2012 Third International Conference on Emerging Applications of Information Technology (EAIT)

Optimal Scheduling of Computational Task in Cloud using Virtual Machine Tree

Raghavendra Achar*, P. Santhi Thilagam†, Shwetha D*, Pooja H*, Roshni* and Andrea*
*Department of Computer Science and Engineering
St. Joseph Engineering College, Mangalore 575028, INDIA
Email:{raghunitk, shwetha131990, kamath.pooja, roshni296, andreafeliciar}@gmail.com
†Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal 575025, INDIA
Email: santhisocrates@gmail.com

Abstract—**The increasing demand in computing resources and widespread adaptation of Service Oriented Architecture (SOA) has made cloud as a new IT delivery mechanism. In cloud, computing resources are provided to the requester as a service, which include Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Cloud Computing is still in developing stage and faces many challenges. Out of the various issues, scheduling plays a very important role in determining the efficient execution of tasks in cloud environment. In this paper we present a scheduling algorithm which uses tree based data structure called Virtual Machine Tree (VMT) for efficient execution of tasks. The proposed algorithm is tested using CloudSim simulator and the results shows that algorithm gives better performance compared to other traditional scheduling algorithms.**

Index Terms—**Virtual Machine, Scheduling, Cloud Computing.**

I. Introduction

Cloud computing is an internet based computing, which provides dynamically scalable and virtualized resources as a service to the requester using pay per use model. In cloud, resources are provided to the requester as a service, which include Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In SaaS, software application is made available by the cloud provider. In PaaS, an application development platform is provided as a service to the developer to create a web based application. In IaaS, computing infrastructure is provided as a service to the requester in the form of Virtual Machine (VM). VM is a software implemented machine which runs on physical machine. Each Virtual Machine created behaves like a physical machine, running different operating systems and applications. Cloud Computing is still in developing stage and faces many challenges. Out of the various issues regarding a cloud, scheduling plays a very important role in determining the efficient execution of tasks in cloud environment. Scheduling refers to the appropriate assignment of tasks to the resources available like CPU, memory and storage, such that there is a maximum utilization of resources. Efficient scheduling is a necessary for both cloud service requesters as well as providers.

The rest of the paper is organized as follows. Section 2 reviews the related work on task scheduling in cloud environment. In section 3 we define our problem statement. Section 4 gives the solution methodology. In section 5 we describe implementation and experiment details. Finally section 6 presents conclusion.

II. Related Work

In this section, we describe the related work of task scheduling in cloud environment. The authors of paper [1] presented an optimized algorithm for task scheduling based on genetic simulated annealing algorithm. This considers the QoS requirements like completion time, bandwidth, cost, distance, reliability of different type tasks. Here annealing is implemented after the selection, crossover and mutation, to improve local searchability of genetic algorithm. In paper [3] authors introduce an utility accrual scheduling algorithm for real-time cloud computing services. This approach uses two time utility function (TUF) namely profit TUF and a penalty TUF. In paper [4] authors present an optimized algorithm for task scheduling based on Activity Based Costing (ABC). This algorithm assigns priority level for each task and uses cost drivers. ABC measures both cost of the object and performance of the activities. An improved cost based scheduling algorithm is presented in paper [5] for making efficient mapping of tasks to available resources in cloud. The algorithm measures both resource cost and computation performance. Algorithm improves computation/communication ratio by grouping the user tasks according to particular cloud resources processing capability. In paper [2] authors introduce a Multiple QoS Constrained Scheduling Strategy to schedule multiple workflows. The proposed system consists of three core components: Preprocessor, Scheduler and Executor. The paper [6] presents transaction intensive cost constraint cloud workflow scheduling algorithm. Algorithm consider execution cost and execution time as the two key considerations. The algorithm minimize the cost under certain user designated deadlines. Our proposed methodology is mainly based on computational capability of Virtual Machines.

