

# Etude et mise en place d'Edge Node sur la base d'OpenEdgeComputing

HOANG Tuan Dung, KAF Merwan, LE CORRE Pierre

**Résumé**—Avec l'arrivée massive des objets connectés, la bande passante du réseau internet risque d'être saturé par la trop grande demande de ressources faites aux serveurs situés dans le Cloud. Afin de palier à ce problème, Orange Labs a proposé l'étude d'une solution existante : OpenEdgeComputing. Cette solution consiste à se servir de la puissance de calculs de machines situés en périphérie du réseau (mini datacenter, livebox, ...) afin de permettre aux objets connectés d'accéder à la puissance de calculs sans remonter jusqu'au serveurs situés dans le cloud. Dans un second temps, l'implémentation d'une nouvelle solution a été mis en oeuvre afin de palier aux problèmes de OpenEdgeComputing.

**Mots clés**—Cloudlet, Edge Node, Open Edge Computing, Open Stack, Cloud, Virtualisation, Zeroconf

## I. INTRODUCTION

Le projet confié par la société Orange s'inscrit dans l'anticipation de la multiplication future des objets connectés à internet. Ces derniers utilisent la puissance de calculs de serveurs distants très haut dans le réseau - que l'on appelle plus communément Cloud computing - afin d'effectuer divers traitements. Cependant, Orange estime que leurs nombres augmentera de façon tellement importante d'ici 5 à 10 ans que ces objets engorgeront le réseau ce qui altérera l'expérience des utilisateurs. Afin de palier à ce problème futur, Orange réfléchit à diverses manières d'éviter aux objets connectés d'effectuer leurs traitements très haut dans le réseau. (cf. figure 1)

Le concept d' Edge Computing permet de fournir de la puissance de calculs beaucoup plus bas dans le réseau et au plus proche des utilisateurs. L'intérêt est donc double, à la fois désengorger le réseau mais aussi diminuer la latence. Dans ce projet le but est d'évaluer une solution open source : OpenEdgeComputing et de la mettre en oeuvre en proposant des outils spécifiques au besoin d'Orange.

## II. CONTEXTE TECHNIQUE

Dans le cadre de ce sujet, il peut-être utile de définir ou de rappeler certaines notions indispensables à la compréhension du projet :

- Cloud computing : Ici la notion principale du cloud computing la plus pertinente pour ce projet est l'IAAS, traduction de "Infrastructure As A Service", elle consiste à vendre des serveurs à une entreprise tiers, ainsi, celle-ci n'a plus à s'occuper du matériel sur lequel ses logiciels fonctionnent et peut, à tout moment, décider d'augmenter sa capacité de traitement, ou au contraire, la diminuer
- Edge computing



FIGURE 1. Schéma du cloud actuel

- Cloudlet : Une Cloudlet peut être considéré comme une sorte de mini-datacenter situé en périphérie du réseau internet, ici la périphérie d'internet est définie comme une zone géographique plus proche des utilisateurs. Un exemple est l'installation potentielle d'une Cloudlet dans le même espace que les antennes 4G (ou bientôt 5G)
- OpenEdgeComputing : TODOTODOTODOTODO
- OpenStack : OpenStack est un ensemble de logiciels permettant de déployer un service de cloud-computing, que l'on nomme aussi IAAS (Infrastructure As A Service)

## III. MÉTHODE

Trois phases distinctes définissent ce projet :

- Recherche sur edge computing et plus spécifiquement sur le projet open source OpenEdgeComputing
- Mise en place du projet OpenEdgeComputing
- Mise en place d'une nouvelle solution

### A. Recherche sur OpenEdgeComputing

La première partie du projet consistait en l'étude des différents projets EdgeComputing, et plus précisément celui de OpenEdgeComputing. Nous avons étudié l'architec-

ture général des différents projets EdgeComputing (cf. figure représentant l'architecture).

En ce moment, il existe plusieurs notions dans le monde d'Edge Computing : Fog Computing, Mobile Edge Computing, Cloudlet, Micro datacenter. Ces notions sont prises en charge par des recherches menées par différentes organismes de recherche et d'entreprises. Cela explique pourquoi les notions Fog Computing, Mobile Edge Computing, Cloudlet, Micro datacenter sont à la fois ressemblantes et complémentaires.

