IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014),
August 01-02, 2014, Dr. Virendra Swarup Group of Institutions, Unnao, India

# An Energy Preserving and Fault Tolerant Task Scheduler in Cloud Computing

R.K. Yadav
Computer Engineering Department
Delhi Technological University
Delhi, India
rkyadav@dce.edu

Veena Kushwaha
Computer Engineering Department
Delhi Technological University
Delhi, India
Veena.kush2012@gmail.com

*Abstract*— **In Cloud, virtualization technology is employed to integrate physical machines into a virtual resource pool to control resources in centralized manner. Recent work considers various strategies with only taking into account only one specific problem of task scheduling without considering the other related problems. Provisioning customer applications in the Cloud while maintaining the application's required quality of service and achieving resource efficiency with power consumption issues are still open research challenges in Cloud computing. Hence, by considering other related problems of the task scheduling while schedule tasks can improve the resource utilization, fault tolerance, energy saving and throughput of the Cloud system.**

**In this paper, we proposed an integrated task scheduling algorithm that takes into account the issues such as VM management and Datacenter management. This algorithm is implemented on CloudSim platform and the obtained experimental results show that the proposed algorithm runs efficiently, reducing the average execution time, average waiting time of tasks and improving the throughput of the Cloud system. The proposed algorithm can be easily integrated with Virtual Machine management strategies and is fault tolerant and can efficiently deal with energy consumption problem.**

***Keywords-Task Scheduling; Cloud; Energy Efficient; CloudSim; Virtual Machine***

## I. INTRODUCTION

Cloud Computing is a new vision of computing as utility, in which resources are stored on servers and customers can use them as a service on provisional basis by using laptops, computers or mobile devices over Internet. In Cloud platforms, the computing power is leased in the form of virtual machine (VM) to users with the help of virtualization technology. Thus, task scheduling problem in cloud system is a two level problem. The scheduling algorithm aims to schedule applications on VMs based on the agreed Service Level Agreement (SLA) terms and deploy VMs on physical resources based on resource availability and power consumption.

VM migration is used for balancing host's load and for managing power wastage due to under-utilized host so that green house effect can be reduced. VM migration means; to transfer information related to the current VM, create and deploy same replica there. VM migration can also be used for improving fault tolerance capability of datacenter. Thus, VM management is closely related with resource management and task scheduling.

In this paper, we proposed a modified Breadth First Search (MBFS) algorithm to find optimal VM for each task. Not only it minimizes the waiting time and makespan of given tasks set but can also be easily integrated with other VM management strategies for saving energy and improving fault tolerance. This algorithm was simulated using the CloudSim [1] toolkit package version 2.1.

The rest of the paper is organized as follows. Section II presents the related work, Section III introduce the details of proposed MBFS algorithm. Section IV presents the simulation results and finally section V concludes this paper.

## II. RELATED WORK

In [2], authors proposed CloudSim work style that resembles the work style of real Cloud. The distributed resources are unified in the virtualized resource pool; the resources are allocated to application in the form of virtual machine. These VMs can be deployed to any of the host of datacenter transparently to the users. The VM can be created, destroyed, migrated dynamically according to the application requirement and system load or work condition. Author presented different component responsible for specific task. It is cleared that these components are inter-related to each other. Hence, while scheduling task to optimal VM, scheduling at these different related components should also be taken into consideration, so that the whole system works efficiently.

As we studied, very few papers proposed: integrated solution of the task scheduling and resource allocation problem of Cloud computing. [3], [4] and [5] presented integrated solution of task scheduling problem. In [4], the first scheduler is responsible to give specification of the required VM, according to resource demand of tasks. And the second scheduler find appropriate host for VM. In addition to handle dynamic demand of task, VM migration strategy has been used. This paper [5] has proposed a two-level scheduler; a meta scheduler based on QoS and a VM scheduler based on backfill strategy. In [3], scheduling heuristic proposed an integrated resource provisioning strategy in which, application are scheduled based on the agreed SLA terms and VMs are deployed based on resource availability. They considered VM deployment as the second level problem. However, deploying VM on an appropriate host has various issues. In recent energy crisis, CO2 emission and fault tolerance capability are of paramount importance. So they should be tackled properly to make datacenter work efficiently. Thus, VM management comprises of creation of VM, deployment of VM, VM migration, VM destruction and VM scaling. So, these related issues should be taken into consideration while scheduling tasks.