III. Problem Statement

In the world of Cloud Computing, scheduling of the requesters task is an interesting issue which is open for research. The success of this rising model is dependent on the effectiveness of techniques used to execute the requesters task in the most optimal way. Thus to achieve this, the following scenario has been taken up as the problem statement.

Let $C = \{d_1, d_2, ..., d_n\}$, where C is a cloud and $d_1, d_2, .., d_n$ are the datacenters. Let $d_i = \{s_1, s_2, ..., s_m\}$ where each datacenter d_i comprises of servers $s_1, s_2, ..., s_m$. Let $s_j = \{v_{j1}, v_{j2}, ..., v_{jl}\}$ where $v_{j1}, v_{j2}, ..., v_{jl}$ are the Virtual Machines in the server s_j. Let $v_{ji} = \{v_{id}, v_{mips}\}$ where v_{id} is the Virtual Machine Id and v_{mips} is the MIPS of the Virtual Machine. Let W = $\{j_1, j_2, j_3, ..., j_p\}$ where $j_1, j_2, j_3, ..., j_p$ are set of task to be executed in the cloud. Let $j_i = \{j_{id}, j_{size}\}$ where j_{id} is the Id of the task and j_{size} is the size of the task. The problem is to optimally schedule the set of task 'W' in the cloud 'C' such that each resources in C is optimally utilized and the overall execution time of the workload 'W' is minimal.

IV. Methodology

Cloud consists of numerous Virtual Machines with varying degree of computational capabilities. The submission order of the tasks and the Virtual Machines in which these tasks are executed greatly influences the execution time of the entire workload. For an optimal scheduling strategy, the tasks and Virtual Machines binding must be wisely chosen. In the proposed novel scheduling mechanism, first we prioritise the tasks and Virtual Machines. We create a tree based data structure called Virtual Machine Tree (VMT) in which each nodes of a tree represents a Virtual Machine. Then grouping of task is done based on number of leaves in the VMT. The modified DFS algorithm will identify the suitable Virtual Machines, for which the submitted tasks will be executed. The details about the prioritizing, grouping and construction of VMT is given below.

A. Prioritizing

In the proposed strategy, the tasks are initially prioritized according to their size such that one having highest size has highest rank. The Virtual Machines are also ranked (prioritized) according to their MIPS value such that the one having highest MIPS has the highest rank. Thus, the key factor for prioritizing tasks is their size and for VM is their MIPS.

B. Virtual Machine Tree (VMT)

A Virtual Machine Tree (VMT) is a binary tree with N nodes. Each node represents a Virtual Machine containing Virtual Machine Id and MIPS. N represents total number of computational specific Virtual Machines in a cloud. The special property of VMT is that node value (MIPS) at level L is greater than or equal to node value at level L+1 where $L \geq 0$. Each node contains zero, one or two child nodes. A node with no child node is called as a leaf node and the node with child nodes is referred as internal nodes. Consider a 5 computational specific Virtual Machines represented by their Id and MIPS as $V = \{\{0, 250\}, \{1, 1000\}, \{2, 250\}, \{3, 500\}, \{4, 250\}\}$. Figure below shows the VMT. The VMT is constructed based on the prioritized order of Virtual Machines from left to right, such that Virtual Machine with highest MIPS becomes the root.

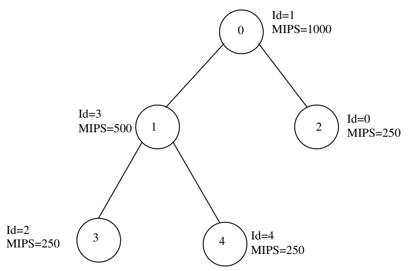

Fig. 1. Virtual Machines as a nodes of a VMT

Here VMT with the root node representing the Virtual Machine with Id 1 and MIPS 1000. The root node has two children. The left child node represents the Virtual Machine with Id 3 and MIPS 500. The right child node represents the Virtual Machine with Id 0 and MIPS 250. Similarly node which represents the Virtual Machine with Id 3 and MIPS 500 has 2 children. The left child of this node represents the Virtual Machines with Id 2 and MIPS 250, right child represents the Virtual Machine with Id 4 and MIPS 250 respectively.

