Efficient Task Scheduling Algorithms for Cloud Computing Environment

S. Sindhu[1] and Saswati Mukherjee[2]

[1] Research Scholar,
Department of Information Science and Technology
sindhu.nss@gmail.com
[2] Professor
College of Engineering,
Guindy, Anna university, Chennai
msaswati@yahoo.com

Abstract. Cloud Computing refers to the use of computing, platform, software, as a service. It's a form of utility computing where the customer need not own the necessary infrastructure and pay for only what they use. Computing resources are delivered as virtual machines. In such a scenario, task scheduling algorithms play an important role where the aim is to schedule the tasks effectively so as to reduce the turnaround time and improve resource utilization. This paper presents two scheduling algorithms for scheduling tasks taking into consideration their computational complexity and computing capacity of processing elements. CloudSim toolkit is used for experimentation. Experimental results show that proposed algorithms exhibit good performance under heavy loads.

Keywords: Cloud computing, IaaS, Private Cloud, Task Scheduling.

1 Introduction

Cloud computing is the latest buzzword in the IT industry. It is an emerging computing paradigm with foundations of grid computing, utility computing, service oriented architecture, virtualization and web 2.0. The user can access all required hardware, software, platform, applications, infrastructure and storage with the ownership of just an internet connection. "A Cloud is a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers" [1]. Some of the applications of cloud computing are on-line gaming, social networking, scientific applications. One of the key issues in public clouds are that of security and privacy [2]. In public clouds data centers hold end-users data which otherwise would have been stored on their own computers. Hence there is a growing demand for private clouds. A private cloud is one which is owned and operated within the firewalls of an organization. It allows an organization to manage its internal IT infrastructure effectively and provide services to its local users [3]. A

A. Mantri et al. (Eds.): HPAGC 2011, CCIS 169, pp. 79–83, 2011.

private cloud should support efficient resource allocation policies that adhere to the specific requirements of an organization (high availability, reliability, QoS). Cloud computing relies heavily on virtualization. Clouds are virtual clusters. Hence efficient scheduling of tasks and virtual machines across the various heterogenous physical machines is a crucial task, especially in a private cloud environment where the resources are limited. Very limited research have been done so far in scheduling, in a private cloud environment, except for some generic algorithms being adopted in various tools like Eucalyptus and OpenNebula, [4][5], that are being used as infrastructure to build private and hybrid clouds. We focus on the task scheduling problem in a private cloud environment in this paper. Some performance metrics like high throughput, low response time, minimum makespan and flowtime are the conventional metrics used for task scheduling. Here we present two scheduling algorithms for scheduling tasks in a private cloud environment, where the main aim is to obtain a minimum makespan. We have used the CloudSim simulator to implement our proposed algorithms since it provides the necessary environment to test the scheduling algorithms in a repeated and controlled environment.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 describes the proposed scheduling algorithms. Section 4 briefs the experimental setup, results and discussion. Finally, Section 5 gives the conclusion and future work.

2 Related Work

Scheduling policies in a cloud environment vary depending on the deployment model of the cloud. This section provides a brief review of some related work done in scheduling in a cloud. Ye Hu, Johnny Wong, Gabriel Iszlai, Marin Litoiu [6] have proposed a new probability dependent priority algorithm to determine the minimum number of servers required to execute the jobs, considering jobs in two different classes, such that the SLAs of both job classes are met. In [7] an optimized algorithm for task scheduling based on Activity Based Costing, is presented, that selects a set of resources, to schedule the tasks, such that the profit is maximized. A heuristic method to schedule bag-of-tasks (tasks with short execution time and no dependencies) in a cloud is presented in [8] so that the number of virtual machines to execute all the tasks, within the budget, is minimum and at the same time speedup is maximum. Hybrid cloud is a model which combines a private cloud and public cloud. That is, during peak load when there are not sufficient resources to execute a task in a private cloud, outsource the same to a public cloud provider and get it done. An optimal scheduling policy based on linear programming, to outsource deadline constrained workloads in a hybrid cloud scenario is proposed in [9]. The scheduling policy used in Eucalyptus [4] are First Fit and Round Robin. In OpenNebula, Haizea can be used as the scheduler backend, which supports advance reservations in the form of a lease, [5]. But none of the existing algorithms have considered the computational complexity of tasks for scheduling. The heuristic algorithms for scheduling jobs on computational grids [10] provides a framework for our investigation. Our proposed work focuses on scheduling tasks in a private cloud environment.

