ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

Resource Optimization Based Scheduling Algorithm in Cloud Environment

Priyanka Dhurvey[1], Nagendra Kumar[2]

[1]M.Tech. Scholar, Dept. of CSE, Shri Ram Institute of Science & Technology, Jabalpur, M.P., India

[2]Asst. Professor, Dept. of CSE Shri Ram Institute of Science & Technology, Jabalpur, M.P., India

ABSTRACT: In the current scenario, Cloud computing carved itself as an emerging technology which enables the organization to utilize hardware, software and applications without any upfront cost over the internet. The challenge before the cloud service provider is, how efficiently and effectively the underlying computing resources like virtual machines, network, storage units, and bandwidth etc. should be managed so that no computing device is in under-utilization or over-utilization state in a dynamic environment. A good task scheduling technique is always required for the dynamic allocation of the task to avoid such a situation. Through this thesis we are going to present the different task scheduling technique, which will distribute the load effectively among the virtual machine so that the overall response time (QoS) should be minimal. This thesis explores methods of scheduling in cloud computing. It helps to understand the wide task scheduling options in order to select one for a given environment. We have analysed various techniques for better utilization of resources and also compared those methods and proposed a new method of scheduling that helps in reducing the makespan. Finally, concluded with some suggestion as future work.

KEYWORDS: Cloud Computing, Makespan, Scheduling algorithms, Task Scheduling.

I. INTRODUCTION

Cloud computing is a new technology currently being studied in the academic world. The definition of the cloud computing from the Gartner: “A style of computing where massively scalable IT-related capabilities are provided as a service across the internet to multiple external customers using internet technologies [1].” The cloud computing platform guarantees subscribers that it sticks to the service level agreement (SLA) by providing resources as service and by needs. However, day by day subscribers’ needs are increasing for computing resources and their needs have dynamic heterogeneity and platform irrelevance. But in the cloud computing environment, resources are shared and if they are not properly distributed then it will result into resource wastage. Another essential role of cloud computing platform is to dynamically balance the load amongst different virtual machines in order to avoid hotspots and improve resource utilization. Therefore, the main problems to be solved are how to meet the needs of the subscribers and how to dynamically as well as efficiently manage the resources.

Scheduling in Cloud Computing: Task scheduling is the process of allocating the resources to the particular job in specific time. The main objective of scheduling is to maximize the resource utilization. Minimizing the waiting time is the goal of scheduling. A good scheduling algorithm yields good system performance. In the cloud there are numerous and distinct resources available. The cost of performing tasks in cloud depends on which resources are being used so the scheduling in a cloud environment is different from the traditional scheduling. In a cloud computing environment task scheduling is a biggest and challenging issue. Task scheduling problem is NP-Complete problem.
Many heuristic scheduling algorithms have been proposed, but more improvement is needed to make system faster and more responsive. The traditional scheduling algorithms like First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR), Min-Min, and Max-Min algorithms is not much better solution to scheduling problems with cloud computing [2]. So we need the better solution to this heuristic problem.

Scheduling in Cloud Computing: Task scheduling is the process of allocating the resources to the particular job in specific time. The main objective of scheduling is to maximize the resource utilization. Minimizing the waiting time is the goal of scheduling. A good scheduling algorithm yields good system performance. In the cloud there are numerous and distinct resources available. The cost of performing tasks in cloud depends on which resources are being used so

the scheduling in a cloud environment is different from the traditional scheduling. In a cloud computing environment task scheduling is a biggest and challenging issue. Task scheduling problem is NP-Complete problem.
Many heuristic scheduling algorithms have been proposed, but more improvement is needed to make system faster and more responsive. The traditional scheduling algorithms like First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR), Min-Min, and Max-Min algorithms is not much better solution to scheduling problems with cloud computing [2]. So we need the better solution to this heuristic problem.

II. RELATED WORK

Task Scheduling Approaches: There are different scheduling algorithms available for cloud computing. In this paper we will discuss about five scheduling algorithms they are First Come First Serve (FCFS), Round Robin (RR), Genetic algorithm, Match-making algorithm and generalized priority algorithm.

