Cost-Effective User Priority Based Task Scheduling in Cloud

[I]Dharmendra D P, [II]Rampur Srinath
[I]PG Student, Dept. of Information Science & Engg., The National Institute of Engg., Mysuru, India
[II]Associate Prof., Dept. of Information Science & Engg., The National Institute of Engg., Mysuru, India

Abstract
Cloud computing is emerging as an incipient paradigm of immense-scale distributed computing. In order to maximize the utility of cloud computing, we require an efficient scheduling algorithm. The traditional Min-Min algorithm is a simple and efficient algorithm that engenders a better schedule that minimizes the total completion time of tasks. However the drawback of it is, load imbalanced, which is one of the central issues for cloud providers. Cloud providers offer computer resources to users on a pay-per-use base. In order to accommodate the ordinant dictations of different users, they may offer different calibers of quality for accommodations. Then the cost per resource unit depends on the accommodations culled by the utilizer. In reciprocation, the utilizer receives guarantees regarding the provided resources. To observe the promised guarantees, utilizer-priority as well as deadline was considered in our proposed algorithm so that user's demand could be slaked more thoroughly. In this paper, a Cost effective Utilizer priority algorithm is introduced on the substructure of Min-Min algorithm, which reduces the composition span and increment the resource utilization and additionally reduces the resource cost.

Keywords
Cloud computing, scheduling, load balance, QoS, user demands.

I. Introduction

Currently Cloud computing has evolved as great potential technology that is kenned as a provider of dynamic accommodations utilizing profoundly and immensely colossal scalable and virtualized resources over the Internet which is a prevalent accessing and communicating medium [1]. Cloud is subject to Utilizer Requisite, Load Balance and other constraints that have direct effect on utilizer consumption of resources controlled by cloud provider. In order to utilize the potency of cloud computing thoroughly, we require an efficacious and efficient task scheduling algorithm. Task scheduling algorithm is responsible for dispatching tasks submitted by users to cloud provider onto heterogeneous available resources.

Cloud computing systems are commonly divided into three types namely Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [2]. The deployment of cloud services is as shown in figure1. It's a challenge in cloud computing that how to efficaciously schedule resources in cloud environment, especially in IaaS. In this environment, computing resources form a shared pool, in which resources are allocated and reallocated to users upon users' demand, and resources are provisioned by multiple suppliers. In IaaS, infrastructure Service suppliers and users form a resource market, in which users want to get the lowest price and best quality of accommodation. In IaaS, resource scheduling is not an isolated quandary. Resource Scheduling should sanction for how to cooperate with virtual machine management.

Any discussion of cloud computing typically commences with virtualization. Simply put, it's the process of engendering a virtual, rather than physical, version of something. Virtualization can apply to computers, operating systems, storage contrivances, applications, or networks. However, server virtualization is at the heart of it. A virtual computer system referred as "virtual machine" (VM): a tightly isolated software container with an operating system and application inside. Each self-contained VM is thoroughly independent. Putting multiple VMs on a single computer enables several operating systems and applications to run on just one physical server, or "host". A thin layer of software called a hypervisor decouples the virtual machines from the host and dynamically allocates computing resources to each virtual machine as needed. Since the Cloud computing is multi-tenant in nature, it deals with different kinds of virtualized resources, hence scheduling places a paramount role in cloud computing. In cloud, utilizer may use hundreds of thousands virtualized resources for each utilizer task. Hence manual scheduling is not a feasible solution.

