Hierarchical Queue-Based Task Scheduling

Wanqing You, Kai Qian, and Ying Qian

***Abstract*—The concepts of Cloud Computing provide users with infrastructure, platform and software as service, which make those services more accessible for people via Internet. To better analyze the performance of Cloud Computing provisioning policies as well as resources allocation strategies, a toolkit named CloudSim was proposed. With CloudSim, the Cloud Computing environment can be easily constructed by modeling and simulating cloud computing components, such as datacenter, host, and virtual machine. A good scheduling strategy is the key to achieve load balancing among different machines as well as to improve the utilization of basic resources. Recently, the existing scheduling algorithms may work well in some presumptive cases in a single machine; however they are unable to make the best decision for the unforeseen future. In real world scenario, there would be numbers of tasks as well as several virtual machines working in parallel. Based on the concepts of multi-queue, this paper presents a new scheduling algorithm to schedule tasks with CloudSim by taking into account several parameters, including the machines' capacity, the priority of tasks and the history log.**

***Index Terms*—Hierarchical queue, load balancing, cloudsim.**

I. Introduction

Cloud computing is an emerging computation model to use existing computing resources that are delivered as a form of service. Cloud computing, which provides computing resources, data and application of clouds, has many attractive benefits, such as scalability and reliability. However, with the popularity of computing service, rapidly increasing of users scale causes a huge amount of resources consumption. It's a critical problem to improve resources utility and ensure high system performance. Therefore, it is important to evaluate the performance of cloud computing environment to predict valid cost to manage the cloud computing system. Tools such as SimJava [1] and GridSim [2] can be used to measure the performance. They are well-known simulation tools in grid computing but they do not support the features such as virtualization of cloud computing.

CloudSim is an extensible simulation toolkit that enables modeling and simulation of cloud computing systems and application provisioning environments [3]. Aiming to evaluate any cloud products in a timely, repeatable and controllable state to assure a good performance, CloudSim has the ability to model the components in cloud computing environment.

CloudSim is the only tool, which can evaluate the performance of this environment and it is based on SimJava and GridSim. It is suitable to simulate the situation with large amount of devices and data in cloud computing. Also, it can simulate the virtualization of computing nodes, network devices, and storage units.

Among all the components in cloud computing environment, load balancing is a really important one. Load balancing possesses a well-defined resources allocation policy thus to have a maximum throughput or a minimum response time. In CloudSim, there are two default allocation policies: Round Robin algorithm in a time-shared system and First Come First Serve algorithm in a space-shared system. These two algorithms may work well to simple scenarios with a small number of tasks. However, situations are much more complicated in a real cloud computing environment. There are several elements interact with each other to influence the final performance of a particular scheduling policy and the utilization of different kinds of resources.

II. Related Work

Some algorithms have been analyzed, improved and simulated in CloudSim to achieve the load balancing, which mainly focus on the minimum response time in cloud computing environment.

In Soumya Ray's work, they analyzed a few load balancing algorithms in cloud computing environments. They firstly identified qualitative components for simulation in cloud environment and then presented the different response time of various load balancing algorithms [4].

In [5], Ajay proposed a Dynamic Round Robin algorithm to optimize the load. Their contributions include two parts: using CloudSim to set up the cloud computing simulation platform, and varying the vital parameters, which shade important impact on load balancing. The result showed that the load had been optimized.

Amit [6] adopted an adaptive QoS (Quality of Service) aware virtual machine provision to achieve a full utilization of resources. By allocating tasks to different queues and set a high priority for urgent tasks, it reached a high throughput compared to other ways. The work of Amit as well as the work of Stefan [7] were focusing on the scheduling of virtual machines. Others may emphasize on CPU [8].

Service Level Appointment (SLA) is a general accepted standard to assess the cloud services. However, as related work mentioned above, most existing work considers a single SLA element, to improve resources utilization, multiple SLA parameters are interactional. In Rajnikant's work [9], it considers multiple SLA parameters such as memory, network bandwidth, and required CPU time to improve the performance and efficiency of algorithm. Most similar, Tian's dynamic scheduling algorithm called Least Integrated-load First (LIF) [10] also take into account multiple-dimensional resources.

Manuscript received October 30, 2013; revised December 31, 2013.

