

# Cloud Computing Simulation for Smart Grid

Hanlin Chen

the Department of Electrical and Computer Engineering

Ohio State University

chen.8059@buckeyemail.osu.edu

April 11, 2019

## 1 Introduction

The growth of power data in smart grid motivates use of high performance cloud computing application. In the cloud model, all data processing, software and hardware are delivered to users via internet in forms of service such as platform as a service (Paas), software as a service (Saas) and infrastructure as a service (Iaas). Compared to traditional computing framework, the cloud computing can handle much higher volume of data from smart grid and provide promising power management to users. The performance of cloud model could be addressed by its processing speed and throughput. In addition, various factors such as hardware infrastructures, allocation of virtual environment and smart grid application are directly linked to these performance measurements. In this paper, we proposed a novel cloud based data center architectures to improve data processing in the smart power grid. The runtime simulation is done to investigate the effect of hardware configuration and allocation policy on execution of smart grid application using open source toolkit Cloudsim [1]. The structure of this paper is as follow. Section II reviews overall architectures of proposed cloud based data center for smart grid. Section III studies two common VM scheduling: time shared policy and space shared policy. Section IV includes all simulation results and time analysis. Section V gives the conclusion.

## 2 Cloud-Based Data Center Architectures

The basic cloud architectures enable supports for maximizing the availability and increasing demand of computing power and data storage[2]. In our experiment, the cloud computing service is provided by one power company. Therefore, only 1 Cloud Service Provider (CSP) or 1 data center is required in our simulation. The data center is accountable for providing seamless infrastructure services for all cloud users in the city. In order to achieved higher utilization of computing, the virtualization of hardware is used and allocating computing hardware on user's request. The advantage of virtualization is that the service provider is not obligated to allocate all hardware resources in advance and charge users based on usage of storage, memory and bandwidth. In addition, it can also prevent applications from

interrupting each other during execution, and provide higher data security. The detail of cloud-based data center architectures are displayed in figure 1.

