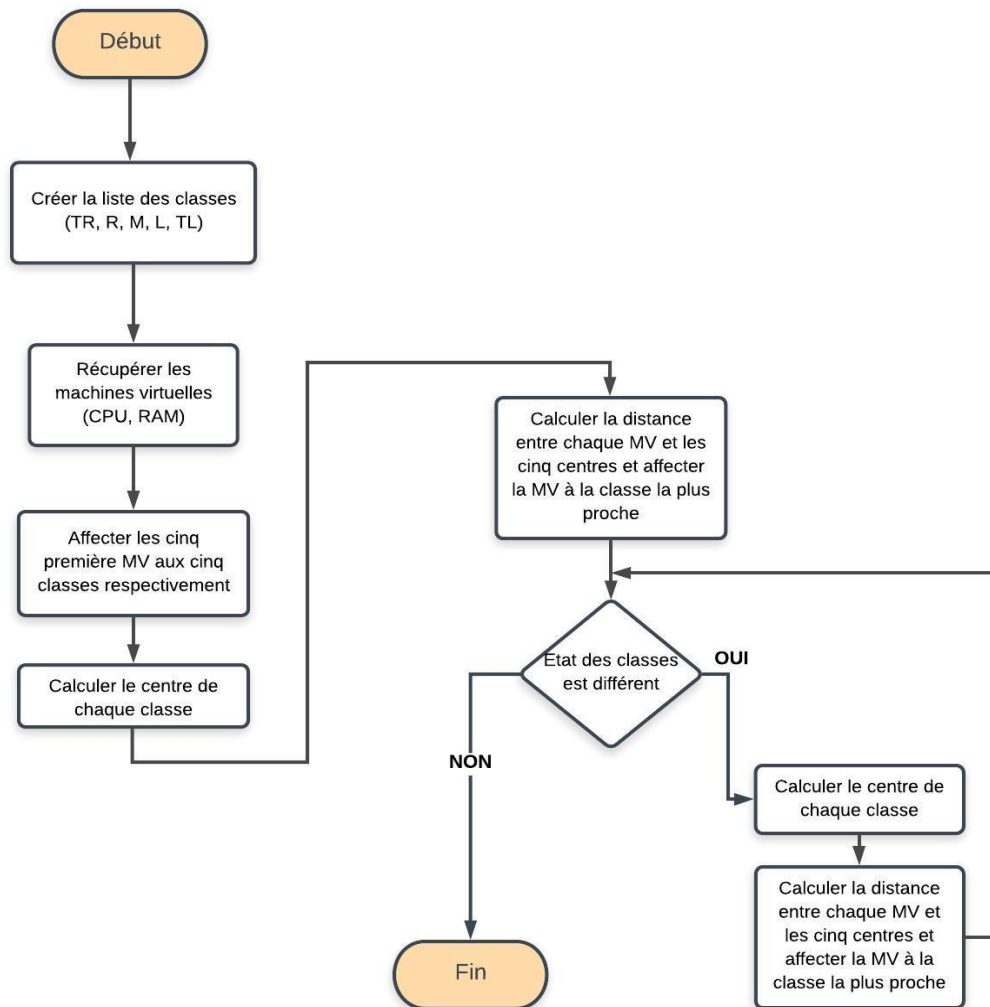


Figure 15 : Diagramme d'activité du Classifieur.[11]

IV-2 : Routeur

Le Routeur commence par calculer le pire temps d'exécution d'une tâche en se basant sur les travaux de [22] (voir la figure 16), puis en fonction de ce temps calculé, de la taille de la tâche et en fonction du centre de gravité C_i de chaque classe, il sélectionne la classe la plus adéquate à la tâche.

```

//La Tâche est un ensemble d'actions (boucle, affectation, test....)
//TTime : tableau dynamique
// On prend le simple algorithme du WCET
1. Parcourir l'arbre des actions en boucle multicritère, si la fin de l'arbre allers vers 5.
2. A chaque fin de feuille on calcul le nombre d'action
3. Si la fin de tache mettre le résultat dans le tableau TTime et Allers vers 1
4. Sinon allers vers 2
5. Parcourir le tableau TTime en cherchant le max
6. WCET :=max(w)
    
```

Figure 16 : Algorithme WCET pour calculer le pire temps d'exécution.

IV-3 : Equilibreur local

Il reçoit la tâche, émise par le routeur, avec son pire temps d'exécution afin de l'affecter à la MV qui lui convient. A chaque arrivé ou départ d'une tâche, cet équilibreur assure l'équilibrage de charge local de sa classe. Pour cela, il calcul la charge de chaque MV (MV_c) selon la formule (3).

$$MV_c = Vp * \sum_{j=1}^m Tj \quad (3)$$

Où Vp est la vitesse du processeur, Tj est le pire temps d'exécution de la tâche et m est le nombre de tâches en attente d'exécution. Puis il calcul la moyenne de charge de la classe selon la formule (4), où n est le nombre de MVs

$$Moy_c = \frac{\sum MV_c}{n} \quad (4)$$

Après il sélectionne les MVs surchargées, ç-à-d ceux dont leur charge est supérieure à la moyenne, afin de migrer certaines tâches vers les MVs les moins chargées. Le choix de la tâche à migrer se fait en se basant sur deux critères essentiels qui sont : la taille de la tâche et son pire temps d'exécution. Afin de minimiser le cout de migration, l'équilibreur local doit choisir la tâche dont la taille est la plus petite possible mais le pire temps d'exécution est le plus élevé. Pour sélectionner une tâche qui satisfait ces deux critères, on utilise la méthode TOPSIS [70] (voir les figures 17 et 18).

1. Lecture des critères
2. Lecture des poids
3. For i : = 1 to n do
4. For j : =1 to 2 do
5. Construire la matrice d'entrée // ligne = Taches, colonne = temps et taille
6. Normaliser la matrice en utilisant la distance euclidienne
7. Calcul de la meilleure et la pire solution (A+ et A-)
8. Calcul de la distance de chaque alternative de la pire et la meilleure solution (E+ et E-)
9. E+ vecteur exprimant la distance de chaque alternative de A+
10. E- vecteur exprimant la distance de chaque alternative de A-
11. Calcul de la proximité de chaque alternative :
12. Le coefficient de proximité de chaque alternative : $S = E- / (E- + E+)$;
13. Afficher (Max(S[i])) ; // la tâche à choisir

Figure 17 : Algorithme d'aide à la décision (TOPSIS).

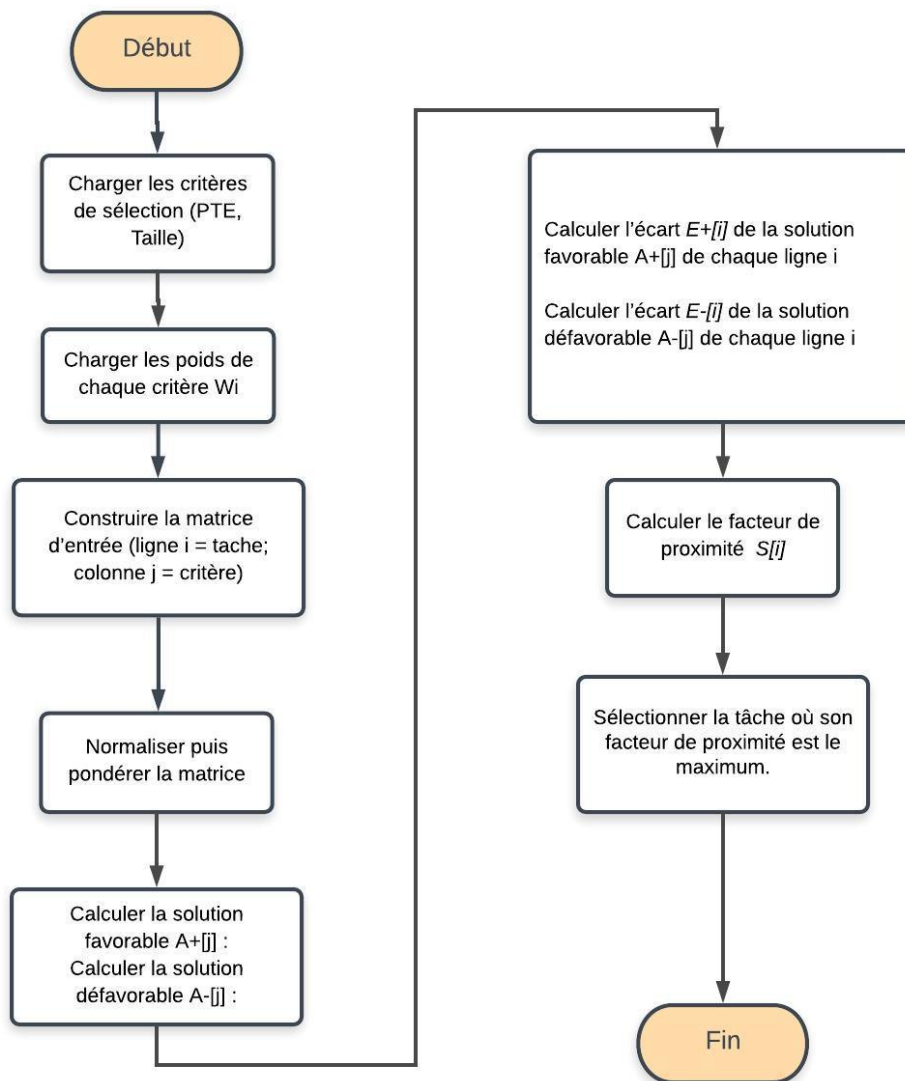


Figure 18 : Diagramme d'activité d'application de la méthode TOPSIS.[11]