C. Grouping

Here we present a grouping mechanism for the set of task submitted to the cloud. Let T_COUNT be the total number of tasks submitted and L_COUNT be the total number of leaf nodes in VMT. The total number of groups G_COUNT for the submitted tasks are computed as follows.

$$G\_COUNT = L\_COUNT$$

If VMT constructed with 5 Virtual Machines, then total number of groups G_COUNT is equal to 3. The number of tasks in each group G is computed as follows.

$$G = \frac{T\_COUNT}{G\_COUNT}$$

The first group G1 contains G_COUNT number of tasks from the prioritized tasks set. The second group G2 contains next G_COUNT number of tasks from the prioritized tasks set and so on. Consider 12 tasks represented by their Id and size as $G = \{\{0, 20000\}, \{1, 10000\}, \{2, 20000\}, \{3, 10000\}, \{4, 10000\}, \{5, 20000\}, \{6, 10000\}, \{7, 20000\}, \{8, 10000\}, \{9, 10000\}, \{10, 20000\}, \{11, 10000\}\}$

After prioritizing and grouping, each group contains following tasks.

G1=$\{\{0, 20000\}, \{2, 20000\}, \{5, 20000\}, \{7, 20000\}\}$
G2=$\{\{10, 20000\}, \{1, 10000\}, \{3, 10000\}, \{4, 10000\}\}$
G3=$\{\{6, 10000\}, \{8, 10000\}, \{9, 10000\}, \{11, 10000\}\}$

D. Virtual Machine Selection

Once the grouping of the tasks are done, suitable Virtual Machines are selected for execution from VMT. The tasks in each group is selected sequentially and submitted to the Virtual Machine. The order is as follows. The first task in the group G1 is executed by the Virtual Machine represented by the root node of the VMT. The second task will be executed by its child, third task will be executed by grand child and so on. Once it reaches the Virtual Machine represented by the leaf node, the next task will be submitted once again to root node and so on. Same procedure is repeated for all the tasks in each group. Figure below shows the VMT for 5 Virtual Machines and total number of groups formed for the 12 submitted tasks.

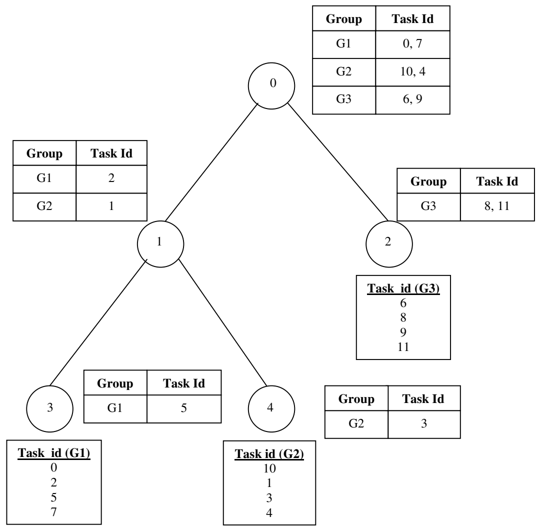

Fig. 2. Grouping and Submission of Tasks

Here total number of tasks submitted will be in 3 groups namely G1, G2 and G3 respectively. Tasks with Id 0, 2, 5, 7 will be in group G1, tasks with Id 10, 1, 3, 4 will be in group G2 and tasks with Id 6, 8, 9, 11 will be in group G3 respectively. The first task with Id 0 in group G1 will be executed by the Virtual Machine represented by root node of the VMT, the second task with Id 2 in group G1 will be executed by the Virtual Machine represented by node 1 which is the first child of the root node. The 3rd task with Id 5 in group G1 will be executed by the Virtual Machine represented by the node 3. Since node 3 is the leaf node, the fourth task with Id 7 in group G1 will be executed once again by the root node and process will be repeated for all tasks in group G1. Similarly first task with id 10 in group G2 will be executed by root node, the second task with Id 1 in G2 will be executed by the Virtual Machine represented by node 1 which is the child of the root node. The path is selected for Virtual Machine in such a way that it reaches the second leaf node. When second leaf node is reached, next task will be executed once again by root node and so on. The process will be repeated for all tasks in group 3 in such way that it takes the path from root node to the 3rd leaf node. We have modified DFS algorithm which is used to select the suitable Virtual Machines for executing the tasks as described previously. The algorithm is given below. This algorithm stores all suitable Virtual Machines in a VMList.

