

Travail à rendre 2 : TP Cloudsim

Le but de ce travail est de mieux comprendre et vous familiariser avec le fonctionnement des simulations avant de passer aux choses sérieuses.

Vous pouvez suivre ces quelques questions + Jouer avec le nombre d'entités, de type de stratégie (fin tout ce qui vous tombe dans la tête), exécuter à chaque fois, me préciser votre scénario exécuté et décrire tout ce que vous constatez avec vos explications.

Pour chaque exemple (1 jusqu'à 8), schématiser la simulation et remplir le tableau suivant : (nbr de centre de données, nbr de hôtes, nbr de hôtes/datacenter, allocation des hôtes, nbr de vms, nbr vms/datacenter, nbr vms/hôtes, scheduling des vms, nbr de cloudlets, scheduling des cloudlets, temps nécessaire pour l'exécution des cloudlets).

1. Exemple1 # Exemple2 :
 - a. Il est clair que la différence entre l'exemple1 et l'exemple2 s'agit du nombre de cloudlets exécutés, mais (dans l'exemple2) : comment cette exécution de cloudlets est elle exprimée au niveau du code (je parle des machines virtuelles, ainsi la planification des cloudlets entre ces machines virtuelles)
 - b. (Toujours dans l'exemple2) : Changer le type de planification des tâches en « space shared », quelles lignes avez-vous modifiées ? Que se passe-t-il aux tâches ? Pourquoi ?
2. Exemple2 # Exemple3 :
 - a. Comment est exprimée la différence entre les deux exemples en termes de code ?
 - b. (Dans l'exemple3) : Qu'elle est la différence entre les machines virtuelles créées ? En quoi ce impacte l'exécution des cloudlets ?
3. Exemple4 :
 - a. Rajouter un hôte avec les mêmes caractéristiques, (incrémentez d'abord l'hostID avant de rajouter l'hôte), que se passe-t-il ? Expliquez la différence par rapport au résultat précédent ? (Voir tableau les résultats dans la console)
 - b. Rajouter deux machines virtuelles et décrire le scheduling des vms entre les hôtes.
4. Exemple6 :
 - a. Réaliser un jeu de nombre de votre choix (rajouter des processeurs PEs, supprimer des processeurs PEs, rajouter des hôtes, supprimer des hôtes, rajouter des vms, supprimer des vms, changer le scheduling à chaque fois, rajouter des tâches, supprimer des tâches, changer le scheduling...) => Faites un tableau pour chaque scénario, décrivez ce qui se passe et expliquez.

Le 6^{ème} exemple va énormément vous aider dans la manipulation, n'hésitez pas à modifier davantage !
IT'S FUN !

Travail à rendre 3 : TP Cloudsim

Gestion des évènements.

L'objectif de ce travail est de comprendre le fonctionnement et la gestion des événements au sein des entités de Cloudsim.

Le but sera de connaître l'ensemble des events et méthodes pendant l'exécution d'une simulation classique.

Après une bonne maîtrise, vous serez apte de modifier le comportement des entités en changeant/rajoutant de nouvelles fonctionnalités. De plus, vous pourrez rajouter vos propres événements, les gérer et les modifier selon vos besoins.

Au-dessous, un output de console en utilisant :

- 2 Datacenters
- Host1 : 4 PEs, Host2 : 2 PEs
- 3 VMs, vmTimeShared
- 5 Cloudlets, cloudletTimeShared

Vous pouvez utiliser votre propre configuration (plus de DCs, plus de VMs, plus de Cloudlets) en cas de besoin.

```
datacenter_1 is starting...
Broker is starting...
Entities started.
0.0: Broker: Cloud Resource List received with 2 resource(s)
0.0: Broker: Trying to Create VM #0 in Datacenter_0
0.0: Broker: Trying to Create VM #1 in Datacenter_0
0.0: Broker: Trying to Create VM #2 in Datacenter_0
0.1: Broker: VM #0 has been created in Datacenter #2, Host #0
0.1: Broker: VM #1 has been created in Datacenter #2, Host #0
0.1: Broker: VM #2 has been created in Datacenter #2, Host #0
0.1: Broker: Sending cloudlet 0 to VM #0
0.1: Broker: Sending cloudlet 1 to VM #1
0.1: Broker: Sending cloudlet 2 to VM #2
0.1: Broker: Sending cloudlet 3 to VM #0
0.1: Broker: Sending cloudlet 4 to VM #1
1.1: Broker: Cloudlet 2 received
2.1: Broker: Cloudlet 0 received
2.1: Broker: Cloudlet 3 received
2.1: Broker: Cloudlet 1 received
2.1: Broker: Cloudlet 4 received
2.1: Broker: All Cloudlets executed. Finishing...
2.1: Broker: Destroying VM #0
2.1: Broker: Destroying VM #1
2.1: Broker: Destroying VM #2
```

Pour faire simple, vous allez vous limiter à décortiquer, uniquement, les étapes **ENTRE** la réception des ressources (marquée en jaune) et l'exécution de toutes les tâches (marquée en vert).

Pour aider, je vous ai mis à disposition les questions suivantes.

1. Les événements sont prédéfinis dans quelle(s) classe(s) ?

2. « 0.0: Broker: Cloud Resource List received with 2 resource(s) »
 - a. Quelle classe/entité permet d'afficher ce résultat ?
 - b. Quelle méthode permet d'afficher ce résultat ?
 - c. La méthode est déclenchée par quel événement (event1) ?
 - d. Quel événement (event2) est déclenché lors de l'exécution de la méthode ?
 - i. Cet événement sera envoyé vers quelle entité ?
 - ii. L'événement est répété combien de fois ?
 - iii. Une fois reçu, l'événement permet de faire quoi ?
 - iv. Décrire le scénario qui s'enchaîne : quel événement se déclenche, entité destination, méthode exécutée ?
 - e. Quel est le but des étapes de a. jusqu'à d. ?
 - i. Définir les caractéristiques des datacenters et demander la création de machines virtuelles
3. La méthode *createVmsInDatacenter(...)* :
 - a. Se trouve dans quelle entité ?
 - b. Permet de faire quoi ?
 - c. Quel événement (event3) est déclenché lors de l'exécution de cette méthode (method3) ? Vers quelle entité ? Quelle méthode est exécutée ?
 - d. Quel est le but de cette étape ?
4. Method3 à son tour déclenche un événement.
 - a. Préciser la source, la destination, la méthode qui sera exécutée.
 - b. Quel est le but de cette étape ?
5. Continuer de cette manière jusqu'à ce que toutes les cloudlets sont exécutés.
6. Réaliser un schéma contenant la totalité des événements, méthodes, sources, destinations...

Le mode débogage est une méthode qui vous aidera sur Eclipse.

Travail à faire en binôme et à envoyer à : lailabouhouch94@gmail.com

Dernier délai : 25/05 à midi,