W. You and K. Qian are with the Department of Computer Science, Southern Polytechnic State University, Marietta, GA, USA (e-mail: wyou@spsu.edu, kqian@spsu.edu).

Y. Qian is with the Department of Computer Science & Technology, East China Normal University, Shanghai, China (e-mail: yqian@cs.ecnu.edu.cn).

DOI: 10.7763/JACN.2014.V2.98

III. Architecture of CloudSim

CloudSim is a simulation tool to figure out the performance of cloud computing systems.

Fig. 1 shows the architecture of CloudSim. CloudSim is based on discrete event engine SimJava, by extending the programming model of GridSim to support the research and development of cloud computing. CloudSim provides support for data center environment based on virtualized cloud, which includes dedicated management interface of virtual machines and modeling and simulation of memory, storage, and bandwidth. During simulation, CloudSim could manage the instances as well as the execution of core entities including VMs (Virtual Machines), host, data center and application [3]. Initial releases of CloudSim used SimJavato support several core functionalities, such as queuing and processing of events, creation of Cloud system entities, communication between components, and management of the simulation clock. However in the current release, the SimJava layer has been removed in order to allow some advanced operations that are not supported by it [3].

The CloudSim provides support for modeling and simulation of virtualized Cloud-based data center environments The fundamental issues, such as provisioning of hosts to VMs, managing application execution, and monitoring dynamic system state, are handled by CloudSim. A Cloud provider, who wants to study the efficiency of different policies in allocating its hosts to VMs (VM provisioning), could implement his strategies by CloudSim. Such implementation can be done by programmatically extending the core VM provisioning functionality.

On top of the CloudSim software is the User Code that exposes basic entities for hosts (number of machines, their specification, and so on), applications (number of tasks and their requirements), VMs, number of users and their application types, and broker scheduling policies. By extending the basic entities given at this layer, a Cloud application developer can perform the following activities: (i) generate a mix of workload request distributions, application configurations; (ii) model Cloud availability scenarios and perform robust tests based on the custom configurations; and (iii) implement custom application provisioning techniques for clouds and their federation [3].

CloudSim has several features that are different from existing tools such as SimJava and GridSim. First of all, CloudSim can support for large scale and virtual machines. Also, it offers a self-contained platform to simulate data centers, service brokers, scheduling, and allocation policies to specialized environment of cloud computing. Lastly, CloudSim provides availability of virtualization engine and flexibility to switch between space-shared and time-shared allocation.

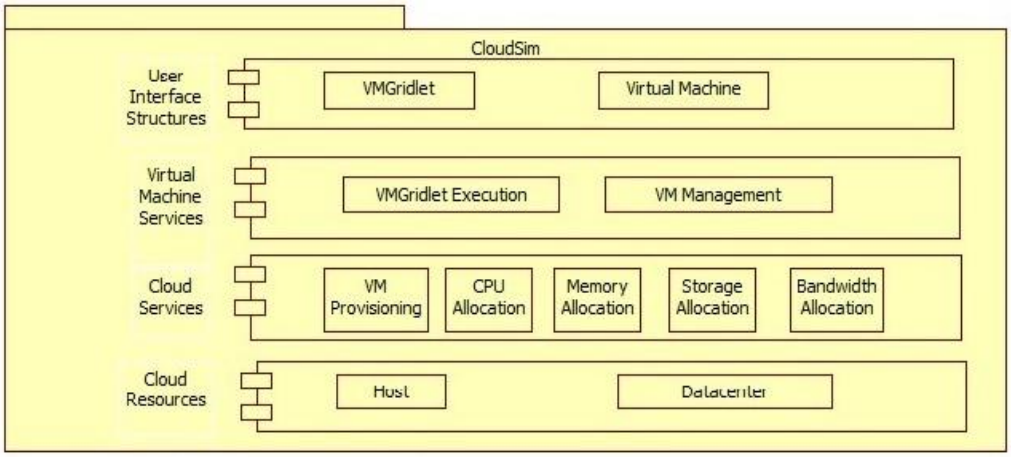

Fig. 1. CloudSim architecture.