Algorithm 1 Modified DFS

```
$prev \leftarrow -99$
push first vertex
**while** Stack $\neq$ Empty **do**
  get unvisited vertex adjacent to stack top
  **if** no adjacent vertex **then**
    **if** prev $\neq$ StackTop **then**
      copy all stack contents to VMList
    **end if**
    pop
    **if** Stack $\neq$ Empty **then**
      prev = StackTop
    **end if**
  **else**
    mark the node as visited
    push adjacent vertex
  **end if**
**end while**
```

Once the Virtual Machines details are stored in VMList, the tasks are submitted to the Virtual Machines in specified order. When the task is submitted to the leaf node Virtual Machine, it must be checked that finish time of the task on the leaf Virtual Machine is not more than the finish time of the same task on the Virtual Machine which is represented as the root or its child. If it is more, instead of submitting the task to the leaf node Virtual Machine, task is submitted to the root node Virtual Machine, thus ensuring that load on each Virtual Machine is balanced.

V. Experiments and evaluation

In order to verify our algorithm, we conducted experiment on Pentium Dual Core Processor 2.6 GHz, Windows XP platform and using CloudSim 3 [7] simulator. The CloudSim toolkit supports modeling of cloud system components such as data centers, host, virtual machines, scheduling and resource provisioning policies. We have created 5 Virtual Machines using Vm component and set the property of RAM as 512 MB for all Virtual Machines, and the MIPS as 250, 1000, 250, 500 and 250 respectively. We have created 12 tasks using Cloudlet component and set the property of Cloudlet length as 20000, 10000, 20000, 10000, 10000, 20000, 10000, 20000, 10000, 0000, 20000 and 10000 respectively. The VMT along with execution time for this configuration is shown below.

In order to test our algorithm with huge number of tasks, we have taken the workload traces from Grid Workloads Archive [8]. Grid Workloads Archive provides workload traces from

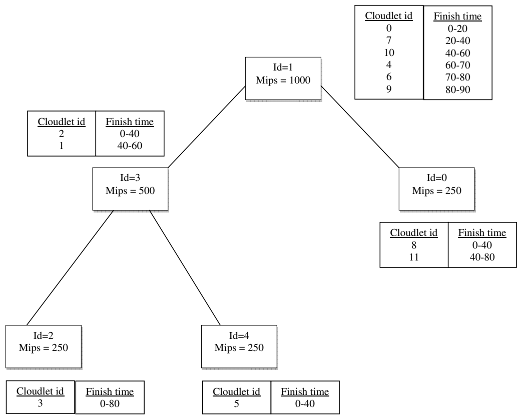

Fig. 3. Task Scheduling in Cloud using VMT

grid environments for the researchers to carry out experiments in simulated environment. Here we have used workload traces Grid5000 and AuverGrid. The number of experiments conducted by varying the number of Virtual Machines and for different workload traces. First we have verified our algorithm on workload traces Grid5000. For this we considered 5 Virtual Machines with MIPS 1000, 500, 250, 250, 250 and RAM size of all Virtual Machine as 512 MB. Experiment is conducted for varying number of tasks like 100, 200, 300, 400 and 500 respectively. For comparison and analysis, we implemented the FCFS, priority based algorithm. The execution time for the proposed algorithm is shown below.