In this paper, the proposed modified Breadth First Search algorithm can be associated with any VM management policy. We gave a brief explanation of how our approach can be efficiently integrated with VM management strategy to solve problem related to energy consumption, improving fault tolerance and reliability.

The approach of [6] is similar to presented in this paper, in the sense that it recycled the already existing virtual machines in Cloud system. So the significant amount of time can be saved to serve more requests that are wasted in creating and destroying virtual machines. But differs that they only consider scheduling of task to VM and if no appropriate VM is available creating new VM. In our case we are also taking consider other related issues.

## III. PROPOSED MBFS ALGORITHM

### *A. Cloud Architecture*

Figure 1 shows the generalized partial diagram of Cloud system. Cloud architecture consists of several layers as shown in fig, application layer, virtualization layer and infrastructure layer. Cloud offers the benefits of virtualization layer. Hence task is not scheduled directly on actual resources in Cloud. Instead, Task scheduling problem in Cloud is a two level problem. At first level: the task is assigned to appropriate VM and at second level: the VM should be allocated to the appropriate Host. In addition VM migration, scaling and destruction are also handled at virtualization layer. A typical application specific metrics include response time, waiting time, processing time, length of a task and number of tasks in request or waiting queue. A typical infrastructure specific metrics include system load, resource utilization. But, when energy consumption, tolerance capability, reliability, temperature maintenance and CO2 emission are taken into account, the objective of task scheduling must be adjusted accordingly.

These different management strategies of VMs and task scheduling are inter-related to each others. So, when task is scheduled to VM; these relevant issues should be taken into account and this is one of the rationale factors behind our proposal.

### *B. First Level: MBFS*

Proposed scheduling approach is based on earliest finish time, high processing capabilities and compatibility with other inter-related issues. We used the binary tree based data structure called virtual machine tree (VMT) for efficient execution of task as used by Modified Depth First Search (MDFS) [7]. We

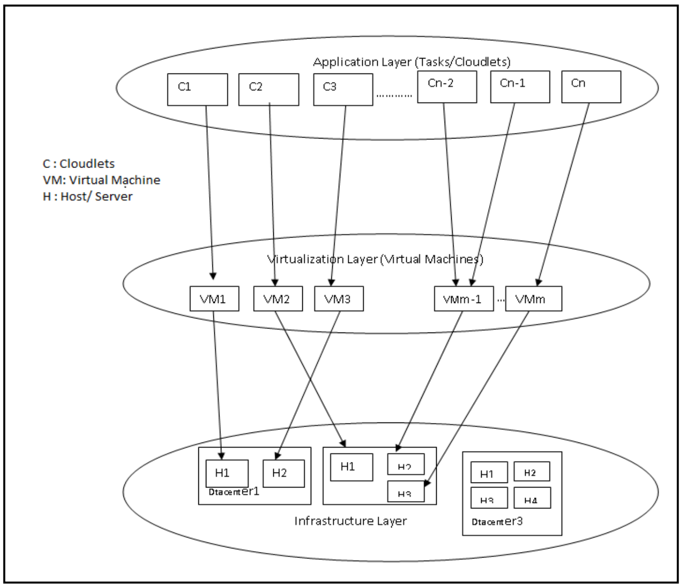


Figure 1. Partial diagram of Cloud System

prioritized task according to their MIPS. Then we used Modified BFS to identify the suitable Virtual Machine, for which the submitted task will be executed. In CloudSim task is known as cloudlet so, we are using the term cloudlet in algorithms. In algorithm 1 the tasks are prioritized based on their size.

**Algorithm 1: Prioritize tasks**

Input: cloudletList
Output: cloudletList is arranged in prioritized order

Step 1: Cloudlets are received by the scheduler
Step 2: Sort the cloudlets based on their size in decreasing order
Step 3: Store the cloudlets in a list, cloudletList

Algorithm 2 create a binary tree of VMs based on prioritized order of Virtual Machines from left to right such that the MIPS of VM at level L is greater than or equal to node value at level L+1 where $L \geq 0$ and their right sibling.

