International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-2 Number-4 Issue-6 December-2012

Improving Performance Guarantees in Cloud Computing through Efficient and Dynamic Task Scheduling

Shital Patil[1], R. A. Kulkarni[2]
Department of Computer Engineering, PICT, Pune, India
BE Computer[1], ME Computer[2]

Abstract

Cloud Computing has changed the way we think about technology. It is a computing model that provides web based software, middleware and other computing resources on demand by deploying technology as a service. Eucalyptus is one of the open source software computing model that provides Infrastructure as a Service cloud. To increase throughput and gain maximum profit in cloud systems task scheduling plays very important role. In this paper we present an efficient, deadline constrained dynamic task scheduling algorithm that schedules tasks by dynamically calculating the load on virtual machines. We consider Eucalyptus as our cloud computing platform.

Keywords

Scheduling, real-time systems and embedded systems, Distributed systems, cloud computing

1. Introduction

In cloud computing Infrastructure as a service acts as the basic layer of the cloud stack. Virtualization is the idea behind this stack. In cloud environment Infrastructure as a Service is a stipulation model in which an organization outsourcers the tools used to support different operations like storage, hardware, servers and networking components. These organizations are customers who are using these services on a pay per use basis. These tools are owned by service providers who are responsible for housing, running and maintaining it. Thus it includes utility computing, virtualization; SLA based services, internet connectivity and administrative tasks. In this paper we are using Eucalyptus that is an open source cloud computing system which is referred for infrastructure as a service. IaaS delivers computing resources on demand as a shared service thus avoiding maintenance and operating of hardware. Apart from providing cost savings users can also access latest software, platform and infrastructure offerings to cultivate business innovations also it should provide services at low cost and high utilization with scalability, reliability. Thus to achieve best performance with available resources is the main objective of service providers and customers. Efficient task scheduling is the way by which we can achieve the above said goals.

The resources in cloud computing are mostly heterogeneous in nature. Cloud providers always try to optimize the uses of resources. But consumers also have heterogeneous requirements in terms of tasks. The tasks may be of different types and with different priorities. The parameters like network bandwidth, task execution time, deadlines are associated with each task. All these parameters can change dynamically. So the dynamic nature of incoming tasks should be considered while scheduling. Also cloud computing provides services for highly data and compute intensive applications and managing large number of online transactions. For such type of applications it is necessary to meet deadlines of each task. All of these requirements can be fulfilled by an efficient task scheduling algorithm Our work in this paper focuses on deriving an optimized scheduling algorithm to improve efficiency and performance of a cloud system. We consider Eucalyptus as our cloud computing software model. Presentably Eucalyptus uses greedy, round robin and power save scheduling strategies. Greedy algorithm always chooses the first free available node. But it does not consider deadlines associated with tasks. The round robin algorithm selects the nodes until one free node is found. This strategy also does not consider the deadlines of tasks. Also most of the nodes have to wait unnecessarily for their turn. In power save algorithm nodes are put to sleep when they are not being used. Whenever required they are awaken and jobs are allocated to them. Jobs are allocated to those nodes which are awakened first. To overcome the issues related to these scheduling algorithms we propose dynamic scheduling strategy in this paper. The rest of the paper is organized as follows. Section two discusses the existing solutions to scheduling problems in cloud systems. In section three we propose our dynamic task scheduling algorithm.

Section four concludes the paper with Summary and future work.

2. Related Work

Task scheduling and resource allocation is very hot topics in distributed computing. Researchers have put forth different types of ideas to increase performance of cloud systems. In paper [1] authors have presented a novel utility accrual approach for on line scheduling of real time services. The algorithm uses profit TUF and penalty TUF to indicate profits for early completions and penalties of missed deadlines and abortions of respected tasks. The approach gives better results compared to EDF but have not considered different types of Virtual machines. Authors of [2] have used the AHP i.e. Analytic hierarchic process to improve the performance by improving consistency ratio by using comparison matrix. But in this work no dynamic resources allocation is considered. Load balancing and resource scheduling are closely related problems to task scheduling. Genetic algorithms [3] are used to tackle these problems. Along with genetic algorithms historical data and current state of system are considered to choose least affective solution and to reduce dynamic migration. In cloud systems users charge for the services they use. In such cases mostly they have fixed budget constraints. Considering this issue the paper [4] proposed BaTS method i.e. budget constrained scheduler. The approach dynamically estimates the completion time while considering an upper bound on budget. For huge datacenters in times of high load the consumer has to take decision to outsource the load to cloud providers. In such cases consumer tries to maximize output and save cost with quality of service requirement. To tackle such problems authors of [**5**] have formulated a binary integer program of scheduling problem. They found out that this method gives good results for public cloud than hybrid clouds. Real time scheduling strategies of operating systems can also be used to improve performance and QoS guarantees of cloud systems. The paper [**6**] presented this approach to increase virtual machine throughput and activation latency with response time. The other issues in cloud computing like load balancing and resource allocation can also be solved for some extent by using scheduling algorithms. Thus the paper [**7]** presents a cloud model and different algorithms to solve the problems of load balancing and resource allocation. Making of scheduling decisions becomes more difficult when there are too many QoS requirements. The solution to this type of decision making is given in paper [8]. In [8] paper authors have implemented ACO i.e. Ant Colony Optimization approach to schedule the large incoming workflow and to fulfil all the QoS constraints of users.