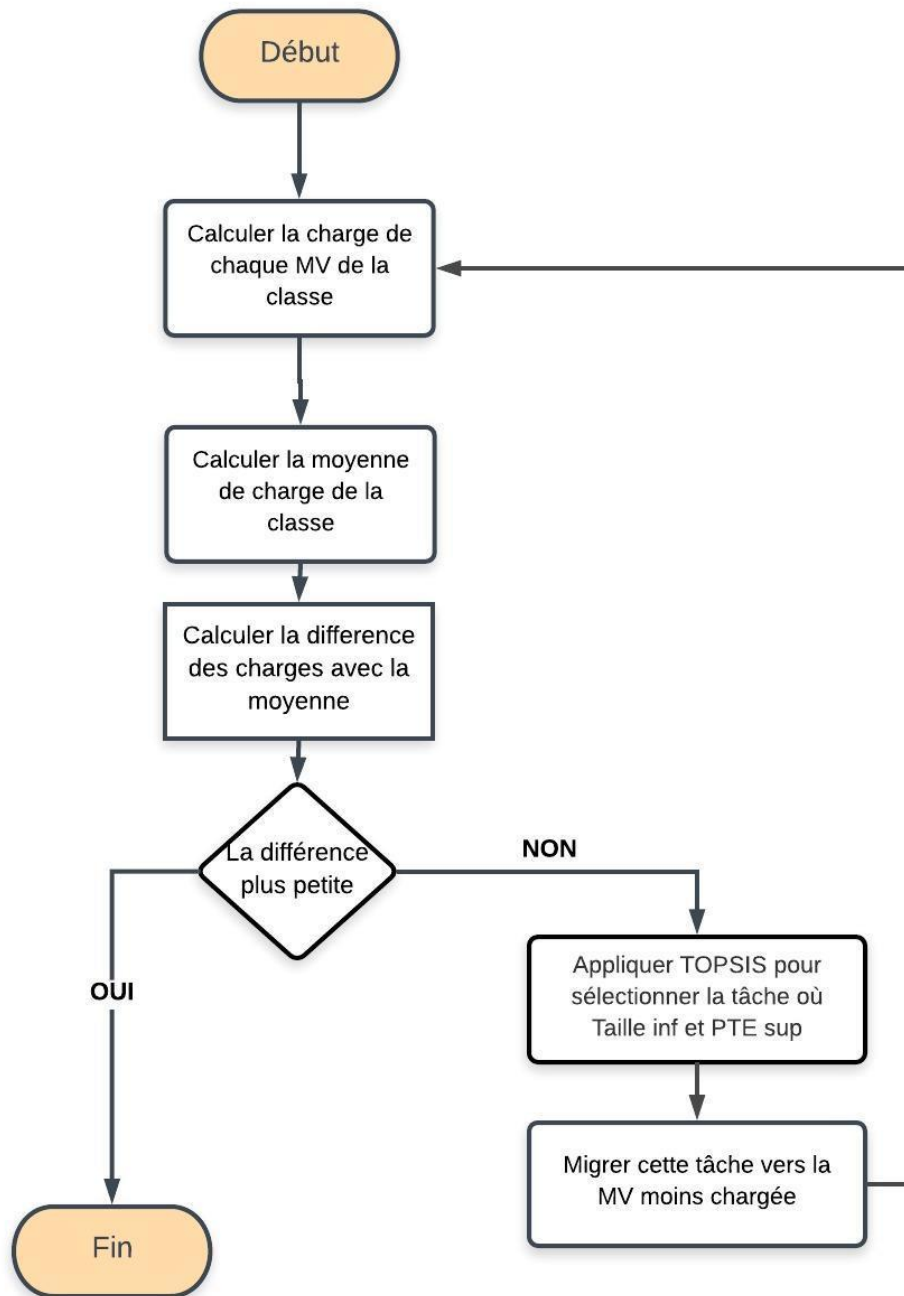


Figure 19 : Diagramme d'activité de l'équilibreur local.[11]

IV-4 : Equilibreur général

De la même façon que l'équilibreur local, l'équilibreur général calcul la moyenne de charge des classes afin de migrer les tâches des classes les plus chargées vers les classes les moins chargées. Dans ce contexte, nous pouvons distinguer deux cas. Le premier cas est lorsque la classe réceptrice est inférieure à la classe émettrice, dans ce cas la tâche candidate doit être choisie parmi celles ayant le pire temps d'exécution le plus petit. Dans le cas contraire, il choisira la tâche dont le pire temps d'exécution est le plus élevé.

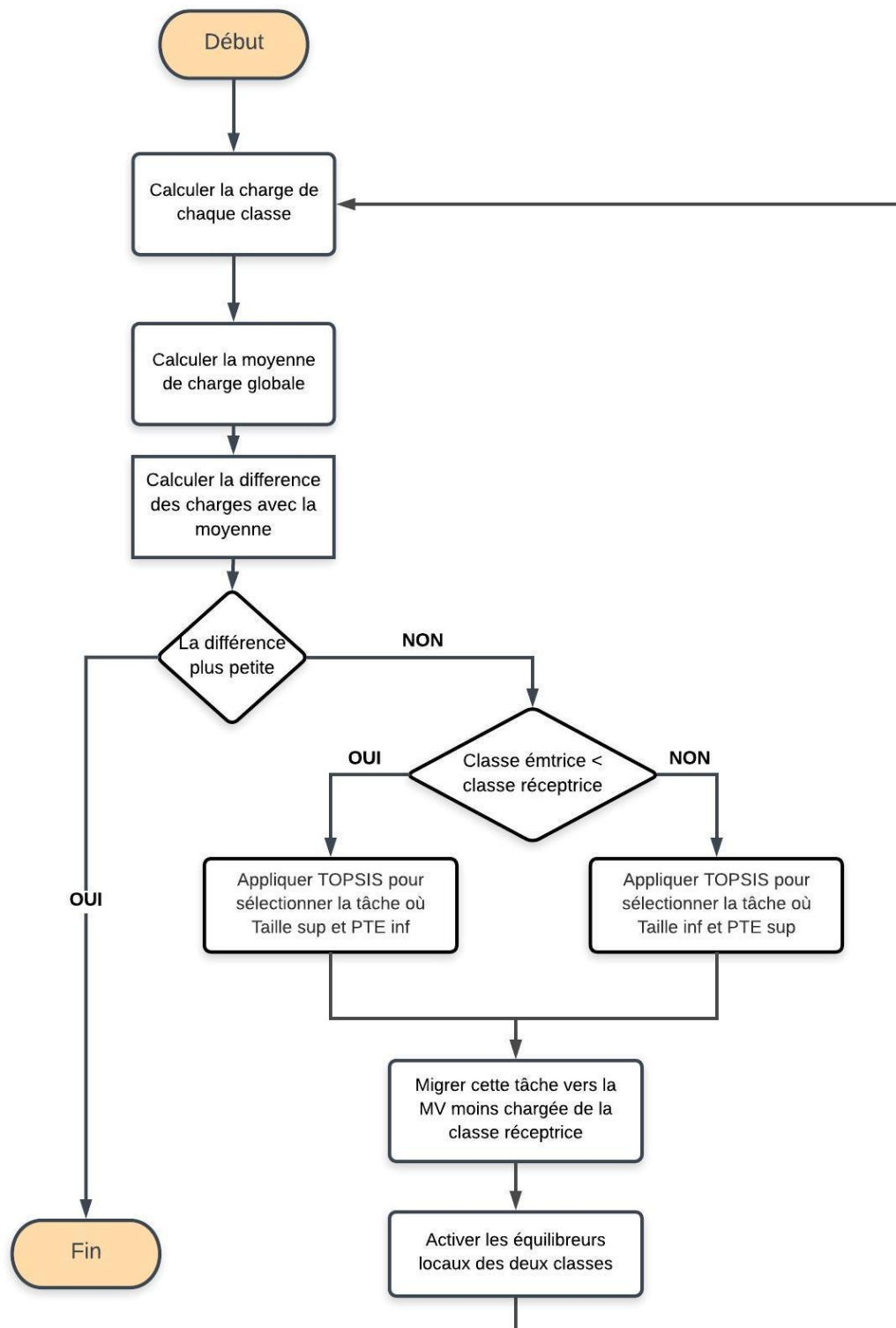


Figure 20 : Diagramme d'activité de l'équilibreur général.[11]

V- Conclusion

Dans ce chapitre, nous avons abordé le problème de déséquilibre de charge entre les MVs d'un CloudIdO, en proposant une architecture basée sur quatre composants essentiels. Le fonctionnement de notre modèle pour équilibrer la charge se déroule en trois phases principales : la première phase consiste à classifier les MVs en cinq classes, la deuxième phase est l'affectation des tâches aux classes et ceci en fonction des caractéristiques des tâches (PTE et Taille) et le centre de gravité des classes, quant à la troisième phase, elle permet d'équilibrer la charge intra-classe et inter-classe.