IV. Scheduling

Inspired by the concept of multi-queue, we present a queue-based task scheduling algorithm which devotes to achieving a minimum completion time of job being scheduled so far as well as to striking the load balancing of all virtual machines. A global queue is adopted to store all the new incoming tasks. In our algorithm, there are three different virtual machines, the power and capacity of which are shown in the Fig. 2. There are three local queues correspond to three different virtual machines.

Fig. 2 shows the rough idea about the configuration we will implement in the algorithm. The main parameters are MIPS (Million Instructions per Second), PEs (or CPU cores), Ram (or memory) and bandwidth.

To optimize the utilization of all resources and achieve load balancing at the same time, this paper proposes a scheduling algorithm to detect the status of all virtual machines in real time. The algorithm, based on the history of new coming jobs, tries to dispatch unscheduled jobs in global queue actively to avoid a long idle time on virtual machines. The core concept is shown in the Fig. 3, and pseudo code is shown in Fig. 4. Take an example. Suppose job 1 has a length of 20MI (Million Instruction), first calculate its completion times in three virtual machines. If the maximum completion time is shorter than the average arriving time and average processing time according to Log, the job 1 should be scheduled now. To do the dispatching, first compare the completion time of each virtual machine by using the formulation $t = \frac{L_i}{C_i}$, where L_i is the whole list of instructions that machine *i* need to process (Unit: MI, Million Instruction) and C_i (Unit: MIPS, Million Instruction Per Second) the capacity of machine *i*. In this example, virtual machine 1 (vm_1) has the minimum completion time among three virtual machines. Therefore, job *i* is allocated to virtual

machine 1.

Virtual Machine	*MIPS*	*PEs*	*Ram*	*Bandwidth*
Virtual Machine 1	300	1	2048	1000
Virtual Machine 2	200	1	1024	1000
Virtual Machine 3	100	1	512	1000

Fig. 2. VM configurations.

We use a variable utilization rate to measure the loading of each virtual machine. The formulation is as simple as this:

$$\text{utilization\_rate}_i = \frac{p_i}{v_i + p_i}$$

where p_i represents the processing time of vm_i and v_i is the idle time of vm_i.

To assess the performance of the whole system, we consider the completion time of the tasks scheduled so far.

$$\text{Completion\_time} = \sum c_{p_i}$$

where c_{p_i} means the completion time of job*i*.

The pseudo code proposed by this paper is shown as below.

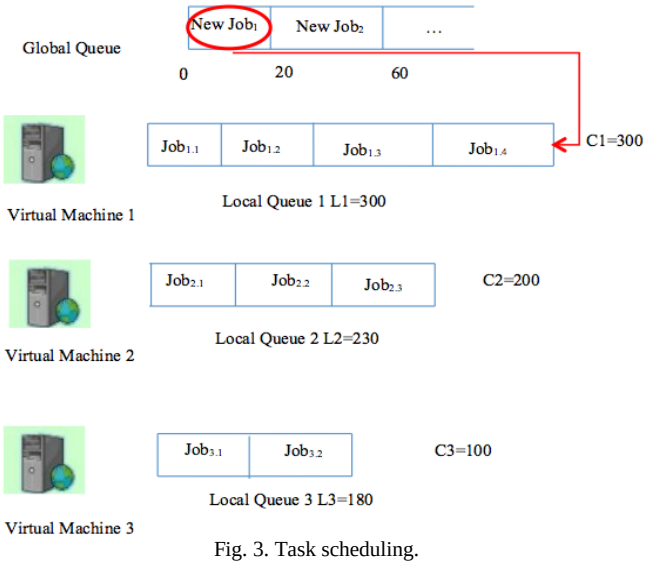

Fig. 3. Task scheduling.