3 Scheduling Algorithms

A good scheduling algorithm should lead to better resource utilization and better system throughput. To formulate the problem, let us consider C_n={ C_1, C_2, …,C_n} be 'n' cloudlets, V_m={ V_1, V_2, …,V_m} be 'm' virtual machines and PE_p={ PE_1, PE_2,…, PE_p} be the processing elements across all the hosts in a datacenter. Makespan is defined as the finishing time of the last job in a set of jobs. Let CT_c be the completion time when the last cloudlet 'c' finishes processing. Our objective is to minimize CT_c.

3.1 Longest Cloudlet Fastest Processing Element (LCFP)

In this algorithm the computational complexity of the cloudlets are considered while making scheduling decisions. The lengthier cloudlets are mapped to Processing Elements (PEs) having high computational power so as to minimize the makespan. In this algorithm the longer jobs finishes quickly when compared with the FCFS where processing requirement of jobs are not considered while making scheduling decisions.

Algorithm

1. Sort the cloudlets in descending order of length.
2. Sort the PEs across all the hosts in descending order of processing power.
3. Create virtual machines in the sorted list of PEs by packing as many VMs as possible in the fastest PE.
4. Map the cloudlets from the sorted list to the created VMs.

3.2 Shortest Cloudlet Fastest Processing Element (SCFP)

In this algorithm the shorter cloudlets are mapped to PEs having high computational power so as to reduce flowtime (sum of completion time of a set of jobs) while at the same time taking into consideration that longer jobs are not starved.

4 Implementation

4.1 Cloudsim Simulation Environment

Cloudsim [11] is a generalized, extensible simulation framework that enables modeling, simulation, and experimentation of Cloud computing infrastructures and application services. In cloudsim, Datacenter component is the main hardware infrastructure that provide services for servicing user requests. A Datacenter is composed of a set of hosts, which are responsible for managing VMs during their life cycles. Host is a component that represents a physical computing node in a Cloud. It is assigned a pre-configured processing capability (expressed in million of instructions per second – MIPS), memory, storage. Virtual Machine Provisioner component is responsible for allocation of application-specific VMs to Hosts in a Cloud-based data center. The default policy implemented by the VM Provisioner is a straightforward policy that allocates a VM to the Host in First-Come-First-Serve (FCFS) basis. In Cloudsim user jobs are called as cloudlets. Each cloudlet is assigned an id, length. Its assumed that, larger the length of cloudlets, the higher is the complexity. Cloudlets can be bound to a Virtual Machine explicitly as specified by user or dynamically at run-time. Cloudlet-Scheduler is the component responsible for mapping the cloudlets to the VMs. The default scheduling policy used by CloudletScheduler is First Come First Served (FCFS) i.e., each cloudlet in the queue is mapped to the list of created virtual machines on a FCFS basis. It does not consider the processing requirement of a job while making the scheduling decision. Also FCFS does not consider the processing requirement of user jobs. It suffers from starvation as lengthy jobs ahead in the queue delay shorter jobs with high response time and also results in poor resource utilization.

4.2 Experimentation Results

The algorithms for simulation are implemented on an Intel Dual Core machine with 320GB HDD and 2GB RAM on Cent OS 5.5. The experiments are conducted on a simulated Cloud environment provided by CloudSim. The speed of each processing element is expressed in MIPS (Million Instructions Per Second) and the length of each cloudlet is expressed as the number of instructions to be executed . The simulation environment consists of one Data Center with two hosts having three and two Processing Elements respectively. Each Processing Element is assigned varying computing power (varying MIPS). The algorithms are tested by varying the number of cloudlets from 10 to 50 and also randomly varying the length of cloudlets. Also, the number of VMs used to execute the cloudlets, are varied accordingly. The overall makespan to execute the cloudlets is used as the metric to evaluate the performance of the proposed algorithms. It has been observed that, for smaller number of tasks, all the three algorithms exhibit more or less similar performance since the length of the queued cloudlets are less. But as shown in Fig. 1, as the number of tasks increase, LCFP exhibits better performance when compared to SCFP and FCFS since longer tasks complete faster thereby reducing the makespan.

