

Mathieu Bivert, CSSR, bivert@essi.fr

---

## PFE: Cahier des charges (DOW)

---

# Placement constraints for a better QoS in clouds



**Entreprise** Université de Nice-Sophia Antipolis (INRIA)

**Lieu** Sophia-Antipolis, France

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

## Résumé

Blablabla français

## Abstract

Blablabla english

# Contents

<b>1</b>	<b>Description du projet</b>	<b>3</b>
1.1	Contexte de travail . . . . .	3
1.2	Motivations . . . . .	4
1.3	Défis . . . . .	4
1.4	Objectifs . . . . .	4
1.5	Scénarios . . . . .	5
1.6	Critère de succès . . . . .	5
<b>2</b>	<b>État de l'art</b>	<b>5</b>
2.1	Description générale . . . . .	5
2.2	foo . . . . .	5
2.3	bar . . . . .	5
<b>3</b>	<b>Méthodologie et planification</b>	<b>5</b>
3.1	Stratégie générale . . . . .	5
3.2	Découpage en lots . . . . .	5
3.3	Plannification . . . . .	5
3.4	Livrables associés au projet . . . . .	6
3.5	Jalons . . . . .	6
<b>4</b>	<b>Description de la mise en œuvre du projet</b>	<b>6</b>
4.1	Interdépendance des lots et tâches . . . . .	6
4.2	Description des lots . . . . .	6
4.3	Résumé de l'effort . . . . .	6
4.4	Gestion du risque . . . . .	6
<b>5</b>	<b>Participants</b>	<b>6</b>
5.1	Mathieu Bivert - CSSR . . . . .	6
5.2	Fabien Hermenier - OASIS/INRIA . . . . .	6

# 1 Description du projet

## 1.1 Contexte de travail

Le monde industriel étant de plus en plus informatisé, la qualité des réseaux s'améliorant, les sociétés informatiques tendent à ne vendre plus de logiciels et de matériels mais louent des structures informatiques.

Les entreprises de services informatiques sont spécialisées dans la maintenance de ces structures, donc par exemple plus performante qu'une société de l'industrie automobile. Celle-ci peut donc choisir de déporter donc la charge et paye pour une assurance sur la qualité du service (QoS).

Plusieurs types d'entreprises existent selon le service qu'elles proposent:

**SaaS** Software As A Service;

**PaaS** Platform As A Service;

**IaaS** Infrastructure As A Service;

**DaaS** Data As A Service;

...

En particulier, un cloud **IAAS** fournit à l'utilisateur l'accès à un ensemble de systèmes d'exploitations. Ces derniers sont très souvent virtualisés, ce qui présente l'avantage de pouvoir faire tourner plusieurs OS sur un même serveur physique. Par exemple, la figure 1 montre l'hyperviseur *Xen* en train de faire tourner, en plus d'un dom0 sous NetBSD, un FreeBSD, deux NetBSDs et une Debian:

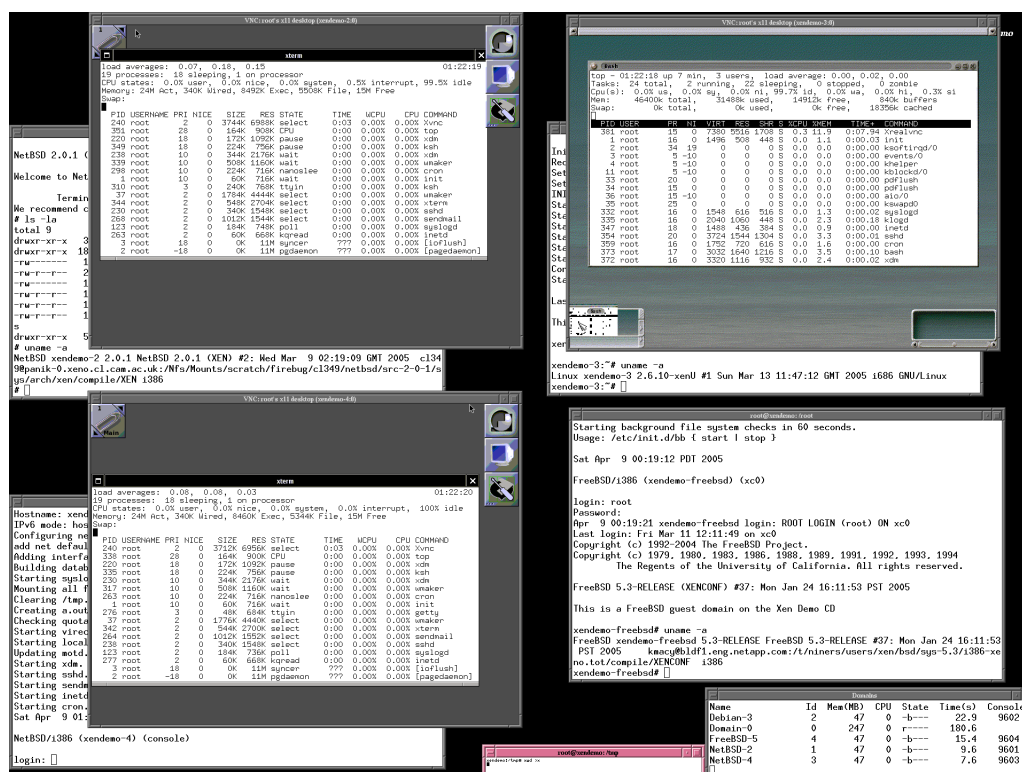


Figure 1: Virtualisation avec Xen

La virtualisation a plusieurs avantages:

- sur une application dites *n*-tiers, il est possible de placer chaque tiers sur une VM, et éventuellement d'en faire des duplications, ce qui améliore la robustesse de l'application;

- l'administration et la gestion des machines est simplifiée : il y a moins de hardware, donc moins de maintenance physique;
- chaque application peut être répartie sur une VM différente. Ainsi, si une application est compromise, elle a moins de chances de pouvoir compromettre d'autres applications que si elles étaient toutes lancées sur une même machine physique.
- utilisation plus performante du matériel : on est plus dans le cas où dans une entreprise, des dizaines de postes de Bureau "assez"puissants sont tous utilisés en même temps à quelques dixième de leur capacité totale;
- ...

Une problématique pour les gestionnaires d'IAAS est donc de pouvoir placer correctement un ensemble donné de VMs  $\mathcal{V}$  sur un ensemble de serveurs physiques  $\mathcal{N}$ .

## 1.2 Motivations

La question de la répartition des machines virtuelles sur les machines physiques se pose alors pour des raisons diverses et variées:

**maintenance** un serveur physique peut tomber en panne, ou nécessiter une réparation, auquel cas les programmes tournant dessus doivent être migré ailleurs, afin de garantir au client une certaine qualité de service (QoS);

**sécurité** il peut s'avérer risquer pour un programme d'un client traitant des données sensibles (eg. données bancaires) de se retrouver au même endroit qu'un programme d'un autre client;

**évolution des besoins** où au cours d'un certain intervalle de temps, les besoins en puissance de calcul d'une entreprise peuvent augmenter (suite à une plus grande popularité par exemple), ou encore, augmentation brusque et irrégulière de la charge à des heures de pointes;

**économie d'énergie** où il peut être avantageux de réduire le nombre de serveurs physiques allumés, pour maximiser le rendement des autres machines physiques du cloud;

**QoS** où, à l'inverse de l'économie d'énergie, il est bon de garder des ressources supplémentaires disponibles immédiatement, de façon à ne pas perdre de temps (et donc en QoS) à redémarrer un autre serveur;

**licence** les entreprises fournissant les systèmes de virtualisation proposent des licences selon différents critères (eg. nombre de machines virtuelles lancées, utilisation de ressources (CPU, RAM, etc.));

**plateforme** plusieurs plateformes de virtualisations sont disponibles (eg. Xen, VMWare, Citrix); une autre contrainte sur la répartition des machines virtuelles se pose alors, un serveur physique ne faisant tourner qu'un seul type de plateforme;

...

## 1.3 Défis

Afin de pouvoir répondre aux besoins exprimés par l'un des domaines cité dans le paragraphe précédent, il est nécessaire de commencer par formaliser le problème. En d'autres termes, donner une définition mathématiques des contraintes impliquées par la problématique choisie, et s'assurer qu'elles sont envisageables en pratique. Finalement, cette représentation abstraite doit être implémentée sous forme de plugin Java pour entropy [FH09] un manager de clusters reposant sur l'algorithme btrplace<sup>1</sup>.