Eucalyptus is used to serve IaaS in cloud computing model, and it is open source, different solutions are presented by authors to improve its efficiency. Thus the authors of paper [9**]** suggested Monte Carlo, Round Robin and Weighted Queuing algorithms to support distributed storage support in private clouds. But these algorithms use static scheduling strategies. In real time we have to follow so many deadline constraints so that dynamic scheduling strategies should be used to maintain the deadline constraints of tasks. Thus considering these issues in the next section we present the dynamic scheduling algorithm for improving throughput and performance guarantees.

3. Proposed Approach

The problem is to schedule the requests to resources which are in terms of tasks, on the available nodes in a cloud system. Each task has a predefined deadline. The tasks should be scheduled in such a way that they should meet their deadlines. The scheduler selects the task having high priority in terms of deadlines. To meet deadline constraints we have used the dynamic scheduling strategy. Whenever a request for resources arrives the scheduler dynamically calculates the load on each node. The load of a physical node can be calculated as the total load of all the virtual machines running on it. The load of a virtual node can be calculated as the total load of tasks running on that node. Also the system calculates the total execution time or total completion time of tasks on each node. Based on these calculated parameters scheduler chooses the node having minimum load and minimum execution time to schedule the tasks. Also the scheduler matches the capacity of chosen node with tasks requirements. If the required and available capacities are matching then only that node is allocated for task execution. But if the desired not is not found then scheduler searches for the next suitable node. For this work we are considering that the tasks are non pre-emptive in nature. The system can be represented as follows.

Let S be the proposed system.
We define S as

S= {T, N, D, E, L, C};

All these terms are defined as follows.

Inputs to the system are T, D as set of Tasks, set of deadlines associated with each tasks and N is the set of nodes and C is the set of capacities associated with the nodes.

T is the set of Tasks stored in a queue as $T= \{t_1, t_2, t_3, t_4, t_5, \ldots t_n\};$

Let D is the set that denotes the deadlines associated with each tasks and denoted as D_i.

$D= \{d_1^t, d_2^t, d_3^t, d_4^t, \ldots d_n^t\};$

Let N be the set of nodes in the systems.

$N= \{n_1, n_2, n_3, n_4, n_5, \ldots n_m\};$

Let C be the set that denotes the capacities associated with nodes and denoted as C_i^t

$C= \{c_1^n, c_2^n, c_3^n, c_4^n, \ldots c_m^n\};$

The problem statement requires dynamic calculation of load and total execution time for each node. Let v_i be the load associated with a virtual node.

The load of a physical node can be calculated as total load of the virtual machines running on it.

$L_{pi}=v_1+v_2+v_3+\ldots\ldots+v_n$ i.e. $L_{pi} = \sum_{i=1}^{n} v_i$

Similarly the load of a virtual node can be calculated as the total load of tasks running on it.

$Lv_i= l_1^t + l_2^t + l_3^t + l_4^t \ldots\ldots\ldots + l_n^t$; i.e.

$$L_{vi} = \sum_{i=1}^{n} lt_i$$

The total execution/completion time of a node can be calculated as follows:

Let E be the total execution/completion time of a particular node. It can be calculated as

$E=t_1^e+ t_2^e + \ldots + t_n^e$; i.e.

$$E = \sum_{i=1}^{n} t_i$$

Where t_i^e is the execution time associated with task t_i.

All these terms are calculated dynamically. We will apply this algorithm with eucalyptus to achieve performance improvements. The mapping of tasks and nodes is as follows. Therefore the system function will be

$F: T \longrightarrow N$

Where $N \in \min (L_i, E_i)$,

$i \in (1,2,3,4,5,\ldots\ldots n)$;

This scheduler can be represented diagrammatically as follows

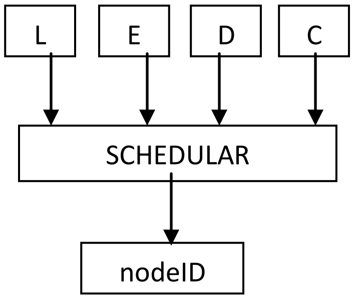

Figure 1: Block Diagram of Scheduler

Scheduler takes input as L, E, D and C i.e. load, execution time, deadline and capacity. Accordingly it takes the decision to schedule the task on right nodes. NodeID is the desired node found for request allocation.

Based on this mathematical model we present our scheduling algorithm as follows.