V. Conclusion

With CloudSim, it is easy to simulate a cloud computing environment. Especially, it would act as a test-bed to test the proposed product before putting into use without a considerable cost. Based on the API provided by CloudSim, this paper proposes a queue-based task scheduling algorithm, thus to detect the status of every machine to make more reasonable decision. Offline global task scheduling is implemented in CloudSim environment. To allocate tasks to various virtual machines, we have considered virtual machines with different configurations (MIPS, memory, and storage). As a more comprehensive view, to simulate more real word environment, SLA should be taken into consideration.

```
//the new incoming tasks will be put into global queue
//waiting to be scheduled
//pi: the processing time of taski
//aveTime: the average processing time of task
//reachTime: the average reaching time of task

globalQueueScheduling()
begin
   //add new coming task to global queue
   globalQueue.add(newTaskList);
   //calculate the average reaching time
   t1=aveReachTimeLog (taskList);
   //calculate the average completed time
   t2=aveCompletedTimeLog(taskList);
   //select the task from global queue which has the
//minimum processing time
   min=select_min_time(taskList);
   //decide whether we should schedule the job now
   if (min<t1&&min<t2)
       allocate_job();//allocate task to vm
end

allocate_job()
begin
   minLoad=vm[0].vmLength/vm[0].vmCapacity;
   minIndex=0;
   for i from 1 to 2
      if(vm[i].vmLength/vm[i].vmCapacity<minLoad)
         minIndex=i;
         continue;
   allocate job to vm[minIndex];
end
```

Fig. 4. Pseudo code.

References

[1] F. Howell and R. Mcnab, "SimJava: a discrete event simulation library for java," in *Proc. International Conference on Web-Based Modeling and Simulation*, 1998, pp. 51-56.

[2] R. Buyya and M. Murshed, "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing," *The Journal of Concurrency and Computation: Practice and Experience*, vol. 14, no. 13-15, pp. 1175-1220, Nov, 2002.

[3] R. N. Calheiros, R. Ranjan, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithm," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, January 2011.

[4] S. Ray and A. D. Sarkar, "Execution analysis of load balancing algorithms in cloud computing environments," *International Journal on Cloud Computing: Services and Architecture*, vol. 2, no. 5, pp. 1, October 2012.

[5] A. Gulati and R. K. Chopra, "Dynamic round robin for load balancing in a cloud computing," *International Journal of Computer Science and Mobile Computing*, vol. 2, no. 6, pp. 274-278.

[6] A. K. Das, T. Adhikary, and C. S. Hong, "An intelligent approach for virtual machine and QoS," in *Proc. Cloud Computing, International Conference on Information Networking*, 2013, pp. 462-467.

[7] S. Boettger, C. Lara, T. Breitner, U. Kebschull, and V. Lindenstruth. Virtual machine scheduling for special purpose clusters. [Online]. Available:

http://www.kip.uni-heidelberg.de/user/boettger/2010BoettgerVirtualDedicatedCluster.pdf.
[8] P. Sharma and M. Gahlawat, "Analysis and performance assessment of CPU scheduling algorithms in cloud using CloudSim," *International Journal of Applied Information Systems*, vol. 5, no. 9, July 2013.
[9] C. S. Pawar and R. B. Wagh. Priority based dynamic resource allocation in cloud computing. [Online]. Available: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6481227.
[10] W. Tian, X. Liu, C. Jin, and Y. Zhong, "LIF: a dynamic scheduling algorithm for cloud data centers considering multi-dimensional resources," *Journal of Information and Computational Science*, vol. 10 no. 12, pp. 3925–3937, August 2013.

Wanqing You was born in Jinjiang, China, in 1990. She received bachelor degree from Xiamen University in 2013.

She is currently a master student and research assistant in Department of Computer Science at Southern Polytechnic State University. Her research interests include mobile, network security, software defined networking (SDN), and cloud computing.

Kai Qian was born in Shanghai, China, in 1947. He achieved his Ph.D of computer science from University of Nebraska, Linton, USA, in 1990.

He is a full professor of computer science at Southern Polytechnic State University, USA. His research interests include mobile and network security, advanced learning technology.

Ying Qian was born in Shanghai, China, on Jan. 31st 1978. She got bachelor Degree from Department of Electronics Engineering, Shanghai Jiao Tong University, Shanghai, China in 1998. She received her master and Ph.D. degree in Department of Electrical & Computer Engineering from Queen's University, Kingston, Ontario, Canada, in 2005 and 2010 respectively.

She is an associate professor in the Department of Computer Science and Technology, at East China Normal University, Shanghai, China. Before she joined East China Normal University in 2013, she was a computational scientist in the Supercomputer Laboratory at King Abdullah University of Science and Technology from 2009. Her research interests include software defined network, high-performance scientific computation, and parallel programming.