Dans le chapitre suivant, nous envisageons de valider notre modèle en effectuant des simulations dans l'environnement CloudSim afin de montrer la pertinence de notre modèle d'un côté, le temps nécessaire pour affecter les tâches aux classes, et bien d'autres paramètres qui nous permettront de ressortir les points positifs et négatifs de notre proposition.

Chapitre 4

Implémentation

I- Introduction

Dans le chapitre précédent, nous avons proposé une architecture de notre modèle basée sur le pire temps d'exécution des tâches et la classification des MVs, et afin d'illustrer les différentes idées et concepts inclus dans l'architecture proposée, nous allons déroulé les principaux aspects de notre architecture sur un exemple concret afin de montrer la faisabilité et la mise en évidence de nos idées.

Dans ce chapitre nous allons présenter, dans un premier temps, l'environnement du travail à savoir les différents langages de programmations ainsi l'environnement matériel et logiciel dont nous aurons besoin pour valider notre architecture à travers la simulation.

II- Définition de la simulation informatique¹

La simulation informatique, ou simulation numérique, est une série de calculs effectués sur un ordinateur et reproduisant un phénomène physique. Elle aboutit à la description du résultat de ce phénomène, comme s'il s'était réellement déroulé. Cette représentation peut être une série de données, une image ou même un film vidéo. Un simulateur peut réagir à des modifications de paramètres et modifier ses résultats en conséquence.

III-Langage et environnement de développement

III-1 : Langage de programmation java :

Java est un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable créé par James Gosling et Patrick Naughton employés de Sun Microsystems avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.²

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels que Unix, Microsoft Windows, Mac OS ou Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Java permet de développer des applications autonomes mais aussi, et surtout, des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des

¹<https://www.futura-sciences.com/tech/definitions/informatique-simulation-informatique-11319/> (17/06/2018)

²http://www.comoria.com/3516/Java_%28langage%29. (17/06/2018)

applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

III-2 : CloudSim

CloudSim est un simulateur qui permet la modélisation, la simulation et l'expérimentation de nouvelles Cloud Computing infrastructures et de services d'application. Dans le cas du Cloud Computing, les outils de Simulations comme CloudSim donnent d'importants avantages aux clients et fournisseurs. Pour les clients, il leurs permet de tester leurs services dans un environnement contrôlable avec exempt du coût et de vérifier la performance avant de les publier sur les Clouds. Pour les fournisseurs, il leurs permet de vérifier les types de location en fonction de divers prix et la charge.[77]

Sans ces outils, à la fois les clients et les fournisseurs doivent s'appuyer sur des évaluations imprécises, ou sur des approches essai-erreur, ces approches peuvent conduire à l'inefficacité de la performance des services. En outre, CloudSim aide les chercheurs et développeurs à tester la performance d'un service d'application développée.

Il ya de nombreux avantages relatives à l'utilisation du CloudSim, comme : (i) l'efficacité de temps : il prend très moins de temps et d'efforts pour mettre en œuvre des applications de Cloud Computing et (ii) la flexibilité : les développeurs peuvent facilement modéliser et tester les performances de leurs applications et ses services dans des environnements hétérogènes (Microsoft Azure, Amazon EC2). [77]

III-3 :L'architecture du CloudSim

La structure logicielle de CloudSim et ses composants est représentée par une architecture en couche comme le montre la Figure 21. Le niveau le plus bas est le moteur de simulation des événements discrets SimJava, qui implémente les fonctionnalités de base requises pour les cadres de simulation du niveau supérieur, telles que les files d'attente, le traitement des événements, création de composants du système (services, hôte, Datacenter, Broker, les machines virtuelles), la communication entre les composants et la gestion de l'horloge de simulation. [77]

CloudSim supporte la modélisation et la simulation de l'environnement de Datacenter basé sur le Cloud, tel que les interfaces de gestion dédiées aux VMs, la mémoire, le stockage et la bande passante. La couche CloudSim gère l'instanciation et l'exécution des entités de base (VM, hôtes, Datacenters, applications) au cours de la période de simulation. Dans la couche la plus haute de la pile de simulation, on trouve le code de l'utilisateur qui expose la configuration des fonctionnalités liées

aux hôtes (ex: nombre de machines, leurs spécifications), les politiques d'ordonnancement de Broker, applications (ex: nombre de tâches et leurs besoins), VM, nombre d'utilisateurs. [77]

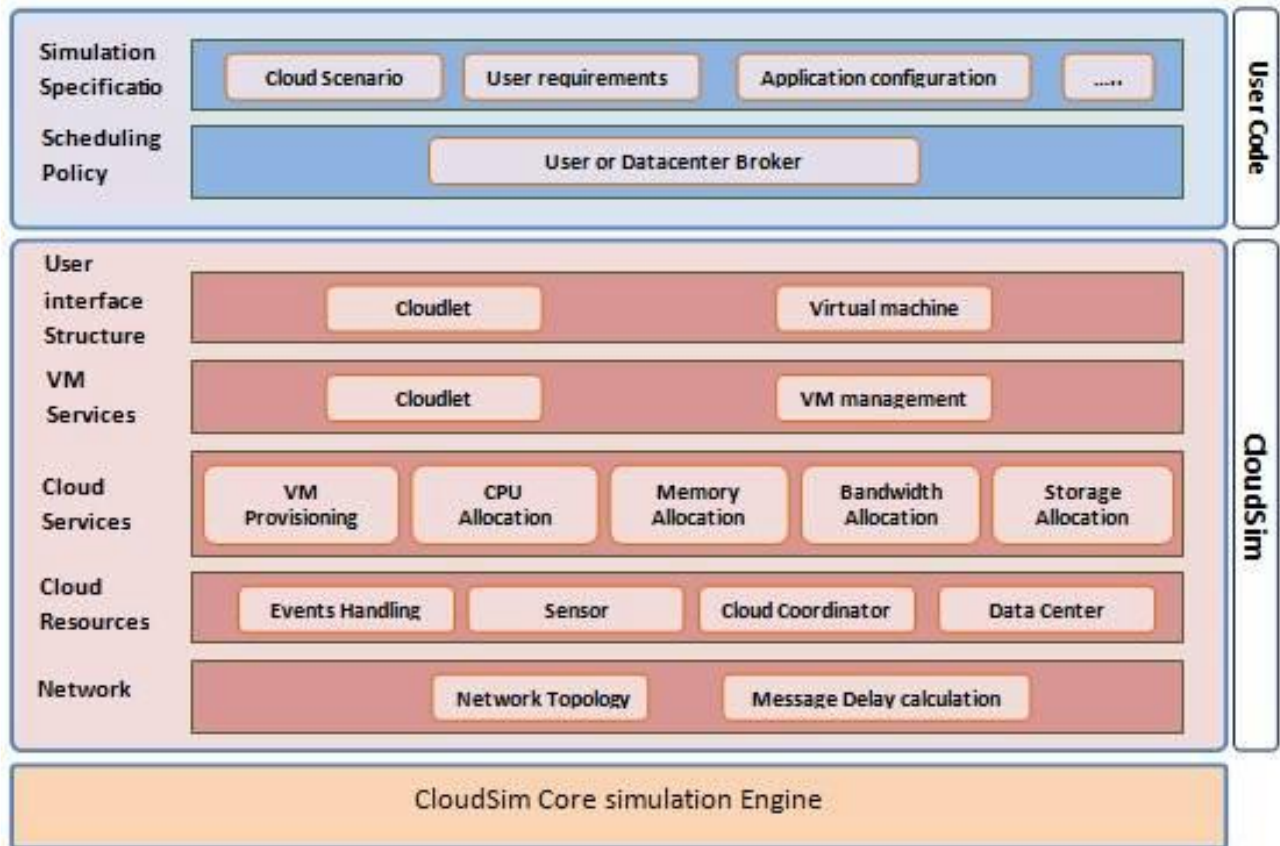


Figure 21 : Les couches de l'architecture du CloudSim. [61]

III-3-1 : Cloudlet

Cette classe modélise les services d'application de basés du Cloud (tels que la livraison, la gestion des réseaux sociaux, et le déroulement des opérations métier de l'entreprise) qui sont généralement déployer dans des Datacenter. CloudSim orchestre la complexité de la demande en fonction de ses exigences de calcul. Il représente la complexité d'une application en termes de ces besoins informatiques (taille en nombre d'instruction, temps de réponse.). Cette classe peut aussi être prorogé pour soutenir la modélisation d'autres mesures de performance et de composition pour des applications telles que les transactions dans les applications orientées base de données. [77]

III-3-2 : Virtual Machine [77]

Cette classe modéliser une instance de virtuelle machine (VM), qui est gérée et hébergée pendant son cycle de vie par le composant Cloud host, un host peut simultanément instancier de multiples VMs et assigner des cœurs de processeur (espace partagé, temps partagé).