**Algorithm 2: VM Tree Construction**

Input: the list of VMs
Outputs: the root of VM Tree

Step 1: Obtain the all already created virtual machines
Step 2: Sort the virtual machines based on their MIPS
Step 3: Construct binary tree based on prioritized order of Virtual Machines from left to right
Step 4: Return the root of VM Tree

Algorithm 3 MBFS selects the appropriate Virtual Machine, at which the submitted task will be executed. In

MBFS, the tasks are crowded towards more powerful VMs. Since the capacity of parent node either equal or high than child node, so in step 4 of algorithm 3, the task are submitted to the parent node in case of tie; hence it ensures that the finishing time will be fast of the next tasks to be submitted to VM Tree. So the waiting time and the execution time are reduced. In our approach the VM having more power, is utilized more and the other VM can be freed, this will improve datacenter management efficiency.

**Algorithm 3: Modified BFS**

Input: VM Tree and cloudletList
Output: mapping among tasks and virtual machines

Step 1: insert root in queue
Step 2: for all the cloudlets available in cloudletList
Step 3: remove a node from queue and store it in current node
Step 4: check the estimated finishing time of cloudlet on current node as well as on parent of the current node
Step 5: submit the cloudlet to the node whose finishing time is less and in case of a tie submit task to parent node. And start process with next cloudlet.
Step 6: delete the front node of queue and insert its children nodes.

### *C. Second Level: Energy saving and Fault tolerance*

Recall that the main focus of this paper is to propose integrated approach of task scheduling problem of cloud computing.

*1) Energy Saving:* In 2010, data centers has consumed 0.5% of the world's total electricity usage and if the demand of energy continues, is projected to quadruple by 2020 [8]. And also, Earlier studies have showed that, the average resources utilization of data center is usually less than 20% .Even at a very low load, such as 10% CPU utilization, the power consumtion is over 50% of the peak power, because a large amount of energy is wasted by the idle resources [9]. Thus this issue needs to be immediately solved.

This problem can be solved by allocating host resources at max then go to next host for allocation of resources. In MBFS approach, the tasks are crowded towards more powerful VM. So these crowded VM can be deployed to same host while, others at other hosts. So, later on the other VMs eventually become free and the host can be switched off to save energy.

2)*Fault tolerance:* A substantial part of electrical energy consumed by computing hardware is transformed into heat. High temperature led to a number of problems, such as reduced the life time of resources as well as reduced system availability and reliability. Cooling system is used to keep the system hardware within their safe operating temperature and prevent crashes and failures. Although, the cooling system cost is higher than the installation cost for example, in a 30,000 ft2 data center with 1,000 standard computing racks, each consuming 10 kW, the initial cost of purchasing and installing the infrastructure is \$2-\$5 million; with an average cost of \$10/MWh, the annual costs of cooling alone are \$4-\$8 million [10].

One of the method to reduce the cooling operating costs, switched off the host when it becomes overheated and reallocate VMs. However, VM migrations are not free in terms of energy because every movement requires time and it is also important to take into account the total cost of such action [11]. The proposed method is suitable with this kind of remedy. In our approach we are using VM Tree to identify the suitable mapping among tasks and VMs. The VM Tree is constructed by the all VMs of datacenter. The hosts can be alternatively switched on and off at second level without affecting the task and VM mapping at first level.

We have to Create and deploy new VMs at the host that was switched off when a currently running host goes at high thermal state. When next time tasks need to be schedule construct VM Tree with these newly created VMs and discards the old VMs. And subsequently the host can be switched off

## IV. Performance Analysis

### *A. Simulator*

A simulator is good to evaluate a new approach, method or algorithm in controlled and repeatable manner. CloudSim provides a series of core classes and functions to establishment and simulation of cloud. The CloudSim toolkit support of modeling the cloud components such as data centers, host, virtual machine, cloudlets, scheduling and resource provisioning policies. This paper adopts CloudSim 2.1 to simulate the algorithm MBFS and MDFS and FCFS.

### *B. Assumptions*

We assumed that Tasks are mutually independent, i.e., there is no precedence relation between tasks, Tasks are non preemptive and Tasks are computationally intensive. The scheduling problem aims to minimize the waiting time, total execution time of tasks and as well as second level VM management to improve fault tolerance and reducing power consumption.

### *C. Experiment setup*

Numbers of experiments are conducted by varying number of tasks and virtual machines. First we have verified our algorithm with 5 Virtual Machines with MIPS 250, 500, 1000, 250, 250 and RAM size of all Virtual Machine as 512 MB. Another set of experiment is conducted with 10 Virtual Machines with MIPS 2000, 5000, 2000, 700, 250, 500, 1000, 1024, 250, 250 and RAM size of all Virtual Machine as 512 MB. To ensure simulated tasks are similar to real jobs in practice; we used Gaussian distribution function to create tasks and ensure their size larger than or equal to 1000 MI. The number of tasks varies 5 to 50 with 5 step size.

