

Résumé : SUPERVISER UN CLOUD COMPUTING

Le but de ce travail est, dans un premier temps, d'étudier le système de virtualisation Linux-KVM ainsi que les outils permettant de récupérer des informations sur les charges CPU, RAM et disque. Puis, dans un second temps, de développer un logiciel de monitoring centralisé des machines virtuelles fonctionnant sur des hôtes équipés de Linux-KVM. Ce travail est associé au projet Visag de la HES-SO (<http://www.tdeig.ch/vmware/Visag.pdf>).



Ce travail s'est déroulé sur huit semaines et a été réalisé en six étapes :

1. Prise en main de Fedora, des outils de virtualisation et de mesure des performances.
2. Étude des résultats de ces précédents outils, mise en évidence des valeurs utiles.
3. Conception des scénarios de test.
4. Réalisation des outils permettant de mettre en pratique ces scénarios.
5. Spécification et développement du superviseur.
6. Mise en pratique des scénarios de test.

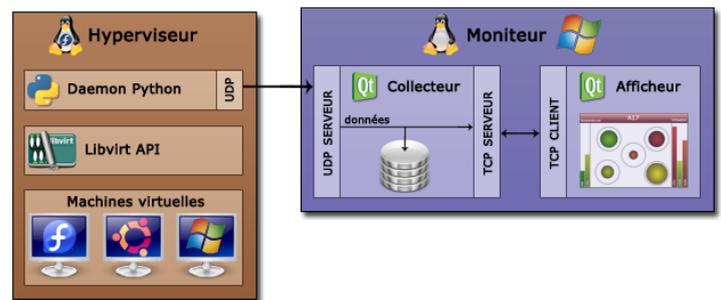
Les scénarios de test ont été conçus dans les premières semaines pour permettre de valider le bon fonctionnement et l'utilité du système de monitoring (nommé le superviseur).

Pour cela, il faut pouvoir générer une charge variable sur divers composants du système telle que le processeur, la mémoire et le disque dur. Des outils dédiés à ces tâches ont été développés grâce aux langages Python et C.

La réflexion sur l'architecture du superviseur n'a pas été triviale. En effet, il a fallu choisir entre la multitude de possibilités que l'informatique nous offre, tout en tenant compte des contraintes.

J'ai choisi de séparer le superviseur en trois sous-programmes:

1. Le daemon programmé en Python qui va récupérer les informations sur l'hyperviseur et ses machines virtuelles pour les transmettre en UDP au collecteur.
2. Le collecteur programmé en C++/Qt qui va recevoir les informations en provenance des daemons de la grille et les retransmettre aux afficheurs connectés en TCP.
3. L'afficheur également programmé en C++/Qt qui se charge de se connecter au collecteur et d'afficher les données à l'utilisateur.



Les communications entre les différents sous-programmes sont effectuées via des sockets avec comme contenu une chaîne au format JSON ([www.json.org](http://www.json.org)).

Diplômant :

**M. SCHAUB LIONEL**

Classe : IN3

Filière d'études : Informatique

Unité d'Enseignement et de Recherche UER4

Timbre de la direction