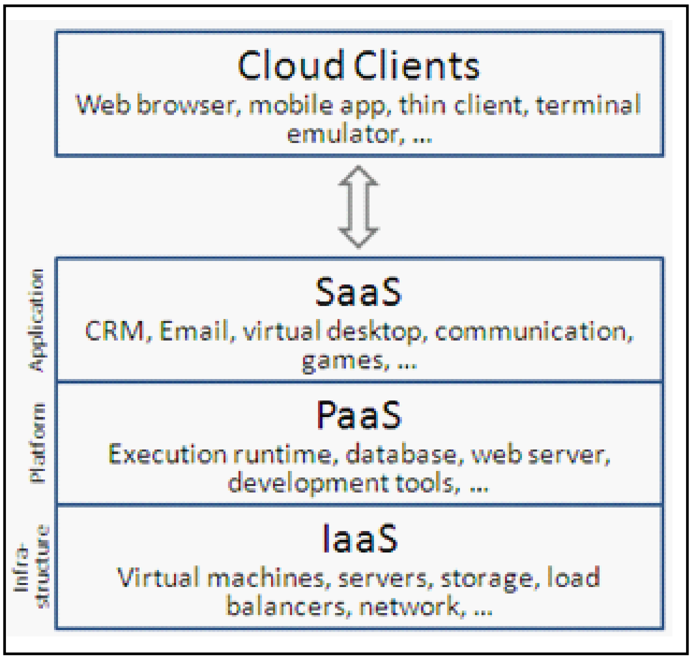

Fig. 1: Deployment of cloud services

The main intention of a cloud task scheduling algorithm is to abbreviate overall the completion time of all tasks submitted by users, enhance the utilization of cloud resources and satisfy requisites of different users. Most of the traditional scheduling approaches largely ignore utilizer-priority issue, e.g. Min-Min, Max-min, Round-Robin, they may not acclimate to the cloud environment well as Cloud computing is not only a modeling technique but an economic model [3]. Cloud providers offer computer resources to users on a pay-as-you-go base. In order to accommodate the demands of different users (e.g. VIP utilizer, mundane utilizer), they may offer different levels of

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

accommodations.

Cloud simulators are required for cloud system testing to decrease the complexity and separate quality concerns. They enable performance analysts to analyze system behavior by focusing on quality issues of specific component under different scenarios [7]. CloudSim is a simulation application which enables seamless modeling, simulation, and experimentation of cloud computing and the application services, proposed by [8,9] due to the problem that existing distributed system simulators were not applicable to the cloud computing environment.

CloudAnalyst was derived from CloudSim and extends some of its capabilities proposed in . This simulator can be applied to examining the behavior of large scaled Internet application in a cloud environment and separating the simulation experimentation exercise from a programming exercise. It also enables a modeler to repeatedly perform simulations and to conduct a series of simulation experiments with slight parameters variations in a quick and easy manner. The CloudAnalyst architecture is shown in Figure 2 [10].

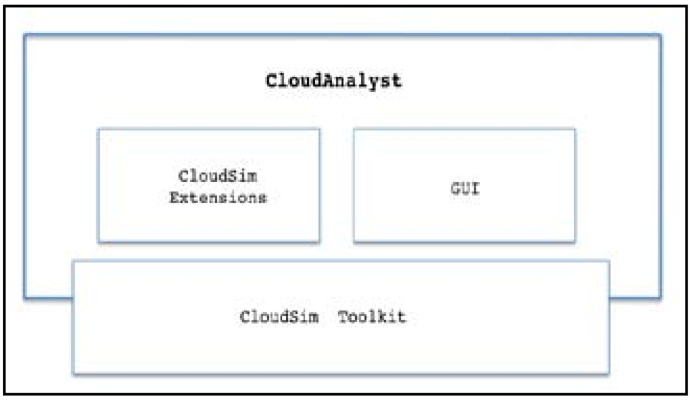

Fig. 2: Architecture of CloudAnalyst

II. Related Work

It is well known that, the intricacy of a general scheduling problem is NP-Complete. The scheduling quandary becomes more challenging because of the unique characteristics belonging to Cloud computing [4]. Some of these characteristics are the following:

- The high heterogeneity of resources: Cloud systems act as sizably voluminous virtual supercomputers, yet the resources could be very disparate, ranging from laptops, desktops, supercomputers and even minuscule contrivances of circumscribed computational resources.
- The high heterogeneity of tasks: Tasks reaching to any Cloud system are diverse and heterogeneity in terms of their utilizer demands.
- User-priority: this characteristic is a paramount issue in Cloud computing. Utilizer-priority must be considered during task scheduling with guarantee that users who pay more can relish better accommodation.