III-3-3 : DataCenter

Cette classe modélise l'infrastructure du noyau du service (matériel, logiciel) offert par des fournisseurs de ressources dans un environnement de Cloud Computing. Il encapsule un ensemble de machine de calcul qui peut être homogènes ou hétérogènes en ce qui concerne leurs configurations de ressources (mémoire, noyau, capacité et stockage). En outre, chaque composant de datacenter instancie un composant d'approvisionnement de ressource qui implémente un ensemble de politiques d'allocation de bande passante, de mémoire et des dispositifs de stockage. L'HOST représente un serveur informatique physique dans un Cloud, il exécute des actions liées à la gestion des machines virtuelles et a une politique définie pour l'approvisionnement en mémoire et bande passante, ainsi que d'une politique de répartition des PE (élément de traitement) sur des machines virtuelles. Un hôte est associé à un DataCenter, et peut héberger des machines virtuelles. [77]

III-3-4 : DataCenterBroker

Cette classe modéliser le courtier (Broker), qui est responsable de la médiation entre les utilisateurs et les prestataires de service selon les conditions de QoS des utilisateurs, et il déploie les tâches de service à travers les Clouds. [77]

Le Broker agissant au nom des utilisateurs, identifie les prestataires de service appropriés du Cloud par le service d'information du Cloud CIS (Cloud Information Services) en négociant avec eux pour une allocation des ressources qui répondent aux besoins de QoS des utilisateurs. [77]

IV- Description de l'application « SAS Balancer »:

La version de CloudSim 3.0.3 n'a pas d'interface graphique, il utilise le mode console pendant l'exécution, qui rend difficile à l'utilisateur de profiter pleinement du simulateur, tel que la modification des paramètres de simulation, l'affichage graphique des outputs (résultat de simulation). Ainsi, comme première étape, nous avons développé une interface graphique qui facilite l'accès et la manipulation du simulateur, puis dans la deuxième étape, nous avons implémenté notre approche par l'enrichissement des classes du simulateur CloudSim, et enfin, nous lançons la simulation afin d'obtenir des résultats et de valider notre approche.

Notre application, appelé « SAS_Balancer », est composée d'une interface principale, qui contient quatre boutons dont le rôle de chacun d'eux est, respectivement, la création des machines virtuelles, la création des taches (cloudlet), la simulation pas-à-pas qui permet de détailler les étapes de la simulation: en commençant par la classification, le routage, l'équilibrage local (exemple d'une classe), et enfin l'équilibrage général, et le dernier bouton permet de voir les résultats globaux de la

simulation, c'est-à-dire les graphes des charges (locales et générales) avant et après l'opération de l'équilibrage, ainsi que, les graphes des temps d'exécution comme le montre la Figure 22.



Figure 22 : L'interface du SAS_Balancer.

Pour créer les machines virtuelles et les tâches, il suffit de cliquer sur les deux boutons à gauche respectivement, comme le montre la figure 23.

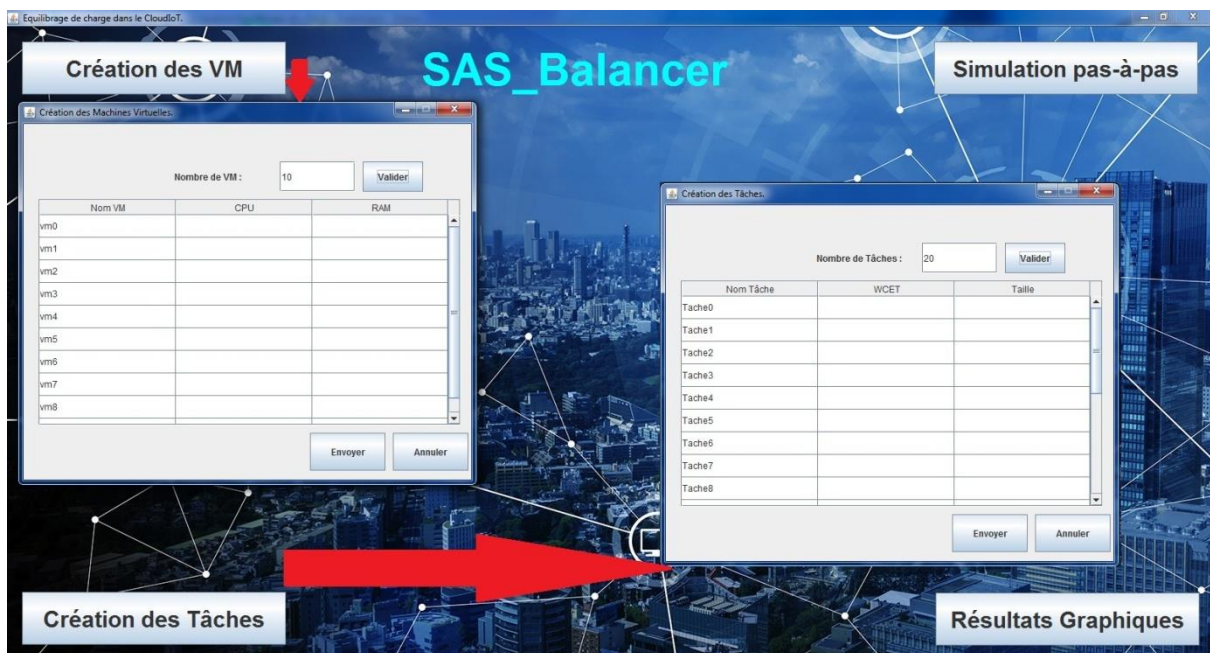


Figure 23 : Création des machines virtuelles et les tâches.

Pour passer à la simulation pas-à-pas, nous avons implémenté un bouton, qui permet d'afficher les trois étapes principales de la simulation: (1) la classification et le routage, (2) équilibrage local

(exemple la classe TR) et enfin (3) l'équilibrage général. Pour cela, il suffit de manipuler les étapes par les deux boutons suivant et précédent, comme le montre les figures 24, 25 et 26 respectivement.

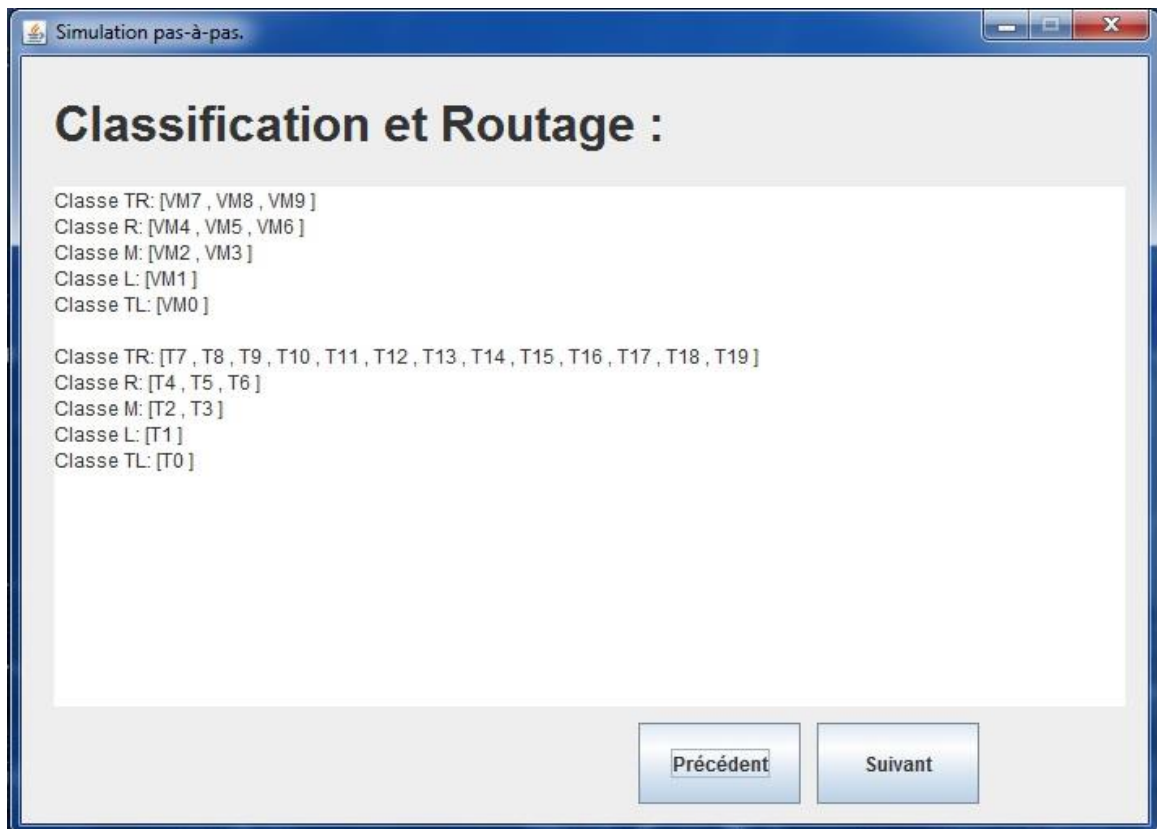


Figure 24 : L'étape de la simulation pas-à-pas (Classification et Routage).

