VU Large-scale distributed computing SS 2013 (184.271)

Assignment

1. Introduction

The goal of this assignment is to implement an energy efficient cloud elasticity manager in a team of three students. You have to simulate a cloud with a fixed number of physical machines (PM), variable number of virtual machines (VM) and also variable number of applications running on VMs. On your PMs you can have multiple virtual machines (VM) where your applications run on. Implement three different scheduling algorithms (given below) and measure some characteristics that are described below in more detail. Additionally, you have the possibility for federation (in- and outsourcing).

2. Cloud components

The cloud consists of VMs, PMs and applications. Characteristics for VMs and PMs are size, RAM and CPU. VMs have an initial consumption (marked with a *) which needs the VM to run on the PM. The VM provides size, RAM and CPU for applications to run. If you have a PM with a size of 50000 MB and the VM has an initial consumption of 100 MB, the VM can only provide 49900 MB for applications. The same holds true for RAM and CPU. A PM has also an initial consumption for the operating system running on it (also marked with a *).

VM:

- Size [MB] 100* … size of PM
- RAM [MB] 50* … size of PM
- CPU [MHz] 150* … size of PM

PM:

- Size [MB] 50 000 (850*)
- RAM [MB] 4 700 (300*)
- CPU [MHz] 2 400 (500*)
- Power consumption: Initial consumption (200 W) + linear dependency of load (0.3W / MHz)

* Initial consumption

Applications:

- Web Based Applications (WBA) uses small size, CPU and RAM of the VM. For example a web site or a web service.

    - Size [MB] 20 … 100
    - CPU [MHz] 50 … 100
    - RAM [MB] 30 … 50
    - Duration and request arrival rate are normal distributed
- High Performance Applications (HPA) have a big consumption of size, CPU and RAM. Think of big rendering tasks or processing of big data.
    - Size [MB] 150 … 500
    - CPU [MHz] 100 … 400
    - RAM [MB] 100 … 700
    - Duration and request arrival rate are normal distributed

Our model for the deployment of VMs and applications looks like the following:

- 1 VM : 1PM (One VM is running on only one PM)
- 1 PM : n VMs (On one PM there a multiple VMs)
- 1 App : 1 VM (One applications is running on only one VM)
- 1 VM : m Apps (One VM can host multiple applications)

3. Scenarios

Scenario A: Only WBA

Scenario B: Only HPA

Scenario C: WBA and HPA mixed (50:50)

4. Scheduling Algorithms

Scheduler A

Initial State: All PMs switched off. Create for every application that arrives an own VM with the parameters of the application (size, CPU and RAM). Fill PMs with VMs until they are full. Start a new PM if you need one. Kill a VM if you don't need it anymore (no applications running on it), kill a PM if there are no VMs running on it. Try to find an efficient way to decide on which PM you start a new VM.

Scheduler B

Initial State: All PMs switched off. If an application arrives, try to modify size, CPU and RAM of an existing VM to run the application. If no VM is running, create a new one (start a new PM if necessary). If the application has finished decrease the size, CPU and RAM of the VM. If no applications are running on a VM, shut down the VM. If no VM is running on a PM, shut down the PM. Try to get a maximum of utilization on every PM. Migration: Try to move applications from VMs to other VMs to get a better utilization and to use less PMs.

Scheduler C

Your own implementation.

Federation

If you have enough resources left, start to source-in applications. Ask a cloud-partner for applications. The partner offers you applications (randomly, based on the scenario) to source in. If you have not enough resources ask a cloud-partner to outsource applications (cloud-partner says yes/no randomly).

5. Input Data

For each scenario create a file as comma separated list (*.csv) for the arrival of the applications to the cloud. Provide at least 500 applications for meaningful results. (Hint: Write a small application to generate your input files.)

The file should have the following format:

```
Timestamp;Size;RAM;CPU;Duration
0;800;100;500;10
3;2548;1205;987;43
...
```

The timestamp specifies the time when an applications gets deployed to the cloud. You can set your intervals for duration and timestamp to a small unit (e.g. 10ms) to speed up your simulation.

Please keep in mind that this format is mandatory. If your implementation can’t read files of this format, we are not able to test your solution. We will provide you some example files for testing.

Provide a possibility (e.g. command line parameter or GUI) to configure the number of PMs and the number of cloud-partners.

6. Output Data

The output data will be a comma-separated-list (*.csv) with the following format:

```
Scheduler;Scenario;TotalPMs;TotalVMs;TotalDuration;TotalPowerConsumption;To
talInSourced;TotalOutSourced
A;A;...
B;A; ...
...
```

Additionally generate for each scenario and each scheduler (all combinations) output files of the following format:

```
Timestamp;TotalRAM;TotalCPU;TotalSize;RunningPMs;RunningVMs;TotalPowerConsu
mption;InSourced;OutSourced
...
```

7. Technology

You can use either Java or C# for your implementation.

8. Timetable

Fr. 12.04.2013 13:00 – 15:00	Presentation of the Lab Project Assignment	Seminarraum Argentinierstr. 8
02.05.2013 - 03.05.2013	Presentation of your concept (Architecture and Scheduler)	t.b.a.
Th. 23.05.2013 10:00 – 12:00	Presentation of mid-term results (10 min)	Seminarraum Argentinierstr. 8
03.06.2013 - 07.06.2013	Presentation of finished product	t.b.a.
Fr. 14.06.2013 11:30 – 15:30	Presentation of end results	Seminarraum Argentinierstr. 8

There will be two consultation hours for every group. All three team-members have to be present. Time slots (about 20 minutes) will be announced via the Tiss-Forum, where you have to register for one slot.

9. Contact

If you have questions regarding the assignment please use the Tiss-Forum. For personal questions you can contact the tutor (martin.jenny@tuwien.ac.at).