### B. Mise en place de OpenEdgeComputing

Le projet qui se base en grande partie sur le projet OpenStack qui permet de gérer un ensemble de serveur afin de constituer un cloud utilisable et manageable a été mis en place après l'installation d'Openstack sur un serveur. En effet, le projet OpenEdgeComputing est en quelque sorte une surcouche d'Openstack simplifiant son utilisation avec des cloudlets en apportant quelques nouvelles fonctionnalités :

- La synthèse de machines virtuelles (VM Synthesis)
- Une gestion de la continuité de certaines machines virtuelles (VM handoff)
- Cartographie des Cloudlets pour sélectionner le plus proche (Discovery)

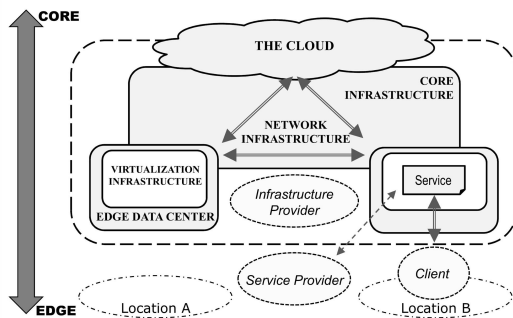


FIGURE 2. Image OpenEdge

La première partie consiste à installer Openstack Kilo, ensuite le projet OpenEdgeComputing possède plusieurs parties qui sont liées aux différentes fonctionnalités ajoutées. La première à installer est la synthèse de machines virtuelle, elle consiste en l'installation d'une version modifiée de qemu ainsi que d'outils permettant la gestion de la synthèse de machines virtuelles. La deuxième partie consiste en une modification de l'interface d'openstack, qui inclut désormais la gestion des cloudlets.

### C. Mise en place d'une nouvelle solution

Par la suite, une nouvelle solution a été implémenté après discussion de l'architecture générale de cette application avec M. LE TOQUIN. (cf. figure 3).

- Il y a deux parties distinctes dans cette architecture.
- La partie Openstack ;
  - La partie applicative.

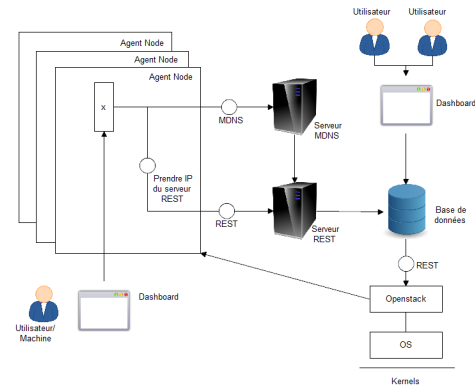


FIGURE 3. Schéma de l'architecture de l'application

1) *Langage, Python*: L'implémentation est faite avec le langage Python. Dans le cadre de ce projet, Python possède plusieurs avantages : simplicité de syntaxe, rapidité de développement. De plus de nombreuses API sont disponibles ce qui permet de développer la partie applicative plus efficacement dans un laps de temps court.

2) *Gestion du cloud, Agent OpenStack*: L'agent OpenStack permet une connection à une infrastructure Openstack et une gestion de certains processus de virtualisation afin de pouvoir utiliser la puissance des équipements sur lesquels cet agent est installé, comme un noeud nova-compute standard. Un noeud nova-compute permet d'exécuter des machines virtuelles sur un ordinateur, sous la supervision d'OpenStack. Cet agent est basé sur docker ainsi que sur ubuntu 10.04 : Ce choix est basé sur deux choses :

- Docker permet un déploiement rapide et sans dépendance logicielle autre que lui-même
- Ubuntu 14.04 est la version imposé par le projet OpenEdgeComputing, car celui-ci ne fonctionne que sur cette distribution linux

Le fonctionnement de cet agent est le suivant :

- Il va dans un premier temps récupérer l'adresse ip du serveur OpenStack qui lui correspond via l'agent node
- Puis il va s'y connecter pour pouvoir offrir sa puissance de calcul à l'infrastructure OpenStack

3) *Interface utilisateur, Agent Node*: L'interface utilisateur appelé Agent Node permet à un utilisateur de l'application de pouvoir venir mettre des ressources matérielles à disposition. Pour cela une interface listant tous les objets compatibles avec le cloud et OpenStack est à sa disposition. (cf. figure 4)

En venant cliquer sur le bouton "NOM DU BOUTON", l'utilisateur enregistre directement son appareil dans la base de donnée où se trouve toutes les ressources mises à disposition par Orange. Pour cela l'utilisation d'un serveur REST permet un enregistrement simple des différentes ressources et l'utilisation d'un serveur mDNS est nécessaire afin de découvrir les serveurs REST et Openstack. Leurs utilisations sont détaillés dans les sous-parties 4 et 5.

