

## UV « Projet » – Sujet DisOStack Distribution de l'architecture OpenStack

2 septembre 2015

<b>Tuteur</b>	Adrien Lebre
<b>Projet</b>	<p>DisOStack</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 les IaaS contemporains ne satisfont pas ce critère, impactant négativement la scalabilité globale.</p> <p>Le travail proposé comporte deux objectifs :</p> <ul style="list-style-type: none"><li>— L'identification de l'ensemble des briques élémentaires ne validant la propriété de scalabilité</li><li>— la proposition d'une solution et sa mise en œuvre au travers d'un prototype expérimental (selon l'avancée des travaux réalisées, les étudiants pourront mener des campagnes d'expériences sur la plateforme Grid'5000).</li></ul> <p>A plus gros grain, ce projet s'insère dans l'initiative Discovery piloté par l'école.</p>

<b>Site Web</b>	<a href="http://beyondtheclouds.github.io">http://beyondtheclouds.github.io</a>
<b>Intitulé</b>	Analyse du système OpenStack 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
- Prise en main de l'EDI IntelliJ
- 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 ou encore RedHat). 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, actuellement en thèse de doctorat à l'école, a identifié la base de données MySQL au sein du composant Nova et a débuté des travaux visant à basculer sur un modèle NoSQL passant à l'échelle de manière native.

**Objectif**

L'objectif du travail proposé dans ce projet d'option s'insère dans la continuité de ces travaux. Il comporte deux volets :

- L'identification de l'ensemble des briques élémentaires ne validant le critère de scalabilité.
- la proposition d'une solution et sa mise en œuvre au travers d'un prototype expérimental (selon l'avancée des travaux réalisées, les étudiants pourront mener des campagnes d'expériences sur la plateforme Grid'5000).

**Livrables**

- L0 : rapport sur l'architecture d'OpenStack (ces principaux composants et leur rôle).
- L1 : identification des briques logicielles ne respectant par la règle du passage à l'échelle.
- L2 : Proposition d'une solution pour un des composants choisis avec le tuteur.
- L3 : implémentation de la solution et évaluation sur la plateforme Grid'5000.
- 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, Marin Bertier, Frédéric Desprez, Jona-

than Rouzaud-Cornabas, Cédric Tedeschi, Paolo Anedda, Gianluigi Zanetti, Ramon Nou, Toni Cortes, Etienne Rivière, and Thomas Ropars. Beyond The Cloud, How Should Next Generation Utility Computing Infrastructures Be Designed ? Research Report RR-8348, Inria, July 2013.