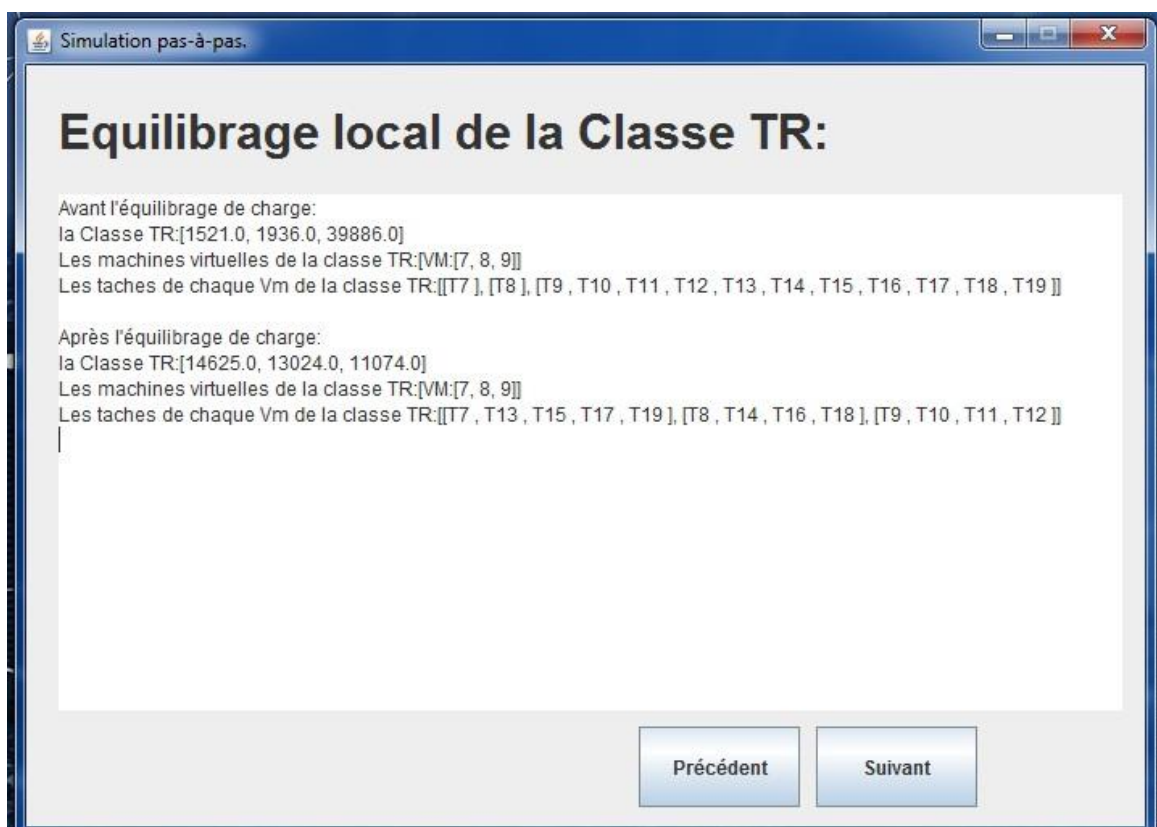


Figure 25 : L'étape de la simulation pas-à-pas (Equilibrage local « TR»).

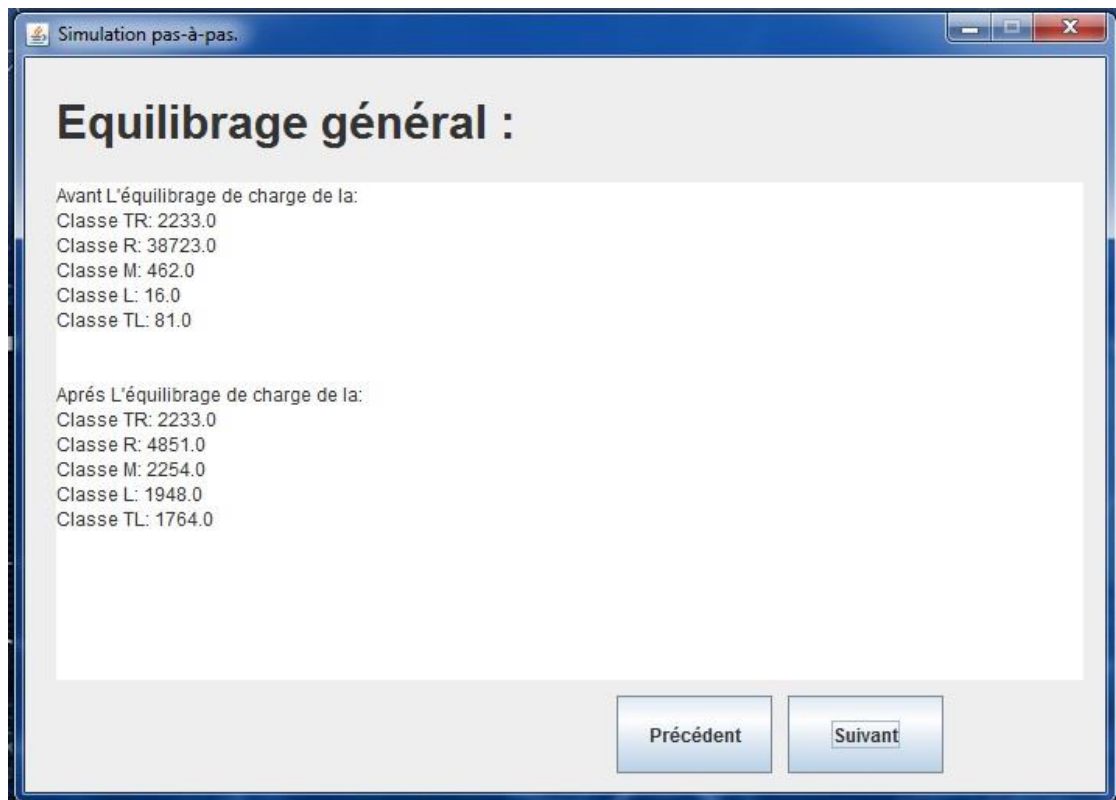


Figure 26 : L'étape de la simulation pas-à-pas (Equilibrage général).

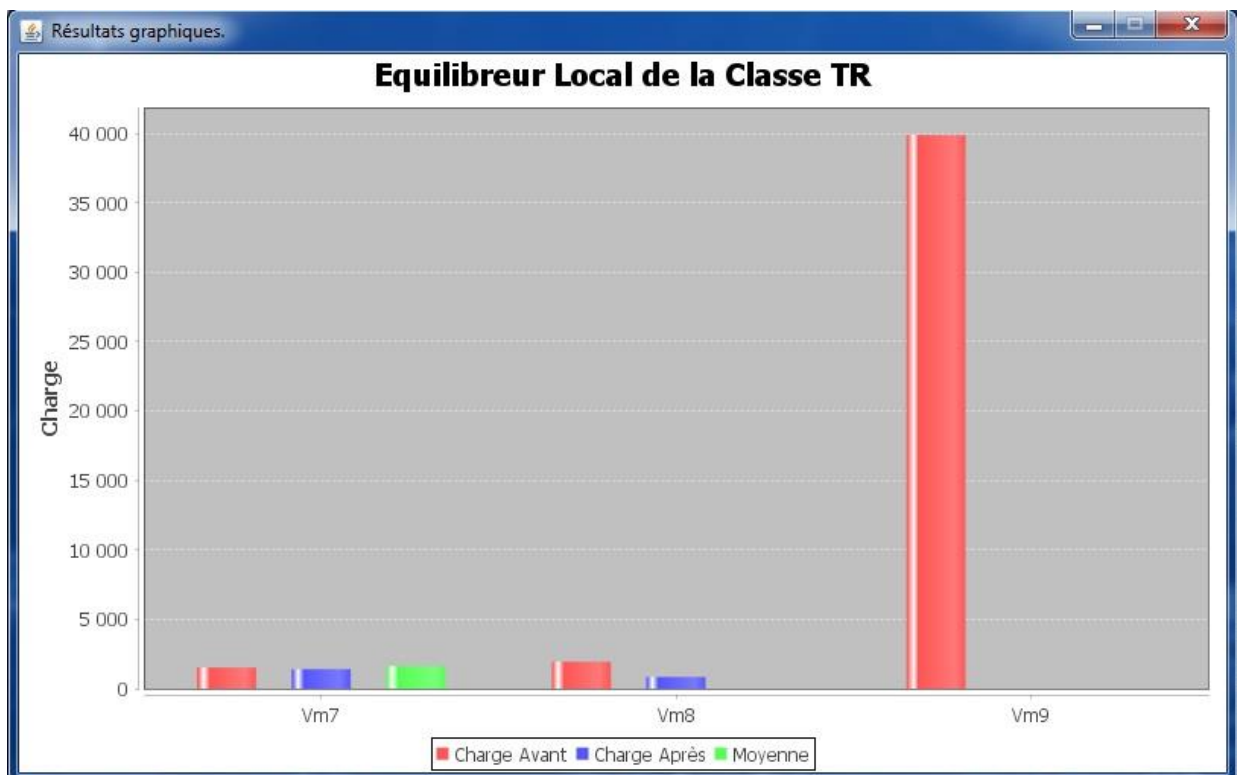


Figure 27 : Les résultats graphiques (Equilibrage local « TR »).

