

Projet d'option GSI Vivaldi

Rapport final

Nicolas Joseph, Raphaël Gaschignard
Guillaume Blondeau, Cyprien Quilici, Jacob Tardieu

19 février 2014

1 Le Contexte

Le projet s'est déroulé est en réalité un sous-projet au projet **DisCoV_Ery**. Ce projet qui est né suite à l'impulsion du département de recherche en informatique de l'École des Mines de Nantes et consiste à explorer ce que pourrait être le modèle d'après cloud qui est en pleine expansion. En effet, il apparaît que ce modèle est construit autour de centre de données centralisés qui eux même peuvent être excentrés géographiquement des zones d'utilisation. De cette remarque vient le modèle **DisCoV_Ery** : rapporter les serveurs auprès des utilisateurs. Plus précisément les rapporter au niveau des différents noeuds de réseau, de manière à construire un cloud complètement décentralisé.

2 La Problematique

Au sein d'un réseau de type cloud, il faut pouvoir migrer des VM¹ sur des machines différentes grâce à une notion de localité. C'est là où notre projet intervient. Grâce à un algorithme prédéfinie, Vivaldi, notre brique logicielle doit être capable de fournir les noeuds les plus proches de la machine courante.

Cette brique logicielle devait être complètement indépendante des autres systèmes **DisCoV_Ery** de manière à pouvoir être remplacée facilement le cas échéant.

1. Machines Virtuelles

3 Méthodologie

3.1 Gestion de Projet

Ce projet a été réalisé en autonomie avec des réunions régulières suivant une méthode de développement agile. Nous reviendront plus en détail sur le détail de la méthodologie de développement dans la partie 3.2. Nous avons choisi un chef de projet qui avait un rôle de management administratif du projet. Il a géré une partie des relations de l'équipe avec les tuteurs et les éléments de gestion de projet comme le suivi d'avancement des différentes tâches, la répartition des tâches ...

Nous avons utilisés plusieurs outils en ligne pour la gestion de projet :

- Trello
- Propulse
- GitHub

Nous avons commencé par utiliser Trello pour mettre en place le cahier des charges et les premières briques de base de Vivaldi et mettre en place différentes dates de rendu. Nous avons ensuite basculé sur Propulse, qui nous semblait un outil un peu plus conventionnel et utile que Trello pour la fin du projet.

Nous avons aussi régulièrement fait des réunions avec les différents acteurs du projet, à la fois les clients et les tuteurs école.

3.2 Developpement

Le projet a commencé par une phase de courtes spécifications. Cette phase a abouti à l'élaboration d'une architecture logicielle permettant de répondre aux contraintes imposées, à un choix de technologie (Scala + Akka) et à la répartition de différentes tâches de développement parallélisables.

Le développement a donc pu commencer pour un premier sprint et la méthodologie était la suivante :

- Chaque développeur code les tests pour toutes les fonctions et fonctionnalités qu'il développe.
- Chaque développeur travaille sur une branche spécifique par fonctionnalité.
- Une fois le développement terminé, chaque développeur crée une pull request GitHub de manière à ce que le code soit relu et approuvé par quelqu'un d'autre
- Une fois le code approuvé, celui-ci est ajouté à la branche principale du projet

Une fois toutes les fonctionnalités développées et testées de manière indépendantes, nous avons développé un test local global de manière à valider

le bon fonctionnement et la stabilité de notre brique logicielle.

Il s'en ai suivit une série de sprints de test - fix de manière à corriger les bugs restant sur la plateforme.

En parallèle nous avons aussi décidé de développer un système de monitoring nous permettant d'aggreger des données sur le système distribué que nous étions en train de développer. Cela nous a permis de verifier que le comportement du réseau vivaldi était celui attendu.

4 Les Résultats

5 Le Bilan du projet

Références

- [1] Frank Dabek, Russ Cox, M. Frans Kaashoek, Robert Morris, *Vivaldi : a decentralized network coordinate system*. SIGCOMM 2004.