The Min-Min algorithm is simple and still stands as the base of present cloud scheduling algorithms. It commences with a set S of all unmapped tasks. Then the resource R which has the minimum completion time for all tasks is found. Next, the task T with the minimum size is culled and assigned to the corresponding resource R (hence the designation Min-Min). Last, the task T is abstracted from set S and the same procedure is reiterated by Min-Min until all tasks are assigned (i.e., set S becomes empty). But the difficulty with this is, at last the task with maximum size which left will be allocated to the resource with low capacity.

A load balancing algorithm is introduced on the ground of Min-Min algorithm in order to reduce the make span and increment the resource utilization [5]. Concurrently, Cloud providers offer computer resources to users on a pay-per-use base. In order to accommodate the authoritative ordinances of different users, they may offer different levels of quality for accommodations. But here the priority and deadline of the tasks is not considered. This algorithm only takes the initial research on task scheduling in Cloud platform. However many issues remain open. Further amendment should be done to handle more perplexed scenario involving dynamic factors such as dynamically transmuting cloud environment and other QoS attributes such as deadline.

A computational Grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [6]. It is a shared environment implemented via the deployment of a sedulous, standards-predicated accommodation infrastructure that fortifies the engenderment of, and resource sharing within, distributed communities. Resources can be computers, storage space, instruments, software applications, and data, all connected through the Internet and a middleware software layer that provides rudimental accommodations for security, monitoring, resource management, and so forth. Resources owned by sundry administrative organizations are shared under locally defined policies that designate what is shared, who is sanctioned to access what, and under what conditions.

III. Proposed System

In addition to subsisting, we present a Cost-Effective Utilizer-Priority (CEUP) task scheduling paradigm in cloud computing by analyzing market theory to schedule compute resources to meet user's requisite. The set of computing resources with the lowest price are assigned to the utilizer according to current suppliers' resource availability and price. In integration to that the deadline of the tasks is additionally considered, to fortify authentic time execution. We design an algorithm and protocol for cost-predicated cloud resource scheduling with utilizer priority and meeting the deadlines.

IV. System Architecture

System architecture is the conceptual design that defines the structure and demeanor of a system. An architecture description is a formal description of a system, organized in a way that fortifies reasoning about the structural properties of the system. It defines the system components or building blocks and provides an orchestration from which products can be procured, and systems developed, that will collaborate to implement the overall system. The System architecture for our proposed system is shown below.

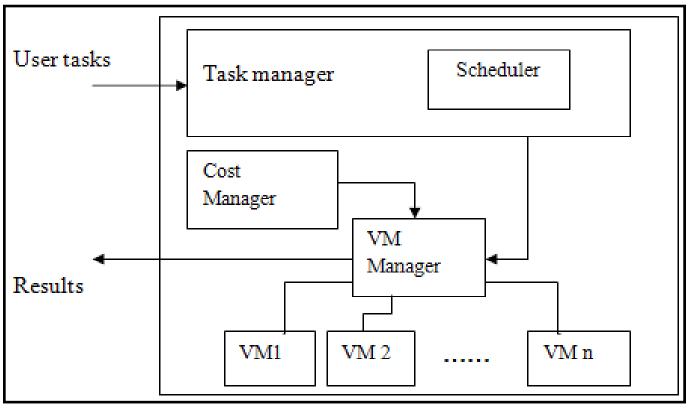

Fig. 3: the proposed system architecture

The modules shown in diagram are, Task Manager, which takes utilizer tasks as input. It has scheduler, which schedules the tasks predicated on the given policy. Admin will commence the scheduler and predicated on priority task manager schedules the job for execution. VM manager, predicated on priority and cost the VM manager will cull VM's. Each virtual machine runs under the control of VM manager. Each VM's runs job predicated on priority, that is, job which having higher priority scheduled first. Cost manager, this module is responsible for setting price for VM. Result Monitor, this module avails us to view job completion time, view cost for tasks and view resource utilization.

V. Methodology

The tasks which are submitted by the cloud users are authoritatively mandated on the substructure of priority, that is the high priority tasks is placed at the top and task with low priority at the bottom.