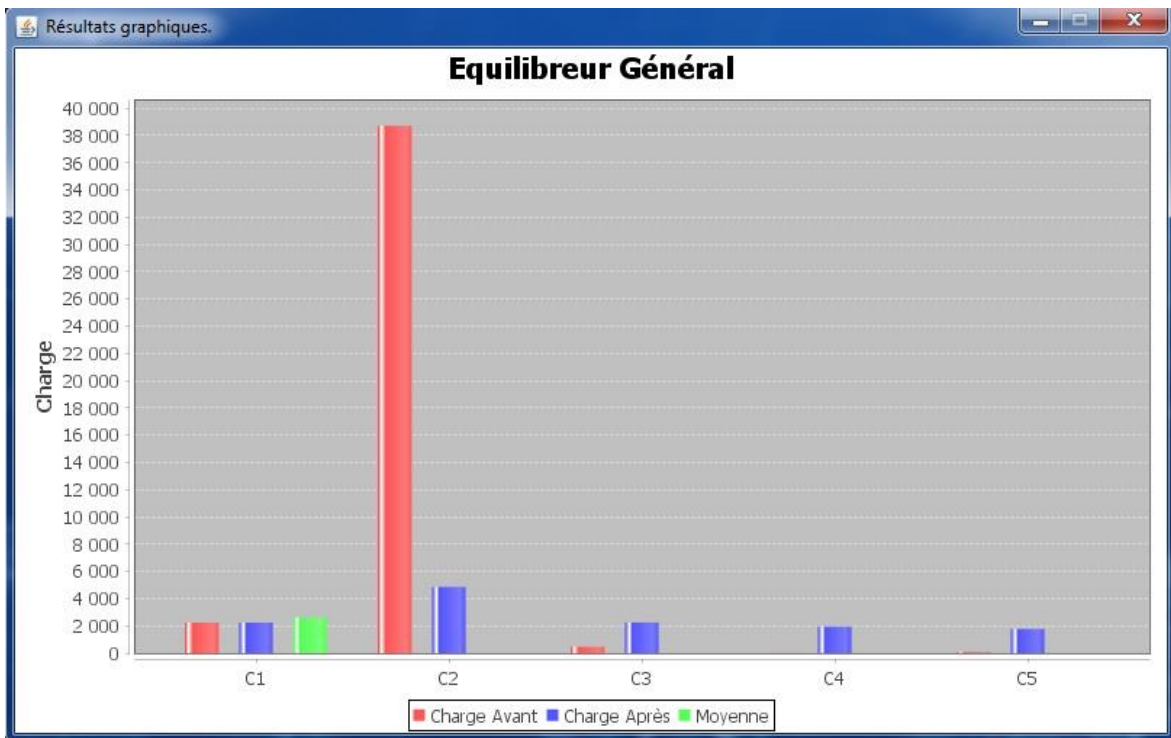


Figure 28 : Les résultats graphiques (Equilibrage général).

Les figures 29 et 30 montrent l'impact du changement du nombre de tâches et du changement du nombre de machines virtuelles sur le temps d'équilibrage de charge et de réponse. Pour cela, dans le premier scénario nous avons fixé le nombre de machines virtuelles à 10 et nous variant le nombre de tâches entre 20 et 4000, puis nous observons les résultats de la simulation comme l'indique le tableau N°4 et la figure 29.

Tableau 4: Liste des MVs et Tâches du scénario N°1.

VM	Tâche	Temps(MS)
10	20	70
10	30	88
10	50	89
10	70	142
10	120	262
10	250	353
10	500	554
10	800	1084
10	1000	1581
10	1500	47362
10	4000	79697

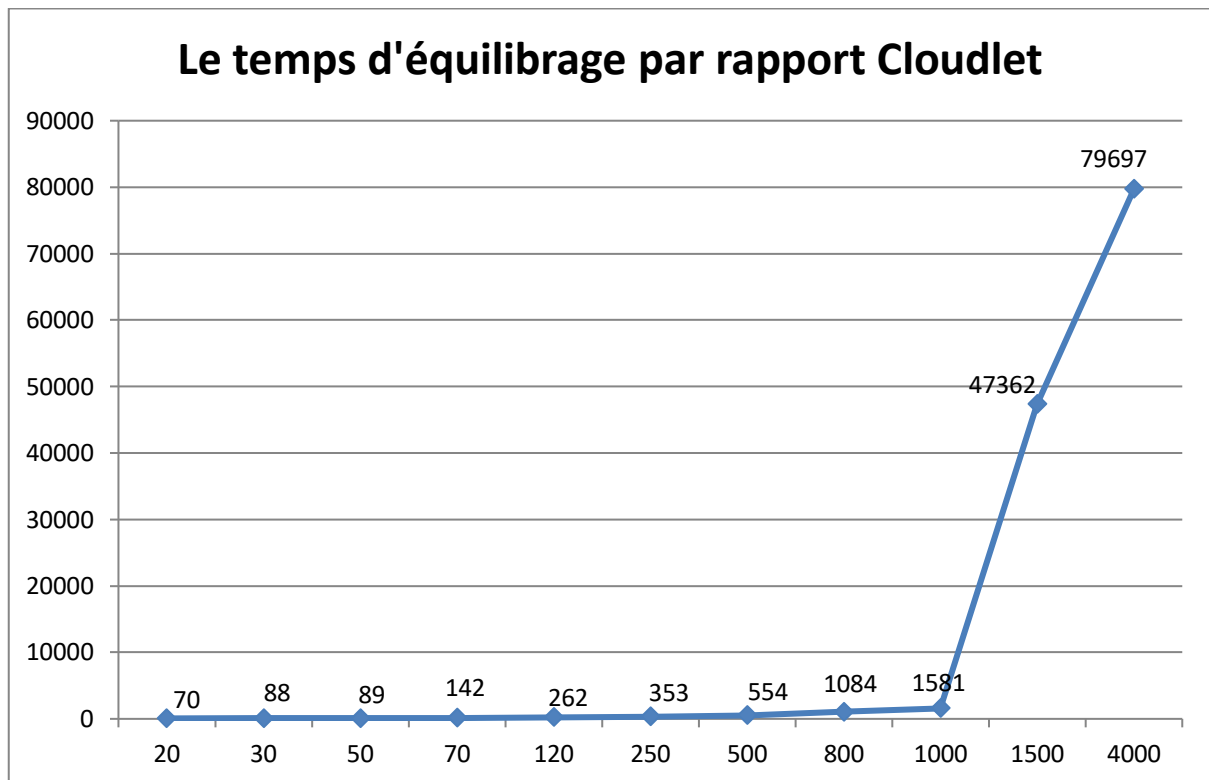


Figure 29 : Le temps d'équilibrage par rapport aux tâches.

Le deuxième scénario est représenté dans la figure 30. Ainsi, à travers ce scénario nous avons fixé le nombre de tâches à 2000, et on a varié le nombre de MVs de 10 à 1700 selon le tableau 5.

Tableau 5: Liste des MV et Tâches du scénario N°2.

Tâche	VM	Temps
2000	10	22612
2000	100	27372
2000	300	149047
2000	500	131045
2000	700	104723
2000	900	91492
2000	1100	85494
2000	1300	69261
2000	1500	49217
2000	1700	42065

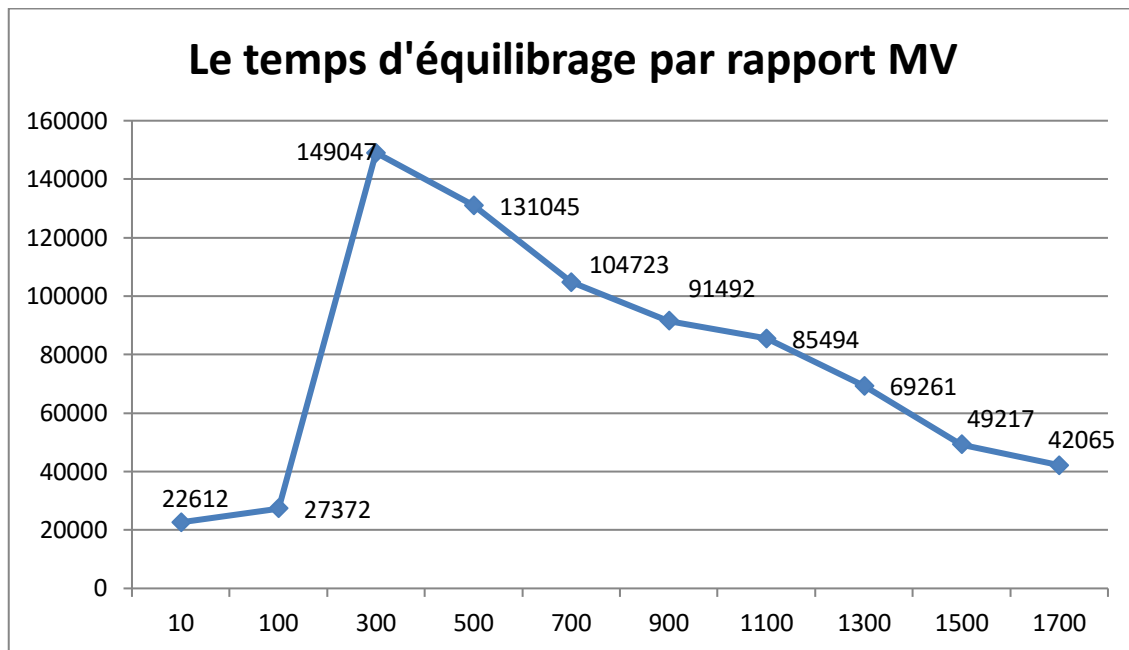


Figure 30 : Le temps de réponse par rapport aux MV.

