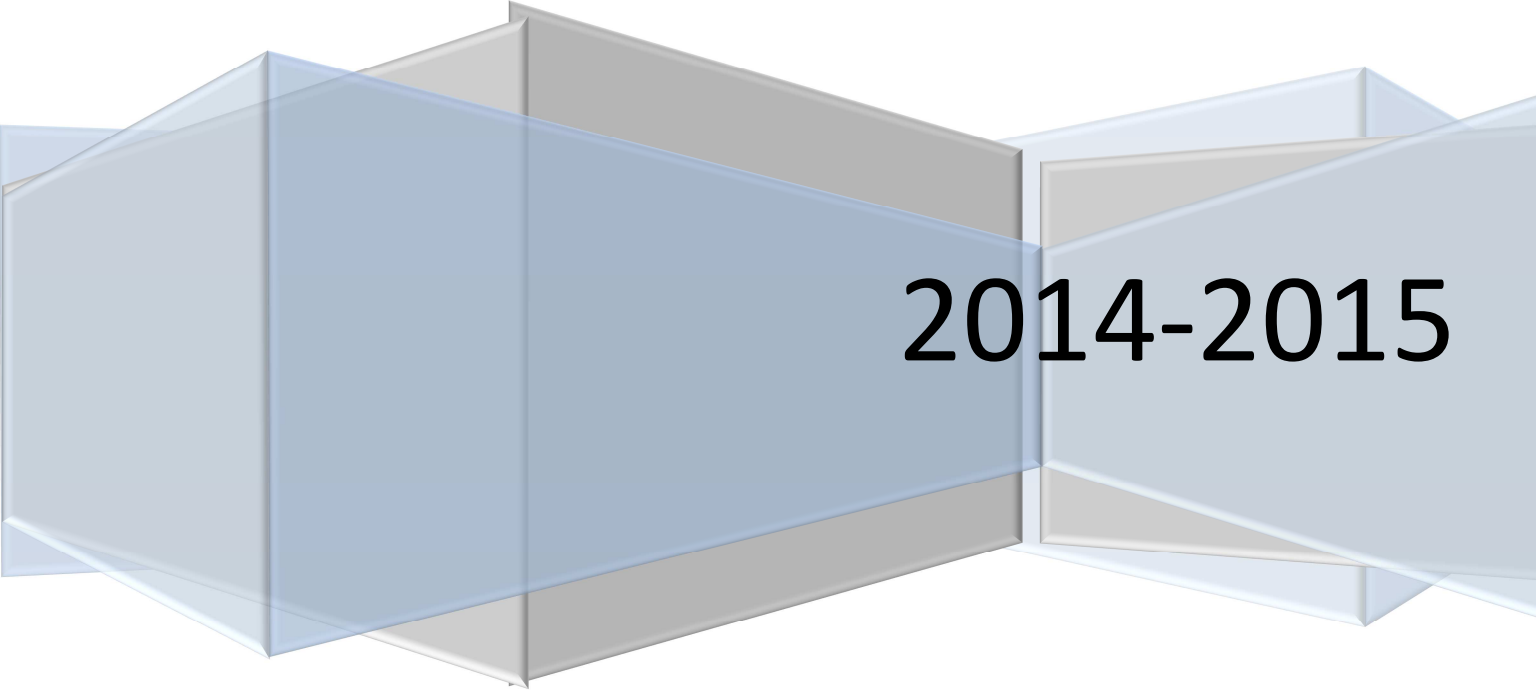


Master2 RISM 2014-2015

RAPPORT DE PROJET



2014-2015

Réalisé par : Djatougbe Sika APETI

Encadré par: Mr Fabien HERMENIER

6 classes principales ont été ajoutées au projet :

I. Classes principales ajoutées au projet – fonctionnement

MyAntiAffinityObserver

Cette classe contient la méthode « PreprocessEvent » qui est la méthode où nous définissons les règles et écrivons dans les logs. Techniquement cette méthode récupère la liste des hosts, vérifie l'id la première Vm qui se trouve dessus, la prend comme référence et enregistre une alerte lorsqu'une autre vm d'affinité différente ne devant pas être sur le même hôte s'y trouve.

MynaivePolicy

Cette classe contient une méthode d'allocation « allocateHostForVm » qui ajoute la vm au premier hôte capable de l'accueillir.

MyantiAffinyPolicy

Cette classe permet de gérer le placement des Vms sur les nœuds en se basant sur les affinités. Elle contient une méthode d'allocation « allocateHostForVm » qui ajoute les Vm selon leur affinité : les Vm ayant des id allant de 1 à 99 sont sur les hôtes différents et pareils pour ceux de 100 à 199 et ainsi de suite.

Elle contient aussi un algorithme qui calcule dans un premier temps le nombre total d'hôtes ensuite récupère le reste de la division des id par 100.

Exemple

```
0 % 100 => 0
1 % 100 => 1
.
.
99 % 100 => 99
100 % 100 => 0
101 % 100 => 1
```

On dispose ensuite chaque Vm sur un hôte ayant l'indice du reste de sa division par 100 ce qui fera que la Vm d'id 0 sera sur l'hôte d'id 0 ou 100 ou 200. La Vm d'id 1 ira sur l'hôte 1 ou 101 ou 201 et ainsi de suite. Nous respectons ainsi l'affinité.

MynoViolationsPolicy

Elle s'assure du respect des SLA en se basant sur le MIPS disponible sur les hôtes avant d'y mettre les VM.

Cette classe contient une méthode d'allocation « allocateHostForVm » qui reprend la même logique que celle du load balance (placer la Vm sur l'hôte ayant le plus grand MIPS dispo) avec un contrôle supplémentaire lors de l'allocation.

MyLoadBalanceObserver

Cette classe contient une méthode « PreocessEvent » qui calcule les différentes rations de RAM CPU et MIPS. L'affichage de ces informations est fait dans la méthode shutdownEntity

MyLoadBalancePolicy

Cette classe contient une méthode d'allocation « allocateHostForVm » qui Trie la liste des hôtes par Mips décroissant. Nous nous servons ici des facilités que nous offrent java en utilisant un comparateur (HostMipsComparator) que nous avons créé.

II. Différents ordonnanceurs développés

- Naive
- antiAffinity
- noViolations
- balance

La commande suivante permet de tester les différents ordonnanceur en remplaçant à chaque fois l'attribut « -Dsched » par le nom correspondant

```
C:\projet\sika>C:\maven\bin\mvn compile exec:java -Dsched=balance -Dday=20110303
```