Algorithm for dynamic scheduling of tasks in a cloud system

```
Inputs: T, D, N;
Output: nodeID;
Begin
resID=0, nodeID=0, done=0;
//Calculate load l and completion time e of each node in the node queue
l=0, e=0;
for each node in the available node queue and not done
calculate load of each node;
calculate completion time of each node;
end
//Select the high priority task for execution
For each task in task queue
Do
If the task has high priority than other tasks in the queue
Then
Select the task for execution
End
//find the best node on which to run the task
For each task in the queue and the task has highest priority
do
If (a node has minimum load and node has minimum completion time)
//Check the capacity of that node
If (capacity of the node >=capacity required by the task)
then
Select the node to assign the task
```

```
nodeID=resID;
done++;
end if
end if
else if (node not found)
return nodeID=-1;
end if
end
```

Following flowchart is derived from the above algorithm. The flowchart gives detailed information of all functions and flow of control among them. The algorithm performs two tasks simultaneously i.e. marinating task queue and selecting node. Then it checks capacity of selected node and capacity required by task. If they are not matching then scheduler selects next matching node for task execution. All these tasks are done by eucalyptus components collectively. They are cluster controller, node controller, instance manager and group manager The proposed algorithm for scheduling will assign high priority task to proper nodes thus it will avoid resource underutilization which is caused because of traditional scheduling algorithms of eucalyptus..

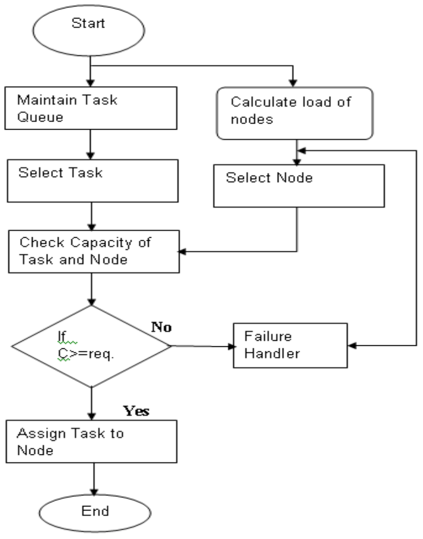

Figure 2: Flowchart showing functional flow of algorithm

4. Conclusion and future work

Cloud computing provides resources on pay per use, on demand and service level basis through internet. Both service providers and users are looking for increased performance guarantees with deadline constraints for real time systems. These guarantees can be provided by task scheduling algorithms. In view of this we have formalized a model of dynamic task scheduler in this paper. The algorithm calculates required scheduling parameters dynamically along with taking consideration of deadlines of the tasks. Future work will include experimental setup and execution of above proposed algorithm and comparison of results of traditional scheduling algorithms of eucalyptus with the proposed dynamic algorithm.

References

[1] S. Liu et al. On-Line Scheduling of Real-Time Services for Cloud Computing. In World Congress on Services, pages 459–464, 2010

[2] Daji Ergu, Gang Kou, Yi Peng, Yong Shi, Yu Shi, “The analytic hierarchy process: task Scheduling and resource allocation in cloud computing environment”, Springer Science+Business Media, LLC 2011

[3] GU, J., HU, J., ZHAO, T., SUN, G. A New Resource Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment. Journal of Computers, North America, 7, jan. 2012.

[4] Oprescu, A., & Kielmann, T. (2010, November). “Bag-of-tasks scheduling under budget constraints.” In Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on (pp. 351-359). IEEE.

[5] Van den Bossche, R., Vanmechelen, K., & Broeckhove, J. (2010, July). Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads. In Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on (pp. 228-235). IEEE.

[6] Cucinotta, T., Giani, D., Faggioli, D., & Checconi, F. (2011). Providing performance guarantees to virtual machines using real-time scheduling. In Euro-Par 2010 Parallel Processing Workshops (pp. 657-664). Springer Berlin/Heidelberg.

[7] Maguluri, S. T., Srikant, R., & Ying, L. (2012, March). Stochastic models of load balancing and scheduling in cloud computing clusters. In INFOCOM, 2012 Proceedings IEEE (pp. 702-710). IEEE.

[8] W. Chen, J. Zhang, “An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem With Various QoS Requirements”, IEEE Transactions on Systems,Man, and Cybernetics - PartC:Applications and Reviews, Vol. 39, No. 1, January 2009.

[9] Król, D., & Kitowski, J. (2011, September). Distributed Storage Support in Private Clouds Based on Static Scheduling Algorithms. In CLOUD COMPUTING 2011, The Second

International Conference on Cloud Computing, GRIDs, and Virtualization (pp. 141-146).
[10] Eucalyptus Cloud Platform http://www.eucalyptus.com
[11] Ubuntu Operating System http://www.ubuntu.com.

Shital Patil has received BE degree in Computer Engineering from Shivaji University, in 2004. She is currently appearing for Master of Computer Engineering in Pune Institute of Computer Technology, Pune. Her research area is Operating System, Cloud Computing.