

Mathieu Bivert, CSSR, bivert@essi.fr

PFE : Rapport de management (D_4) (draft)

Mars 2013

Placement constraints for a better QoS in clouds



Coût du livrable 38 heures

Budget total du projet 304 heures

Entreprise Université de Nice-Sophia Antipolis

Lieu Sophia-Antipolis, France

Responsable Fabien Hermenier, équipe OASIS, fabien.hermenier@unice.fr

Contents

1	Description du projet	3
2	Synthèse des résultats obtenus	3
2.1	Support du typage	3
2.1.1	Cas particulier	3
2.1.2	Cas général	3
2.2	Nouvelles contraintes	3
3	Implication des personnes	4
3.1	Fabien Hermenier	4
3.2	Mathieu Bivert	4
4	Synthèses des livraisons	4
5	Suivi budgétaire	4
5.1	Consommation du budget	4
5.2	Synthèse	4
6	Suivi des lots	4
7	Synthèse et retour d'expérience	4

1 Description du projet

On renvoie le lecteur à la section *Description du projet* du DOW pour description plus précise de l'environnement, du vocabulaire et des besoins de l'utilisateur.

BtrPlace est un logiciel développé à l'INRIA par Fabien Hermenier qui place efficacement un ensemble de machines virtuelles sur un ensemble de serveurs physiques à l'aide de contraintes. Celles-ci sont principalement de deux natures:

1. les contraintes spécifiées par l'utilisateur, qui répondent généralement à des cas d'utilisations particuliers;
2. les contraintes imposées par les ressources disponibles, portant par exemple sur la mémoire, la puissance de calcul disponibles sur un nœud.

Dans le cadre de ce projet, on se propose de travailler en deux temps:

1. ajouter au modèle théorique sous-jacent la possibilité de *typer* les nœuds et les VMs pour rendre compte de la présence de multiples hyperviseurs (Xen, VMWare, etc.);
2. concevoir et implémenter des *contraintes* mettant en œuvre ce typage.

Pour ce faire, le travail a suivi une progression incrémentale:

1. on a commencé par chercher un cas simple du typage, et on implémente une contrainte le décrivant;
2. dans un premier temps, on exprime le typage dans un cas plus général, d'abord mathématiquement, puis, toujours d'une façon abstraite, sous forme de contrainte;
3. enfin, on implémente des contraintes mettant en œuvre ceci.

2 Synthèse des résultats obtenus

2.1 Support du typage

2.1.1 Cas particulier

On ajoute une contrainte permettant d'associer à des nœuds un nouvel type, c'est-à-dire en pratique, d'effectuer une action de retypage sur un ensemble de nœuds. Cette dernière se décompose en:

1. migrer les éventuelles VMs tournant sur ce nœuds;
2. éteindre le nœud;
3. le redémarrer sous un nouvel hyperviseur.

2.1.2 Cas général

On associe à chaque nœud un vecteur contenant un élément pour chaque hyperviseur possible. Sémantiquement, les éléments de ce vecteur représentent le nombre de machines virtuelles d'un type donné tournant sur ce nœud. Une valeur nulle indique que l'hyperviseur n'est pas utilisé; si le vecteur ne contient que des zéros, alors le nœud est considéré comme éteint.

Il s'en suit qu'une seule composante du vecteur peut-être non nulle : un nœud ne peut pas faire tourner deux hyperviseurs simultanément.

2.2 Nouvelles contraintes

En plus de la contrainte du cas particulier, deux nouvelles contraintes peuvent être implémentée. La première permet de répondre aux limitations induites par les licences des logiciels de virtualisation, en fixant une limite au nombre de machines virtuelles tournant sur un nœud. Une variante de celle-ci est de restreindre le nombre de nœuds tournant de façon simultanée sous un même hyperviseur. Enfin, afin de s'assurer de la présence de certaines plateformes pour faciliter les opérations de déploiement future, on peut aussi s'assurer qu'il existe au moins un nombre donné de nœuds faisant tourner un hyperviseur.

3 Implication des personnes

XXX

3.1 Fabien Hermenier

3.2 Mathieu Bivert

4 Synthèses des livraisons

Les livraisons sont effectuées pour mercredi 7 mars 2013, ce qui n'est pas conforme au planning prévu. L'un des problèmes majeurs rencontrés repose sur l'approche incrémentale utilisée. Bien que celle-ci permette de progresser facilement, elle a tendance à lier fortement les différentes composantes du projet. L'interdépendance entre les éléments des lots complique la possibilité de rendre au temps fixé dans le DOW les différents livrables.

Pour la même raison, la structure des livrables D_2 et D_3 n'est pas optimale : en effet, la majeure partie de l'implémentation du modèle décrit dans D_2 est à placer dans les contraintes présentes dans D_3 . Ces problèmes n'avaient pas été envisagés dans le DOW.

5 Suivi budgétaire

5.1 Consommation du budget

Le temps de rédaction des rapports et pour la gestion du management du projet ont été surévalués : il aurait été plus judicieux de passer plus de temps sur l'implémentation du code.

5.2 Synthèse

Globalement, le budget fixé a été respecté. Le projet était peut-être un peu trop ambitieux au vu du temps difficile à prévoir pour les cours et événements para-scolaires.

6 Suivi des lots

XXX

7 Synthèse et retour d'expérience

En ne prenant en compte que le projet, la gestion du temps était loin d'être optimale. Bien que la partie théorique soit assez facile à comprendre, sa mise en œuvre dans BtrPlace est bien plus difficile. La complexité de l'implémentation est à mon avis l'un des facteurs majeurs justifiant le temps passé à implémenter le modèle. Une solution à ce problème de complexité serait de ré-implémenter BtrPlace dans un langage plus adapté pour la programmation par contrainte (Prolog, Lisp, etc.), mais cela nécessite d'étudier le coût d'une telle ré-écriture. En effet, la nouvelle solution peut-être plus facile à comprendre et à étendre, mais elle peut aussi:

1. être moins performante et plus difficilement optimisable;
2. nécessiter un temps d'implémentation pas forcément rentable par rapport au temps pouvant être mis à étendre le code actuel par une personne en ayant une connaissance approfondie.

À l'heure actuelle, les ajouts en terme de code sont incomplets, mais:

1. l'implémentation du typage telle que décrite plus haut permet de résoudre un autre projet proposé par Fabien Hermenier, à savoir la gestion de la limitation sur le nombre de VMs pour un hyperviseur donné pour des questions de licence;

2. le modèle théorique peut-être facilement étendu pour gérer d'autres limitations au niveau des ressources, en fonction du type de l'hyperviseur. En effet, les licences des systèmes de virtualisation imposent généralement des limites sur la quantité de RAM utilisable, sur la puissance CPU ou encore limite le nombre d'interface réseau, la quantité de RAM, etc.