Selon la figure 29, on remarque bien que le temps d'équilibrage augmente quand le nombre de tâches augmente. Par contre dans la figure 30 on constate que le temps de réponse commence à diminuer lorsque le nombre de MVs au environ de $\frac{1}{4}$ du nombre des tâches, parce que le temps d'équilibrage est réduit quand le système possède un nombre très petit de MV (le temps d'affectation), et après le $\frac{1}{4}$ du nombre de tâches, le temps de réponse diminue à cause du nombre de MVs qui augmente par contre le temps de l'équilibrage de charge augmente.

V- Conclusion

Nous avons présenté dans ce chapitre les différentes fonctions de notre Application et son mode d'utilisation, nous sommes intéressés dans notre projet de fin d'étude par le domaine de l'équilibrage de charge dans le CloudIoT, le but de notre solution proposée consiste à créer des classes de machine virtuelles en fonction de leur performance en utilisant pour cela la méthode K-MEANS d'une part, et d'équilibrer la charge entre les MV à l'intérieur de chaque, et aussi d'équilibrer la charge entre les classes.

Nous avons validé notre approche en mettant en œuvre les méthodes et les algorithmes proposés dans le chapitre 3. Les résultats obtenus montrent que notre approche permet ainsi effectivement un équilibrage de charge entre les différents MV dans le contexte du CloudIoT.

Conclusion

Générale

I/- Conclusion générale

Le CloudIdO est l'un des paradigmes computationnels les plus populaires et les plus intéressants, en raison de ses nombreux avantages, qui ont été explorés dans les dernières années. Il souffre de plusieurs problèmes dont le problème de déséquilibre de charge, où on a essayé de régler le dernier en proposant et développant notre modèle **SAS_Balancer**.

SAS_Balancer permet d'assurer l'équilibrage de charge entre les différentes machines virtuelles d'un CloudIdO, il est composé de quatre composants essentiels :

Classifieur : qui permet de classer les machines virtuelles selon la vitesse du CPU et la taille de RAM, en cinq classes nommées respectivement (Très rapide, Rapide, Moyenne, Lente, Très lente).

Routeur : qui permet de calculer les distances entre la tâche reçue et les centres de gravité des cinq classes, puis il envoie la tâche à la classe la plus proche par rapport aux distances précédemment calculées.

Équilibreur local : permet d'équilibrer la charge entre les différentes Machines Virtuelles, en migrant les tâches entre les MVs, appartenant à la même classe, tout en prenant en considération le pire temps d'exécution (PTE) d'une tâche et sa taille.

Équilibreur général : chargé d'équilibrer la charge entre les classes en migrant les tâches d'une classe surchargée vers une classe la moins chargée.

Les résultats obtenus à travers la simulation montrent que notre proposition permet effectivement l'équilibrage de charge entre les différentes Machines Virtuelles d'un CloudIdO. Certes, le travail réalisé couvre une partie des problématiques du déséquilibre de charge, ce qui reste à faire comme futur travail sera présenté dans la section suivante.

II/- Perspectives

Ce projet de fin d'étude, reste un thème ouvert pour les futurs étudiants. En effet, nous suggérons que l'opération de l'équilibrage de charge soit totalement décentralisée. Pour cela, nous proposons l'amélioration de notre approche par l'ajout des agents intelligents qui communiquent entre eux de façon totalement décentralisée d'un côté, et qui prévoient le déséquilibre du système afin de le régler avant son apparition.

Références

Bibliographiques

Références bibliographiques

- 1 Aazam, M., Huh, E.-N., (2014). Fog computing and smart gateway-based communication for cloud of things. In: Future Internet of Things and Cloud (FiCloud), pp. 464–470.
- 2 Abdelwahab, S., Hamdaoui, B., Guizani, M., Rayes, A., (2014). Enabling smart cloud services through remote sensing: An internet of everything enabler. *Internet of Things Journal, IEEE* 1 (3), 276–288.
- 3 Aceto, G., Botta, A., de Donato, W., Pescap`e, A., (2013). Cloud monitoring: A survey. *Computer Networks* 57 (9), 2093–2115.
- 4 Alakeel, A.M. (2010). A Guide to Dynamic Load Balancing in Distributed Computer Systems. *International Journal of Computer Science and Network Security*, 10 (6): 153-160.
- 5 Alessio Botta, Walter de Donato, Valerio Persico, and Antonio Pescap_e. (2014), On the integration of cloud computing and internet of things. In Future Internet of Things and Cloud (FiCloud), pages 23,30.
- 6 Alexander Gluhak, SrdjanKrcro, Michele Nati, Dennis Pfisterer, Nathalie Mitton, and TahiryRazafindralambo, (2011), A survey on facilities for experimental internet of things research. *Communications Magazine, IEEE*, 49(11):58,67.
- 7 Alhakbani, N., Hassan, M. M., Hossain, M. A., Alnuem, M., (2014). A framework of adaptive interaction support in cloud-based internet of things (iot) environment. In: *Internet and Distributed Computing Systems*. Springer, pp. 136–146.
- 8 Al-Rayis, E. et H. Kurdi (2013). Performance Analysis of load balancing architectures in cloud computing. In: *IEEE European Modeling Symposium, Manchester, UK*.
- 9 Atkins et al., C., (2013). A Cloud Service for End-User Participation Concerning the Internet of Things. In: *Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 273–278.
- 10 Ballon, P., Glidden, J., Kranas, P., Menychtas, A., Ruston, S., Van Der Graaf, S., (2011). Is there a need for a cloud platform for European smart cities? In: *eChallenges e-2011 Conference Proceedings, IIMC International Information Management Corporation*.
- 11 Benabbes, S., A, Necib, S, M, Hemam, (2018). Vers un nouveau modèle d'équilibrage de charge dans le CloudIoT, ASD2018, Maroc, pp :601-614.
- 12 Cardellini, V., M. Colajanni, et P.S. Yu (1999). Dynamic Load Balancing On Web-server Systems. *IEEE Internet Computing*, 10: 28.39.
- 13 Chen, S.-Y., Lai, C.-F., Huang, Y.-M., Jeng, Y.-L., (2013). Intelligent home-appliance recognition over IoT cloud network. In: *Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 639–643.

Références bibliographiques

- 14 Chien, N.K., et H.D. Loc (2016). Load balancing algorithm based on estimating finish time of services in cloud computing. In: 18th International Conference on Advanced Communication Technology, Pyeongchang, South Korea.
- 15 Christophe, B., Boussard, M., Lu, M., Pastor, A., Toubiana, V., (2011). The web of things vision: Things as a service and interaction patterns. *Bell Labs Technical Journal* 16 (1), 55–61.
- 16 Corsar, D., Edwards, P., Velaga, N., Nelson, J., Pan, J., (2011). Short paper: addressing the challenges of semantic citizen-sensing. In: 4th International Workshop on Semantic Sensor Networks, CEUR-WS. pp.90–95.
- 17 Distefano, S., Merlino, G., Puliafito, A., (2012). Enabling the cloud of things. In: *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pp. 858–863.
- 18 Dobre, C., Xhafa, F., (2014). Intelligent services for big data science. *Future Generation Computer Systems* 37, 267–281.
- 19 Doukas, C., Maglogiannis, I., (2012). Bringing iot and cloud computing towards pervasive healthcare. (IMIS), pp. 922–926.
- 20 European Commission. (2013), Definition of a research and innovation policy leveraging Cloud Computing and IoT combination. Tender specification, SMART 2013/0037.
- 21 Evans, D., (2012). The internet of everything: How more relevant and valuable connections will change the world. Cisco IBSG, 1–9.
- 22 Fahim, Y., E.H. Benlahmar, L. Elhoussine, et A. Eddaoui (2016). Une nouvelle conception d'équilibrage de charge dans le Cloud Computing. In: 4^{ème} Journée sur les Technologies d'Information et de Modélisation, Casablanca, Morocco.
- 23 Fischer, A., (2014). Deux méthodes d'apprentissage non supervisé : synthèse sur la méthode des centres mobiles et présentation des courbes principale. *Journal de la Société Française de Statistique*, 155(2): 2-35.
- 24 Fox, G. C., Kamburugamuve, S., Hartman, R. D., 2012. Architecture and measured characteristics of a cloud based internet of things, pp. 6–12.
- 25 Gachet, D., de Buenaga, M., Aparicio, F., Padrón, V., (2012). Integrating internet of things and cloud computing for health services provisioning: The virtual cloud carer project. (IMIS), pp. 918–921.
- 26 Gao, F., (2013). VSaaS Model on DRAGON-Lab. *International Journal of Multimedia & Ubiquitous Engineering* 8 (4).
- 27 Gerhard Nussbaum, (2006). People with disabilities: assistive homes and environments. In *Computers Helping People with Special Needs*, pages 457,460.