### *D. Simulation Results*

In this section, we will show comparative simulation results of proposed algorithm.

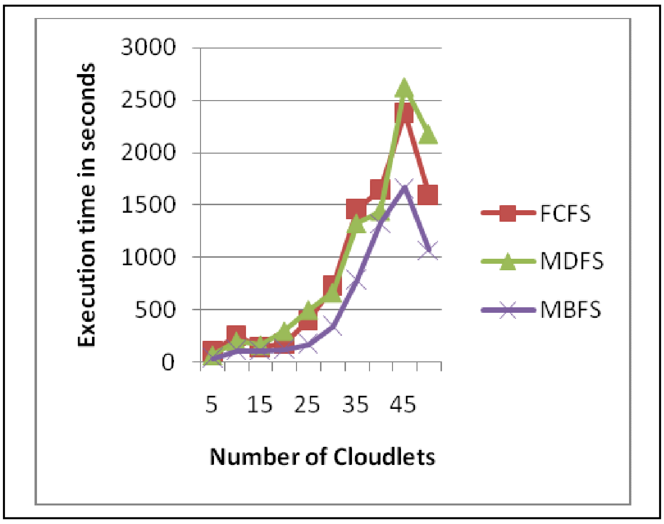


Figure 2. Average execution time with 5 virtual machines and different number of cloudlets

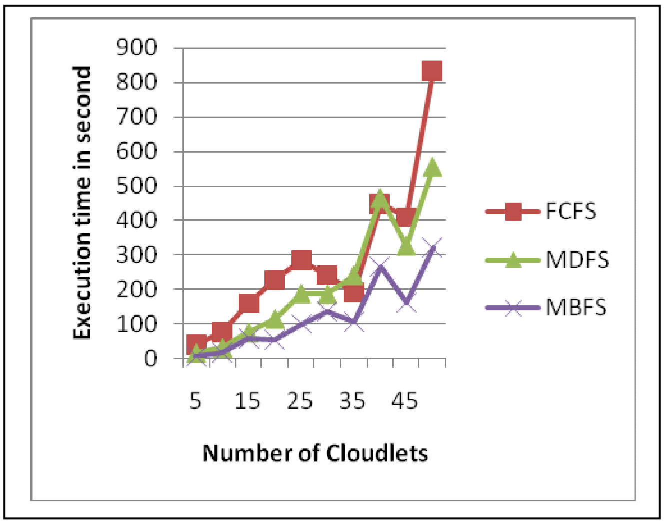


Figure 4. Average execution time with 10 virtual machines and different number of cloudlets

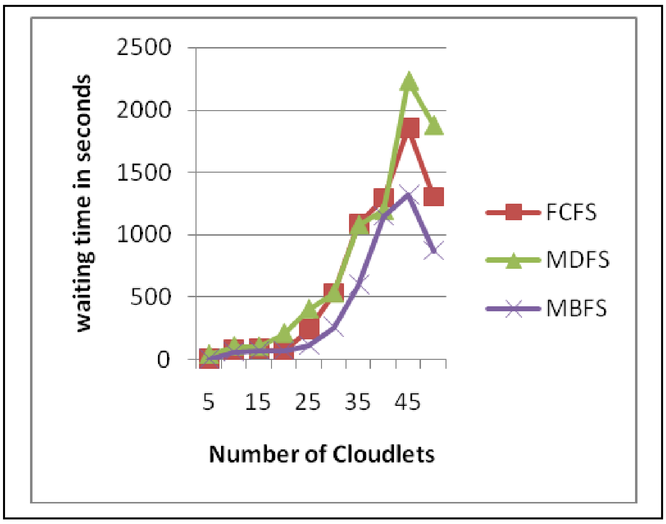


Figure 3. Average waiting time with 5 virtual machines and different number of cloudlets