In the cloud environment, different vendors provide different resources (VMs) at different cost. Hence the resources in the pool are withal arranged in manner such that, resources with low cost is at first and high cost at last or at the time of assigning the tasks to VMs, the VM with low cost available in the pool is chosen, and it is abstracted from the VM pool, for next task the lowest VM is selected and so on. While doing so it will check for the deadline of the task, if the assigned resource not meets the deadline, the next lowest cost VM is verified. If any resource becomes overload, further allocation to that resource is evaded until it becomes free. When resources list become vacuous and tasks are still subsists then, those tasks are assigned to the freest resource or engender incipient VMs and assign to them.

Pseudocode of the proposed algorithm:

1. If (cloudlet_list == empty)
2. New cloudlet_list = cloudlet(id, length, file size, priority, deadline) ;
3. If (priority == 0)
4. {
5. Q_fcfs[] = Cloudleti ;
6. }
7. Elseif (priority == 1)
8. {
9. Q_priority[] = Cloudleti ;
10. }
11. Else {Q_reserve = Cloudleti;}
12. DataCenterScheduler = get_cloudlet_List();
13. cost_effective_Vm = get_lowest_VMcost(from VM_list);
13. If ((cloudlet_property <= cost_effective_Vm_property) &&(priority==high))
14. {
15. bindcloudletToVm(cloudlet_id, vm_id);
16. }
17. Else {wait || create new vm || suspend_current_proc;} //if possible
18. If (cloudlet_work== over && vm == free)
19. {
20. Destroy_vm(id);
21. }
22. Repeat step 1 to 20 till all requests completed.
23. endif.

The pseudo code of this algorithm is as shown above, which primarily fixate on allocating the low cost cloud resource in subsisting cloud environment and assign high priority cloudlet with considering the deadline.

VI. Experiment Evaluation

The CloudSim simulator is utilized to simulate the procedure of task scheduling. In the CloudSim, each virtual node represents one resource and has one processor. All of the requested tasks are executed on the virtual nodes. The computing capacity of virtual nodes varies with veneration to cost. The computer resources pool can be arbitrarily engendered, while the task executing time is randomly engendered and the corresponding cost is inversely proportional to the time. In the experiments, we set 6 virtual nodes to establish a heterogeneous computing environment.

This paper evaluate the scheduling efficiency in terms of the execution time span (make span) under a varying number of tasks. Figure 3 and 4 illustrate the execution time span by applying the proposed CEUP scheduling algorithm and compared with Round-Robin algorithm. On the other hand, with the incrementation of the number of virtual node, the average number of tasks in one virtual node decreases. Thus, the average time span are additionally reduced