Références bibliographiques

- 28 Gubbi, J., Buyya, R., Marusic, S., Palaniswami, M., (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29 (7), 1645–1660.
- 29 Han, D.-M., Lim, J.-H., (2010). Smart home energy management system using IEEE 802.15. 4 and zigbee. *Consumer Electronics, IEEE Transactions on* 56 (3), 1403–1410.
- 30 Hank, P., M'uller, S., Vermesan, O., Van Den Keybus, J., (2013). Automotive ethernet: in-vehicle networking and smart mobility. In: *Proceedings of the Conference on Design, Automation and Test in Europe*. EDA Consortium, pp. 1735–1739.
- 31 Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelffle, (2010). *Vision and Challenges for Realising the Internet of Things*. Luxembourg.
- 32 He, W., Yan, G., Xu, L. D., (2014). Developing vehicular data cloud services in the iot environment. *Industrial Informatics, IEEE Transactions on* 10 (2), 1587–1595.
- 33 Hemam, S.M., O., Hioual, et O., Hioual (2017). Load balancing between nodes in a volunteer Cloud Computing by taking into consideration the number of Cloud services replicas. In: *3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco.
- 34 HENNANE Hemza, BELBACHIR Iméne, (2015), *Equilibrage de Charge dans Les Systèmes Distribués*, Université Abou Bakr Belkaid– Tlemcen, thèse de Master.
- 35 HORIZON 2020 WORK PROGRAMME 2014-2015., (2014). *Industrial leadership. leadership in enabling and industrial technologies*, LEIT Information and Communication Technologies.
- 36 Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami, (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645,1660.
- 37 Jeffery, K., (2014). Keynote: CLOUDS: A large virtualisation of small things. In: *The 2nd International Conference on Future Internet of Things and Cloud*.
- 38 Ji, Y.K, Kim, Y.I., Park, S., (2014). Big data summarization using semantic feture for iot on cloud, *Contemporary Engineering Sciences*, Vol. 7, no. 22, 1095 - 1103.
- 39 Kapoor, S., et C. Dabas, (2015). Cluster based load balancing in cloud computing. In: *Eighth International Conference on Contemporary Computing (IC3)*, Noida, India.
- 40 Kumar, N., (2013). Smart and intelligent energy efficient public illumination system with ubiquitous communication for smart city, *(ICSSS)*, pp. 152–157.
- 41 Kuo, A. M.-H., (2011). Opportunities and challenges of cloud computing to improve health care services. *Journal of medical Internet research* 13 (3).

Références bibliographiques

- 42 Lazarescu, M., (2013). Design of a wsn platform for long-term environmental monitoring for iot applications. *Emerging and Selected Topics in Circuits and Systems*, IEEE, 45–54.
- 43 Lee, K., Murray, D., Hughes, D., Joosen, W., (2010). Extending sensor networks into the cloud using amazon web services. (NESEA), pp. 1–7.
- 44 Li, W., Zhong, Y., Wang, X., Cao, Y., (2013). Resource virtualization and service selection in cloud logistics. *Journal of Network and Computer Applications* 36 (6), 1696–1704.
- 45 Löhr, H., Sadeghi, A.-R., Winandy, M., (2010). Securing the e-health cloud. In: *Proceedings of the 1st ACM International Health Informatics Symposium*. ACM, pp. 220–229.
- 46 Luigi Atzori, Antonio Iera, and Giacomo Morabito, (2010). The internet of things: A survey. *Computer networks*, 54(15):2787-2805.
- 47 Madsen, H., Albeanu, G., Burtschy, B., Popentiu-Vladicescu, F., (2013). Reliability in the utility computing era: Towards reliable fog computing, (IWSSIP), pp. 43–46.
- 48 Mell, P., et T. Grance (2009). The NIST definition of cloud computing, National Institute of Standards and Technology, version 15.
- 49 Misic, V., Misic, J., (2014). *Machine-to-machine communications: Architectures, standards and applications*.
- 50 Mitton, N., Papavassiliou, S., Puliafito, A., Trivedi, K. S., (2012). Combining Cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*, 1–10.
- 51 Nkosi, M., Mekuria, F., (2010). Cloud computing for enhanced mobile health applications. In: *Cloud Computing Technology and Science (CloudCom)*, IEEE.
- 52 Olejnik, R., I. Alshabani, B. Toursel, E. Laskowski, et M. Tudruj (2009). Load balancing metrics for the SOAJA framework. *Scientific International Journal for Parallel and Distributed Computing, Scalable Computing: Practice and Experience*, 10 (4): 1-10.
- 53 Parwekar, P., (2011). From Internet of Things towards cloud of things. In: *Computer and Communication Technology (ICCCT)*, pp. 329–333.
- 54 Pereira, P. P., Eliasson, J., Kyusakov, R., Delsing, J., Raayatinezhad, A., Johansson, M., (2013). Enabling cloud connectivity for mobile internet of things applications. In: *Service Oriented System Engineering (SOSE)*, IEEE.
- 55 Peter Mell and Tim Grance, (2009). The NIST definition of cloud computing. *National Institute of Standards and Technology*, 53(6):50.
- 56 Petrolo, R., Loscr`i, V., Mitton, N., (2014). Towards a smart city based on cloud of things. In: *Proceedings of the 2014 ACM international workshop on Wireless and mobile technologies for smart cities*. ACM, pp. 61–66.

Références bibliographiques

- 57 Pierre-Jean Benghozi, Sylvain Bureau, Françoise Massit-Folea, (2008), L'Internet des objets. Quels enjeux pour les Européens ? Rapport de la chaire Orange "innovation and regulation", Ecole polytechnique et TELECOM Paris Tech.
- 58 Prati, A., Vezzani, R., Fornaciari, M., Cucchiara, R., (2013). Intelligent Video Surveillance as a Service. In: Intelligent Multimedia Surveillance. Springer, pp. 1–16.
- 59 Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N Calheiros, (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Algorithms and architectures for parallel processing, pages 13.
- 60 Rao, B. P., Saluia, P., Sharma, N., Mittal, A., Sharma, S. V., (2012). Cloud computing for Internet of Things & sensing based applications. In: Sensing Technology (ICST), 2012 Sixth International Conference on. IEEE, pp. 374–380.
- 61 Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose and Rajkumar Buyya, (2011). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software – Practice and Experience*, 41:23–50
- 62 Rolf H Weber, (2009). Internet of things {need for a new legal environment? *Computer law & security review*, 25(6):(522,527).
- 63 Sidhu, A.K., et S. King (2013). Analysis of Load Balancing Techniques in Cloud Computing. *International Journal of Computers & Technology*, 4(2): 737-741.
- 64 Simmhan, Y., Kumbhare, A. G., Cao, B., Prasanna, V., (2011). An analysis of security and privacy issues in smart grid software architectures on clouds. In: *Cloud Computing (CLOUD)*, IEEE, pp. 582–589.
- 65 Suciu, G., Vulpe, A., Halunga, S., Fratu, O., Todoran, G., Suciu, V., (2013). Smart Cities Built on Resilient Cloud Computing and Secure Internet of Things. In: *Control Systems and Computer Science (CSCS)*, pp. 513–518.
- 66 Tharam Dillon, Chen Wu, and Elizabeth Chang, (2010). Cloud computing: issues and challenges. (AINA), pages 27-33.
- 67 Wang, C., Bi, Z., Xu, L. D., (2014). Iot and cloud computing in automation of assembly modeling systems. *Industrial Informatics, IEEE Transactions on* 10 (2), 1426–1434.
- 68 Xu Li, Rongxing Lu, Xiaohui Liang, Xuemin Shen, Jiming Chen, and Xiaodong Lin, (2011), Smart community: an internet of things application. *Communications Magazine, IEEE*, 49(11):68-75.

Références bibliographiques

- 69 Yakhchi, S., S. M. Ghafari, M. Yakhchi, M. Fazeli, et A. Patooghy (2015). ICA-MMT: A load balancing method in cloud computing environment. In: 2nd World Symposium on Web Applications and Networking (WSWAN), Sousse, Tunisia.
- 70 Yezza, A. (2017). La méthode TOPSIS expliqué pas à pas. Rapport proposé Mais 2015, mis à jour Avril 2017.
- 71 Yun, M., Yuxin, B., (2010). Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid, (ICAEE), pp. 69–72.
- 72 Zaslavsky, A., Perera, C., Georgakopoulos, D., (2013). Sensing as a service and big data. arXiv preprint arXiv:1301.0159.
- 73 Zhao, F., (2010). Sensors meet the cloud: Planetary-scale distributed sensing and decision making. (ICCI), pp. 998–998.
- 74 Zhu, J., Chan, D., Prabhu, M., Natarajan, P., Hu, H., Bonomi, F., (2013). Improving web sites performance using edge servers in fog computing architecture, (SOSE), pp. 320–323.
- 75 Zikopoulos, P., Eaton, C., Deroos, D., Deutsch, T., Lapis, G.,(2011). Understanding big data: Analytics for enterprise class hadoop and streaming data. McGraw-Hill Osborne Media.
- 76 Zissis, D., Lekkas, D., Mar. (2012). Addressing cloud computing security issues. Future Gener. Comput. Syst. 28 (3), 583–592.
- 77 Lanani, S., (2015), Une approche BPM (Business Process Managment) par composition d'applications dans le cloud computing, Thèse de Magister, Université Mohamed Khider - Biskra.