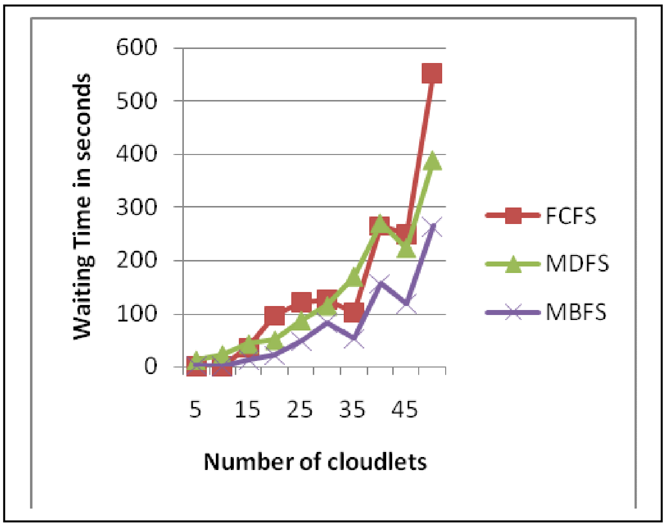


Figure 5. Average waiting time with 10 virtual machines and different number of cloudlets

The obtained results show that the proposed algorithm is more efficient than FCFS and MDFS algorithms. In this experiment, we consider execution time and waiting time as the main metric. Fig 2 and fig 4 compare the average execution time obtained by FCFS, MDFS and MBFS and fig 3 and fig 5 compare the average waiting time of FCFS, MDFS and MBFS. From figures, it can be seen that the execution time and waiting time taken to complete the tasks is less in MBFS as compared to MDFS and FCFS.

## V. Conclusion And Future Work

This paper discusses cloudlets scheduling algorithm based on modified breadth first algorithm in cloud environment. The rationale behind the algorithm is earliest finish time, lower average completion time as the tasks are crowded towards more powerful VM. The proposed approach considers inter-related issues such as energy efficiency and fault tolerance to manage resources of datacenter. It efficiently supports management of datacenter by allocating host resources at max for deploying virtual machines. So that, the idle host can be switched off. Then, Waiting time and Execution time of MBFS are compared with MDFS and FCFS. Simulation results show considerable reduction in average execution time and average waiting time.

As a future work, the availability vector that is used to prioritize the tasks could be extended to incorporate information about task requirements like deadline, budget. Virtual machines are prioritized based on their MIPS in this paper. In future the other characteristics such as memory and bandwidth can also be taken into consideration to prioritize the VMs.

## REFERENCES


[1] Rodrigo N. Calheiros1, Rajiv Ranjan2, Anton Beloglazov1, C´esar A. F. De Rose3 and Rajkumar Buyya1, "CloudSim: a toolkit for modeling and evaluation of Cloud computing environments and evaluation of resource provisioning algorithms", Published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com).

[2] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong, Dan Wang, "Cloud Task scheduling based on Load Balancing Ant Colony Optimization", IEEE 2011.

[3] Vincent C. Emeakaroha, Ivona Brandic, Michael Maurer and Ivan Breskovic, "SLA-Aware Application Deployment and Resource Allocation in Clouds", 2011 35th IEEE Annual Computer Software and Applications Conference Workshops

[4] Yiqiu Fang, Fei Wang, and Junwei Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", 2010 Springer-Verlag Berlin Heidelberg

[5] R. JEYARANI, N. NAGAVENI, and R. VASANTH RAM, " Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment", 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing

[6] Amit Kumar Das, Tamal Adhikary, Md. Abdur Razzaque and Choong Seon Hong, "An Intelligent Approach for Virtual Machine and QoS Provisioning in Cloud Computing", 2013 IEEE

[7] Raghavendra Achar, P. Santhi Thilagam, Shwetha D, Pooja H, Roshni_ and Andrea, "Optimal Scheduling of Computational Task in Cloud using Virtual Machine Tree", 2012 Third International Conference on Emerging Applications of Information Technology (EAIT).

[8] W. Forrest. How to cut data centre carbon emissions[EB/OL]. http://www.computerweekly.com/Articles/2008/12/05/233748/How-to-cut-data-centre-carbon-emissions.htm. December 2008.

[9] Jingling Yuan, Xing Jiang and Luo Zhong_Hui Yu, "Energy Aware Resource Scheduling Algorithm for Data Center using Reinforcement Learning", 2012 Fifth International Conference on Intelligent Computation Technology and Automation

[10] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmal, and R. Friedrich, "Smart cooling of data centers," in Proceedings of the Pacific RIM/ASME International Electronics Packaging Technical Conference and Exhibition (InterPACK), 2003, pp. 129–137

[11] Tom Gu´erout and Mahdi Ben Alaya, "Autonomic energy-aware tasks scheduling", 2013 Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises


978-1-4799-6393-5/14/$31.00 ©2014 IEEE