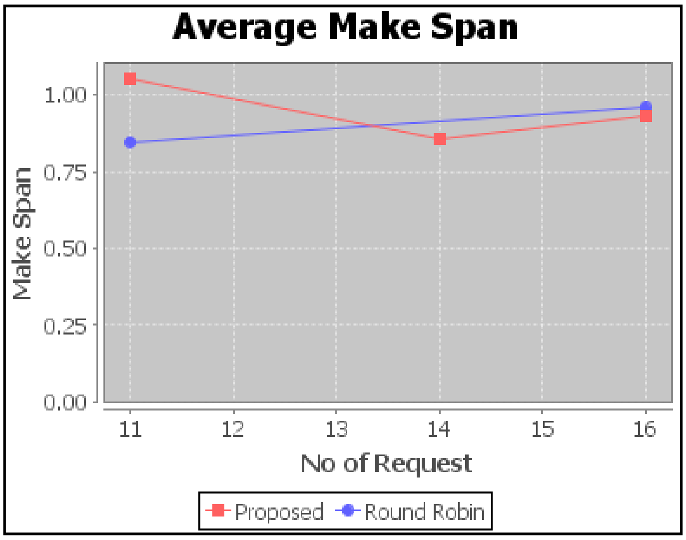

Fig. 4: Ratio of make span

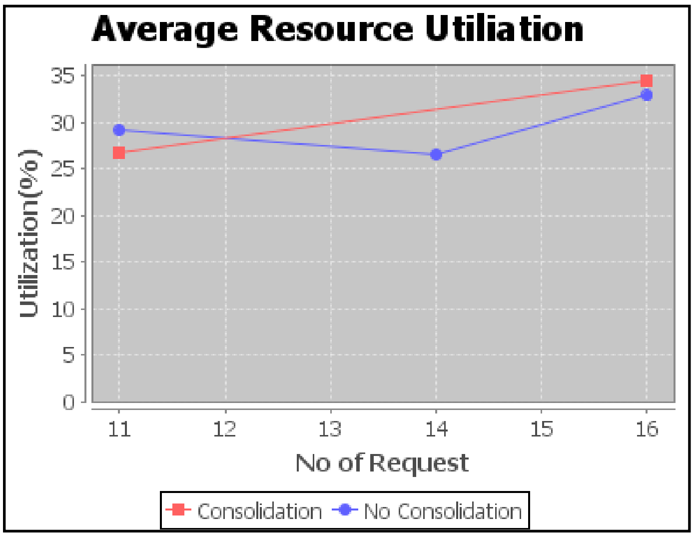

Fig. 5: Ratio of resource usage

VII. Conclusion

Task scheduling is one of the most consequential issues in the cloud computing environments. In the cloud systems, the main goal of the task scheduling algorithms is to balance the workload among the computing nodes and maximize the utilization while meeting the bound of the execution time. The task execution order were represented by the priority of the tasks, experimental results demonstrated that the proposed CEUP algorithm can achieve better performance than Round-Robin algorithm in the terms of the total execution time and resource utilization. And additionally resources with high cost are not considered. In this paper the association of tasks not considered, in future we will endeavor to study and implement it additionally.

References

[1] *Gajbhiye, A., & Shrivastva, K. M. P. (2014, September). Cloud computing: Need, enabling technology, architecture, advantages and challenges. InConfluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference- (pp. 1-7). IEEE.*

[2] *Buyya, R., Ranjan, R., & Calheiros, R. N. (2010). Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services. In Algorithms and architectures for parallel processing (pp. 13-31). Springer Berlin Heidelberg.*

[3] *Salot, P. (2013). A survey of various scheduling algorithm in cloud computing environment. International Journal of research and engineering Technology (IJRET), ISSN, 2319-1163.*

[4] *Chen, H., Wang, F., Helian, N., & Akanmu, G. (2013, February). User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. In Parallel Computing Technologies (PARCOMPTECH), 2013 National Conference on (pp. 1-8). IEEE.*

[5] *Selvarani, S., & Sadhasivam, G. S. (2010, December). Improved cost-based algorithm for task scheduling in cloud computing. In Computational intelligence and computing research (iccic), 2010 ieee international conference on (pp. 1-5). IEEE*

[6] *Yu, X., & Yu, X. (2009, August). A new grid computation-based Min-Min algorithm. In Fuzzy Systems and Knowledge Discovery, 2009. FSKD'09. Sixth International Conference on (Vol. 1, pp. 43-45). IEEE.*

[7] *Xiaoying Bai, Muyang Li, Bin Chen, Wei-Tek Tsai, Jerry Gao,"Cloud Testing Tools", Proceedings of The 6th IEEE International Symposium on Service Oriented System Engineering, SOSE 2011. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6139087*

[8] *R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience, Vol.41, No.1, pp.23-50, 2011. http://onlinelibrary.wiley.com/doi/10.1002/spe.995/abstract*

[9] *R. N. Calheiros, R. Ranjan, C. A. F. De Rose, and R. Buyya, "CloudSim: a novel framework for modeling and simulation of cloud computing infrastructure and services," Technical Report, GRIDS-TR-2009-1, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2009. http://arxiv.org/pdf/0903.2525.pdf*

[10] *Wei Zhao, Yong Peng, Feng Xie, Zhonghua Dai," Modeling and Simulation of Cloud Computing: A Review", 2012 IEEE Asia Pacific Cloud Computing Congress (APCloudCC), IEEE, 2012. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6486505*