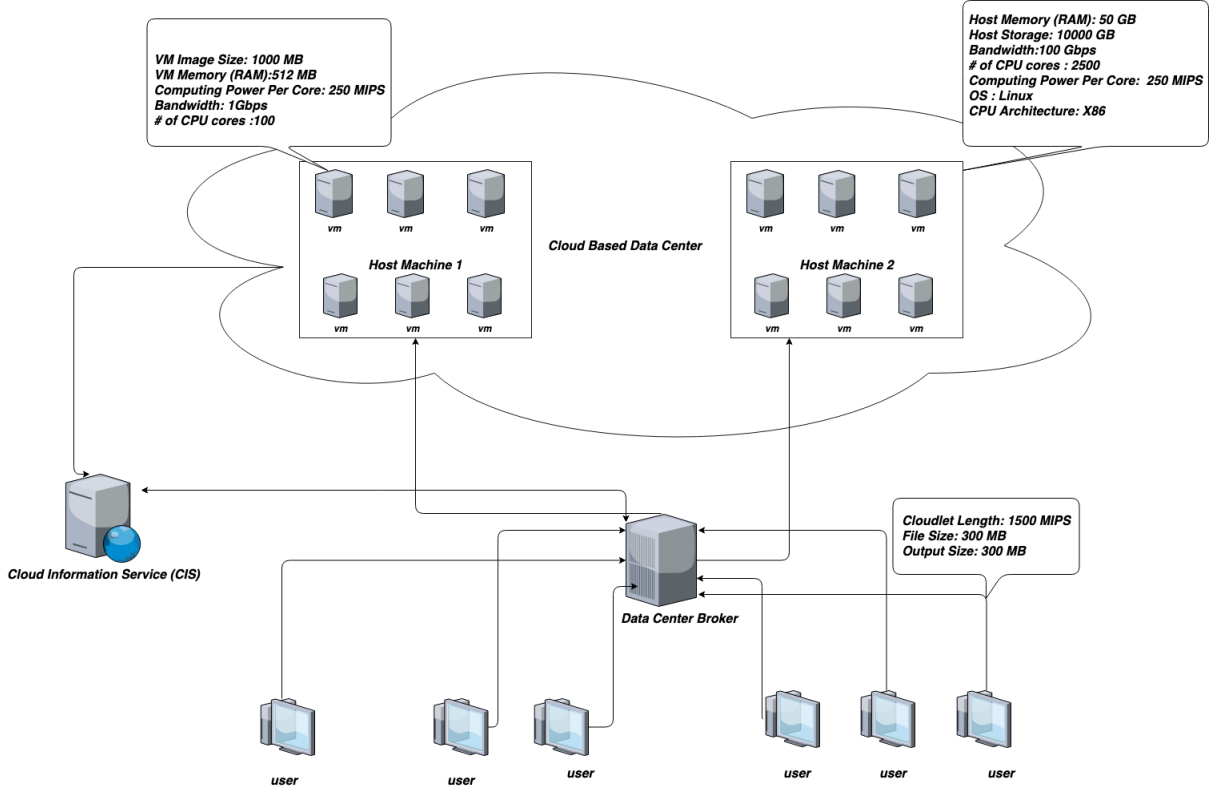


Figure 1: The Smart Grid Cloud Architecture

The data center has two host machines or servers and each host has the configuration: RAM = 50 GB, storage = 10000 GB , bandwidth = 10 Gbps and 2500 CPU cores with processing power of 250 MIPS per core. We divide the city where has 30000 users into 50 regions and each region is served by 1 VM. Therefore, number of VM is equal to 50. Each VM has the configuration: image size = 1000 MB, RAM= 512 MB and 100 CPU cores with processing power of 250 MIPS per core. The total storage and processing power of two host should be larger enough to handle all data generated from these 50 regions. The cloudlet is the job or task that users submit to cloud for processing and has fixed instruction length and data transfer size. For simplicity, in our simulation all user submit cloudlets with same configuration: length = 1500 MIPS , file size = 300 MB and output size = 300 MB.

As shown in figure 1, users submit the cloudlets to cloud for processing via data center broker. The broker search status of all computing resource and data center characteristics that previously registered by data center in cloud information service (CIS), then allocate virtual machines in the hosts to process cloudlets. After all cloudlets is completed, the broker destroys all allocated VMs.

### 3 Time-shared and Space-Shared Allocation Policy

Since the intuition of cloud provider is to grow company profit by maximizing user's satisfaction. The effective allocation of resources and tasks is key factor to improve user's experiences. Cloudsim provides time-shared allocation and space-shared allocation policies at both host level and VM level. These two policies works similar to First Come First Served (FCFS) and Round Robin (RR) algorithms[3]. At the host level, the host machine allocate processing power for each VM. At the VM level, the the VM allocated processing power for each cloudlet. When the time-shared policy is applied, the VMs or cloudlets can run simultaneously and share same processing power concurrently. There are not queuing delays among them. When the space-shared allocation policy is applied, the VMs or cloudlets can not share same processing power at same time. Only one group of VMs or cloudlets can run at time. Once they finish the execution, next group comes in and starts the execution.

The concept of time-shared allocation and space-shared allocation policies can be illustrated in following examples of the allocation modeling:

1. A data center has one host.
2. Each host has only 2 available CPU cores.
3. Users request 2 VMs (VM 1, VM 2)and each VM requires 2 CPU cores.
4. Users submit 8 cloudlets to cloud and each cloudlet requires 1 CPU core to process.
5. VM 1 runs cloudlet 1,2,3,4 and VM 2 runs cloudlet 5,6,7,8.

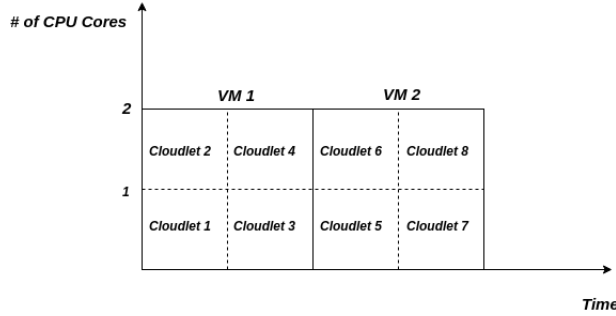


Figure 2: Space-Shared Allocation Policy for VMs and Cloudlets

Figure 2 illustrates that space-shared allocation policy applies at both host and VM level. At the host level, Each VM requires 2 CPU core to process their cloudlets . Hence VM 2 has to wait for VM 1 finishes, and then executes its cloutlets. Within each VM, since each cludlet requires 1 CPU core and they cannot share same CPU core at same time. Therefore, only 2 cloudlets can execute at give time instance. The second 2 cloutles have to wait until the first 2 cloudlets finish, and then start their own executions.