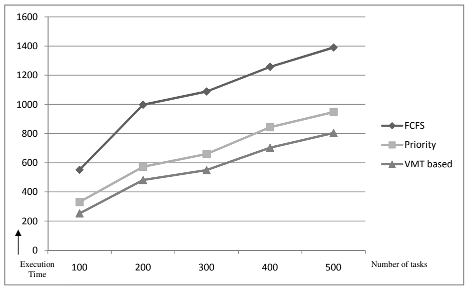

Fig. 4. Grid5000 workload for 5 Virtual Machines

Another set of experiment is conducted for workload traces from AuverGrid. Here instead of 5 Virtual Machines, we have created 10 Virtual Machines with MIPS 5000, 2000, 2000, 700, 250, 500, 1000, 1024, 250 and 250 respectively. RAM size of each Virtual Machine is 512 MB. Experiment is conducted for varying number of tasks like 100, 200, 300, 400 and 500 respectively. The execution time is shown below.

The results shows that the proposed algorithm is more efficient than FCFS and priority based algorithms. In this experiment, we consider execution time and resource utilization as the main metric. The execution time for each workload is lesser as compared to other algorithms. This is because the selection of Virtual Machine order in the proposed algorithms

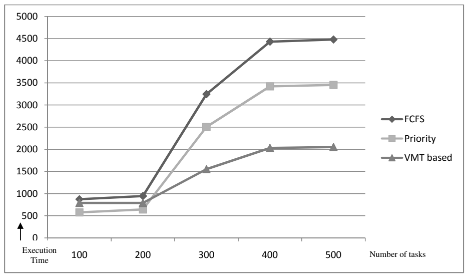

Fig. 5. AuverGrid workload for 10 Virtual Machines

leads to higher resource utilization. The usage of idle machines is carried out efficiently in the proposed method.

VI. Conclusion

In this paper we present a novel scheduling algorithm, which efficiently schedules the computational tasks in cloud environment. We have created tree based data structure called Virtual Machine Tree. We modified DFS which select the suitable Virtual Machines for execution. The experiment is conducted for varying number of Virtual Machines and workload traces. The experiment conducted is compared with FCFS and priority based algorithm. The results shows that the proposed algorithm is more efficient than FCFS and priority based algorithms.

Acknowledgment

We sincerely acknowledge Grid5000 and AuverGrid team for providing workload traces to conduct experiment on scheduling.

References

[1] G. Guo-Ning and H. Ting-Lei, "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment," In Proceedings of International Conference on Intelligent Computing and Integrated Systems, 2010, pp. 60-63

[2] M. Xu, L. Cui and H. Wang, "Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing," In Proceedings of IEEE International Symposium on Parallel and Distributed Processing with Applications, 2009, pp. 629-634

[3] S. Liu, G. Quan and S. Ren, "On-line Scheduling of Real-time Services for Cloud Computing," In Proceedings of 6th World Congress on Services (SERVICES-1), 2010, pp. 459-464

[4] Q. Cao, W. Gong and Z. Wei, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing," In Proceedings of Third International Conference on Bioinformatics and Biomedical Engineering, 2009, pp. 1-3

[5] S. Selvarani and G. S. Sadhasivam, "Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing," In Proceedings of IEEE International Conference on Computational Intelligence and Computing Research, 2010, pp. 1-5

[6] Y. Yang, Kelvin, J. chen, X. Lin, D.Yuan and H. Jin, "An Algorithm in SwinDeW-C for Scheduling Transaction Intensive Cost Constrained Cloud Workflow," In Proceedings of Fourth IEEE International Conference on eScience, 2008, pp. 374-375

[7] N. Rodrigo, R. Rajiv, B. Anton, A. Csar. and R. Buyya ,"CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,"Journal of Software: Practice and Experience Volume 41, Issue 1, pages 2350, January 2011

[8] The Grid Workloads Archieve, [online] Available:http: //gwa.ewi.tudelft.nl/ pmwiki, [visit:June 2012]