4) *Processus de découverte, mDNS*: Lorsqu'un utilisateur se connecte à l'interface de l'application, il doit pouvoir mettre à disposition les ressources qu'il souhaite. Pour ce faire, la première étape consiste en la découverte du serveur REST



FIGURE 4. Image de l'interface utilisateur



FIGURE 5. Exemple de JSon

ainsi qu'en la découverte du serveur Openstack. Le protocole mDNS-SD signifie multicast Domain Name Service - Service Discovery. Il s'agit d'un mécanisme permettant d'adresser des messages dans le réseau sans avoir besoin de savoir avec quel périphérique l'on communique (mDNS). Il permet également de découvrir des services (Service Discovery). Pour mettre en place le service de découverte, la bibliothèque Python, zeroconf a été utilisée. La solution zeroconf permet de découvrir de manière simple les services se situant sur le même réseau. L'implémentation du serveur mDNS permet alors de publier le service du serveur REST, ainsi que du serveur Openstack et ensuite d'en transmettre les informations sur le réseau. Un script permet ensuite à l'Agent Node la découverte de ces serveurs et de communiquer avec.

5) *Serveur REST*: Une fois la découverte effectuée, afin que l'utilisateur puisse mettre à disposition ses appareils, il doit pouvoir enregistrer son appareil dans la base de données regroupant toutes les ressources disponibles sur le réseau d'Orange. Pour ce faire, nous avons implémenté un serveur REST (Representational State Transfer). Le serveur REST permet de mettre à disposition des clients une interface permettant d'envoyer des requêtes via un format défini au préalable et ensuite d'enregistrer les données reçues dans une base de données. Nous avons utilisé le format Json, très utilisé et simple d'utilisation. (cf. figure 5)

6) *Affichage de la base de données*: Afin de pouvoir rendre compte du bon enregistrement des appareils dans la base de données, nous avons créé une interface affichant la base de données. (cf. image dashboard) Afin de pouvoir rendre compte du bon enregistrement des appareils dans la base de données, nous avons créé une interface affichant la base de données. (cf. figure 6)



FIGURE 6. Représentation de notre Dashboard

#### IV. RÉSULTATS

##### A. Recherche sur OpenEdgeComputing

Les recherches effectuées sur OpenEdgeComputing n'ont pas été particulièrement convaincantes. Le projet OpenEdgeComputing est un projet universitaire assez bien documenté, cependant ce projet n'est plus maintenu depuis plus d'un an maintenant (3ans pour le code source). En plus, ce projet OpenEdgeComputing a été partiellement privatisé par les entreprises. Cela explique pourquoi nous avons une difficulté de progresser dans notre recherche. Le deuxième projet traitant de la même problématique est OpenFog cependant ce

projet n'en est encore qu'à ses balbutiements, et le premier forum public de ce consortium ne se tiendra qu'en mars.

### *B. Mise en place de OpenEdgeComputing*

La mise en place d'OpenEdgeComputing n'a pas été concluante en effet le manque de maintien du projet fait que de nombreux liens notamment au niveau des VMs ne sont plus disponibles. Le projet est à l'heure actuelle très difficilement exploitable et absolument pas prêt pour une quelconque mise en production.

### *C. Mise en place d'une nouvelle solution*

La nouvelle solution mise en oeuvre est concluante et fonctionne bien d'un point de vue logique à petite échelle : le stack de logiciel fonctionne dans un réseau local sans routeur. Il faudrait cependant déployer cette application à grande échelle afin de voir si celle-ci est scalable.

## V. ANALYSE

Une importante partie de notre travail a été consacré à la découverte de Edge Computing et de l'état de l'art des existants. Cependant peu de chose ont été faites dans ce domaine et il nous a fallu repartir de 0 et implémenter notre solution. Celle-ci n'est probablement pas exploitable en l'état à grande échelle mais permet de poser les bases sur lesquelles s'appuyer pour un développement futur plus important.

## VI. DISCUSSION

Le Edge Computing n'en est qu'à ses balbutiements, il n'y a pas encore de standard de disponible sur le marché que cela vienne d'un acteur privé ou du monde de l'open source. Cependant certains paradigmes semble revenir comme le montre l'article (scientifique traité par Merwan a retrouvé et a cité). La solution qui a été mise en oeuvre est basé sur ces paradigmes.

## VII. RÉFÉRENCES BIBLIOGRAPHIQUES