

UV « Projet » – Sujet Mercury

Distribution de l'architecture OpenStack

Tuteur	Adrien Lebre
Projet	<p>Mercury</p> <p>OpenStack est une suite logiciel permettant d'opérer des plateformes de Cloud Computing de type IaaS. Bien le passage à l'échelle (i.e. scalabilité) soit un critère décisif lors du choix d'un système pour opérer ce genre de plateforme, et bien que l'accent soit mis sur la scalabilité au cours du développement, de nombreuses briques fondamentales constituant le système OpenStack ne satisfont pas ce critère. L'équipe ASCOLA revisite depuis 2013 les principaux composants de cette suite logiciel afin de traiter ces limitations.</p> <p>Le travail proposé dans ce projet s'inscrit dans cette action. il consiste à :</p> <ul style="list-style-type: none">— Analyser le modèle de données exploité dans le composant OpenStack Glance— Remplacer le système MySQL par le système REDIS en exploitant le pilote ROME développé dans l'équipe.— Valider le prototype qui aura été mis en oeuvre au dessus de la plateforme Grid'5000. <p>A plus gros grain, ce projet s'insère dans l'initiative Discovery piloté par l'école.</p>
Site Web	http://beyondtheclouds.github.io
Intitulé	Remplacement du système MYSQL du composant OpenStack Glance en vue de sa distribution à large échelle.

Domaines

- Intégration logicielle
- Cloud Computing
- Modele de données (SQL, NoSQL, Clé/Valeur)

Compétences requises

- Fondamentaux Cloud Computing
- Maîtrise des modèles de données relationnels.
- Programmation objets/composants

Compétences à acquérir

- Expertise sur l'architecture OpenStack
- Langage python
- Développement de schémas de données NoSQL
- Contribution à un projet « open source »

Contexte

Afin de supporter la demande croissante d'informatique utilitaire (UC : Utility Computing) tout en prenant en compte les aspects énergétique et financier, la tendance actuelle consiste à construire des centres de données (ou centrales numériques) de plus en plus grands dans un nombre limité de lieux stratégiques. Cette approche permet sans aucun doute de satisfaire la demande tout en conservant une approche centralisée de la gestion de ces ressources mais elle reste loin de pouvoir fournir des infrastructures de calcul utilitaire efficaces et durables. L'initiative de recherche DisCoVery propose une approche radicalement différente permettant de prendre en compte la localité des demandes dès le départ. Pour ce faire, elle propose de tirer parti de tous les équipements disponibles sur l'Internet afin de fournir des infrastructures de calcul utilitaire qui permettront de part leur distribution de prendre en compte plus efficacement la dispersion géographique des utilisateurs et leur demande toujours croissante [1]. Un des aspects critique pour l'émergence de telles plates-formes de calcul utilitaire "local" (LUC : Locality aware Utility Computing) est la disponibilité de mécanismes de gestion appropriés exploitant les dernières contributions en terme d'algorithmiques distribués. Des premiers travaux menés à l'école des Mines ont permis de valider plusieurs mécanismes via la mise en œuvre de prototypes expérimentaux. Afin de passer du mode recherche à un mode plus industriel, nous avons commencé à regarder comment intégrer nos propositions au système OpenStack.

Problème : Les faiblesses dans la conception d'OpenStack

OpenStack est une suite logiciel permettant d'opérer des plateformes de Cloud Computing de type IaaS. Elle est aujourd'hui la solution OpenSource retenue et supportée par un grand nombre d'acteurs aussi bien académiques et qu'industriels (850 institutions sont impliquées dans son développement, avec des acteurs de premier rangs tels que HP, RedHat ou encore plus récemment Google). Bien qu'une des règles fondamentales pour le développement du système a pour objectif d'adresser la problématique du passage à l'échelle de chaque composant (le terme scalabilité sera utilisé par la suite), le système comprend plusieurs briques logicielles qui ne répondent pas à cette règle. A titre d'exemple, Jonathan Pastor, candidat à une thèse de doctorat à l'école, a identifié la base de données MySQL au sein du composant Nova et a proposé un pilote permettant de basculer sur un modèle NoSQL passant à l'échelle de manière native.

Objectif

Le travail demandé s'insère dans la continuité du projet proposé en 2014/2015 qui a permis d'identifier la faiblesse propre au système MySQL dans plusieurs sous-composants, en particulier le mécanisme Glance. Il comporte deux volets :

- Le basculement vers un modèle NoSQL pour le mécanisme Glance en s'appuyant sur le pilote ROME conçu par M. Pastor ;
- La validation du prototype implanté au dessus de la plateforme Grid'5000).

Livrables

- L0 : rapport sur l'architecture d'OpenStack (ces principaux composants et leur rôle) ;
- L1 : Le modèle SQL dans Glance et les extensions nécessaires au pilote ROME afin de permettre d'utiliser le système REDIS en place du système MySQL ;
- L2 : Version alpha du prototype et test de comportement sur la plateforme Grid'5000
- L3 : Version beta et évaluation des performances (selon l'avancement du projet les étudiants évalueront leur proposition via l'outil)Rally.
- L4 : synthèse des travaux réalisés.

Méthodologie du projet

Ce projet sera réalisé suivant une méthode agile. Il devra impliquer une contribution au projet « open source » Discovery.

Bibliographie :

[1] Adrien Lebre, Jonathan Pastor, Frédéric Desprez, white paper, July 2015.