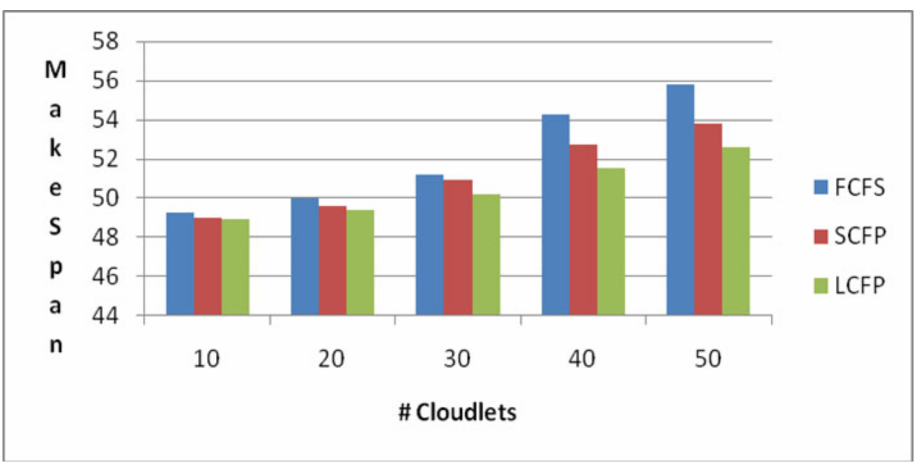

Fig. 1. Graph between number of cloudlets submitted and the makespan for FCFS, LCFP and SCFP

5 Conclusion

Cloud Computing is an upcoming research area. Scheduling of tasks, so as to improve resource utilization, while at the same time, considering the QoS of tasks, is an important problem, in a private cloud environment, since in a private cloud, the resources are limited. This paper explores the use of two sscheduling algorithms for scheduling tasks which considers the processing requirement of a task and the computational capacity of a resource while making scheduling decisions. In future we would like to experiment with more algorithms that use heuristic methods for scheduling and also consider the priority of tasks.

References

1. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. In: Future Generation Computer Systems, vol. 25(6), pp. 599–616. Elsevier Science, Amsterdam (2009)
2. Dikaiakos, M.D.: Cloud Computing: Distributed Internet Computing for IT and Scientific Research. IEEE Transactions on Internet Computing 13(5), 10–13 (2009)
3. Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I.: Virtual Infrastructure Management in Private and Hybrid clouds. IEEE Transactions on Internet Computing 13(5), 14–22 (2009)
4. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., So-man, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-source Cloud-computing system. In: IEEE International Symposium on Cluster Computing and the Grid, CCGrid 2009 (2009)
5. Open Nebula, `http://www.opennebula.org`
6. Hu, Y., Wong, J., Iszlai, G., Litoiu, M.: Resource provisioning for cloud computing. In: Conference of the Centre for Advanced Studies on Collaborative Research, CASCON 2009, New York (2009)
7. Cao, Q., Wei, Z.-B., Gong, W.-M.: An Optimized Algorithm for Task Scheduling Based On Activity Based Costing In Cloud computing. In: 3rd International Conference on Bioinformatics and Biomedical Engineering, Beijing (2009)
8. Silva, J.N., Veiga, L., Ferreira, P.: Heuristics for Resource Allocation on Utility Computing Infrastructures. In: 6th International Workshop on Middleware for Grid Computing, New York (2008)
9. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In: 3rd IEEE International Conference on Cloud Computing, Miami (July 2010)
10. Abraham, A., Buyya, R., Nath, B.: Nature's Heuristics for Scheduling Jobs on Computational Grids. In: 8th IEEE International Conference on Advanced Computing and Communications, ADCOM 2000, India (December 2000)
11. Calheiros, R.N., Ranjan, R., De Rose, C.A.F., Buyya, R.: CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services (2009)