Figure 3 illustrates space-shared allocation policy for VMs and time-shared allocation for cloudles. At the host level, VM 2 has to wait for VM 1 finish in order to start its executions. At the VM level, the time-shared policy allow them to share same CPU core while each cloudlet require 1 CPU core. Therefor 2 cloudlets can share one CPU core, and 4 cloudlets can execute simultaneously. Since each CPU core is shared by 2 cloudlets, the execution time of each cloudlet become longer compared to previous scheme in figure 2.

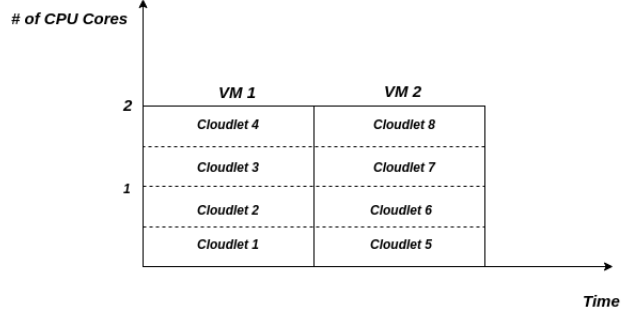


Figure 3: Space-shared Allocation for VM and Time-Shared Allocation for Cloudlets

Figure 4 illustrates time-shared allocation policy for VMs and cloudlets. At the host level, 2 VMs can share 2 CPU cores concurrently. At the VM level, 2 cloudlets share 1 CPU cores at same time. Overall, 1 CPU core is shared by 4 cloudlets, consequently it takes much longer to execute them compared to the scheme in Figure 3.

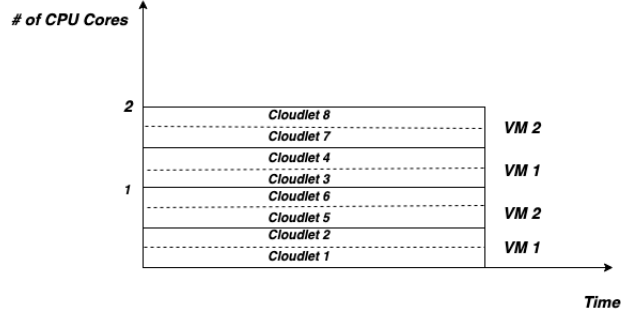


Figure 4: Time-Shared Allocation Policy for VM and Cloudlets

Figure 5 illustrates that time-shared allocation for VMs and space-shared allocation for cloudlets. At the host level, 2 VMs can share 2 CPU cores at same times. At the VM level, since each cloudlets can not share the same CPU core, therefore only 2 cloudlet are allow to run at give time instance, the second 2 cloudlets must wait for first 2 cloudlets finish and then start execution.

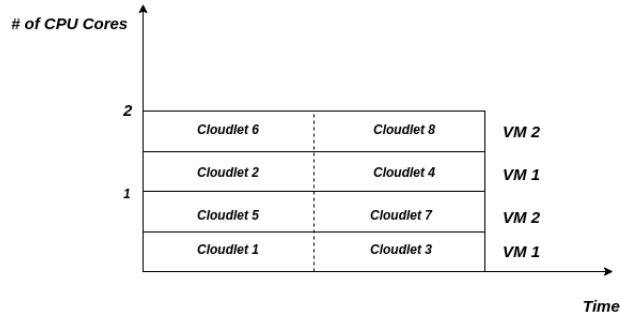


Figure 5: Time-Shared Allocation Policy for VM and Cloudlets

## 4 Simulation Results

## References

- [1] Cloud Computing and Distributed Systems (CLOUDS) Laboratory, “Cloudsim.” [Online]. Available: <http://www.cloudbus.org/cloudsim/>
- [2] K. Nagamani, C. G. V. N. Prasad, and K. S. Chatrapati, “A novel framework for optimal component based data center architecture,” in *2016 International Conference on Information Communication and Embedded Systems (ICICES)*, Feb 2016, pp. 1–10.
- [3] H. Gibet Tani and E. Chaker, “Cloud computing cpu allocation and scheduling algorithms using cloudsim simulator,” vol. 6, pp. 1866–1879, 08 2016.