## 1.4 Objectifs

Actuellement, les trois derniers points cités ne sont pas forcément formalisé/implémenté sous une forme satisfaisante. Le projet consiste donc à choisir l'un de ces domaines et à l'ammener vers une forme satisfaisante.

Le dernier point est celui sur lequel se porte ce projet.

---

<sup>1</sup><http://btrp.inria.fr/sandbox/about.html>

## 1.5 Scénarios

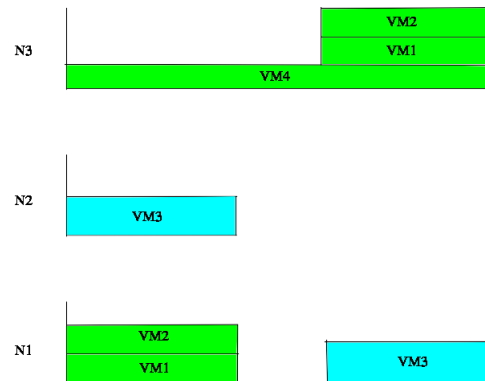


Figure 2: Exemple de changement de système de virtualisation. En vert des VMs Xen, en cyan une VM VMWare

Sur le diagramme ci dessus, le serveur physiques  $N_2$  doit être mis hors-ligne pour des questions de maintenance, via la contrainte  $offline(N_i)$ .<sup>2</sup>.

Comme aucun serveur VMWare n'est disponible, il est nécessaire de supprimer un serveur Xen, capable d'accueillir  $VM_3$ , par exemple  $N_1$ .

## 1.6 Critère de succès

bis repetita: trouver un bon formalisme; définir et implémenter les contraintes.

## 2 État de l'art

### 2.1 Description générale

### 2.2 foo

### 2.3 bar

## 3 Méthodologie et planification

### 3.1 Stratégie générale

bis repetita: trouver un bon formalisme; définir et implémenter

### 3.2 Découpage en lots

bis repetita: trouver un bon formalisme; définir et implémenter

### 3.3 Plannification

gantt

Id	Titre du livrable	Lot(s)	Nature	Date
$D_0$	Cahier des charges	1	Document	$S_4$
$D_1$	Gestion du typage et du déploiement	1	Document	?
$D_2$	Ensemble de contraintes	1	Document et Logiciel	?
$D_3$	Rapport de management	1	Document	$S_{20}$
$D_4$	Diaporama de présentation finale	1	Document	$S_{20}$

Id	Jalon de fin de phase	Lot(s)	Date	Vérification
$J_0$	planification	1	$S_4$	$D_0$
$J_1$	formalisation	1	$S_n$	$D_1$ partiel
$J_2$	implémentation	1	$S_{n+k}$	$D_2, D_1$ partiel
$J_3$	projet	1	$S_{20}$	$D_1, D_2, D_3$ et $D_4$

### 3.4 Livrables associés au projet

### 3.5 Jalons

## 4 Description de la mise en œuvre du projet

### 4.1 Interdépendance des lots et tâches

bis repetita: trouver un bon formalisme; définir et implémenter

### 4.2 Description des lots

bis repetita: trouver un bon formalisme; définir et implémenter

### 4.3 Résumé de l'effort

### 4.4 Gestion du risque

## 5 Participants

### 5.1 Mathieu Bivert - CSSR

Étudiant à Polytech'Nice Sophia, spécialisé en Cryptographie, Systèmes Sécurité et Réseaux.

### 5.2 Fabien Hermenier - OASIS/INRIA

**Fabien Hermenier** a reçu un doctorat en 2009 à l'université de Nantes. Depuis 2011, il enseigne en tant que professeur adjoint à l'université de Nice Sophia-Antipolis. Son travail de recherche s'articule autour des plateformes d'hébergement, de la virtualisation, du calcul autonome et de la gestion des ressources. Depuis 2006, il travaille sur des algorithmes de placement de machines virtuelles pour faire face à l'augmentation des SLA dans les plateformes d'hébergements.

---

<sup>2</sup><http://www-sop.inria.fr/members/Fabien.Hermenier/btrpcc/offline.html>

## Références

- [FH09] Jean-Marc Menaud Gilles Muller Julia lawall Fabien Hermenier, Xavier Lorca. Entropy : a consolidation manager for clusters. 2009.