A. First come first serve (FCFS) Algorithm: FCFS is mainly used for parallel processing, it is selected for incoming task and it is aimed at resource with smallest waiting queue time. The CloudSim toolkit supports FCFS scheduling strategy for internal scheduling of jobs. Allocation of application-specific VMs to host in a cloud based data center is the responsibility of the virtual machine provisioned component. The default policy of implemented by VM provisioning is straightforward policy that allocates a VM to the host in FCFS basis. The disadvantage of FCFS is it is non- preemptive algorithm. The shortest tasks which are at the back of the queue having to wait for the long task at the front finish. Its turn around and response time is low.

B. Round Robin (RR) Algorithm: The Round Robin algorithm focuses on fairness and on distributing the load equally to all nodes. Each job in a queue has same execution time and it will be executed in turn. The scheduler starts assigning VM to each node and move further for next VM to place in next node. Algorithm is applied for all the nodes until one VM is assigned to each node. Again it goes to the first node repeat this process to the next VM request.
The advantage is that it utilizes all the resources in the balanced order. Disadvantage is high power consumption as many nodes are turned on. If four resources have to be run on a single node, all the nodes will be turned on when Round Robin is used. This consumes high power.
Supports RR CloudSim toolkit supports RR scheduling strategy for internal scheduling of jobs. CloudSim toolkit supports RR scheduling strategy for internal scheduling of jobs.

***C.* Genetic algorithm:** Genetic algorithms are stochastic search algorithms based on the mechanism of natural selection strategy. It starts with a set of initial solution, called initial population, and will generate new solution using genetic operators. The genetic algorithm approach computes the impact in advance that it will have on the system after the new VM resource is deployed in the system, by utilizing historical data and current state of the system. It then picks up the solution, which will have the least effect on the system. The advantage of this technique is it can handle a large searching space, applicable to complex objective function and can avoid trapping by local optimum solution. Authors of [3] have developed a cost-based job scheduling algorithm, which provide a multi QoS scheduling in cloud computing environment.

D. Match-Making Algorithm: The algorithm first filter out the nodes or hosts those do not meet the VM requirements and do not have enough resources (Like CPU, Memory, Processors etc) to place and run the VM. Rank will be given to nodes as per the gathered information by the monitoring drivers. If any variable comes in monitoring then it will be included in to rank expression. The result of rank expression is given to the cloud scheduler and monitoring driver makes decision for VMs placement and reconfiguration. Open Nebula has default match making scheduler that implement the Rank scheduling policy. Open Nebula comes with Haizea Scheduler that support advance reservation of resources and queuing of best effort requests. The goal of this algorithm is to prioritize resources those are most suitable for the VM. Those resources with a higher rank are used first to allocate VMs.

***E. Improved Cost Based Algorithm* [4]:** This algorithm ameliorates the traditional cost-based scheduling algorithm for making appropriate resource allocation. The tasks are grouped as per processing power of resources.
***F. Earliest Feasible Deadline First* [5]:** The principle of this algorithm is very simple to understand. In this scheduling algorithm, the task having the shortest deadline is getting scheduled. Earliest Deadline First (EDF) is a dynamic scheduling. Whenever a scheduling event occurs (end of a task, the release of the new task.) then the queue will be

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

searched for the process that is closest to its deadline, the found process will be the next that is going to be scheduled for execution.

***G. A Priority based Job Scheduling Algorithm* [6]:** In this algorithm, the approach is presented for job scheduling by using mathematical calculations. In this algorithm for scheduling the priority is considered and each job request for resources with some priority. This paper discusses issues related to the algorithm such as consistency; complexity and makespan i.e. finish time. As per author makespan can be reduced further by improving the algorithm.

***H. A Priority based Scheduling Strategy for VM Allocation* [7]:** The objective of proposed algorithm is to gain more benefits to the service providers since the current resources are not adequate to process all the requests. This technique can increase the benefits than applying typical FCFS strategy. If more information can be made available, e.g. the regular pattern of the usage the algorithm can be improved further.

***I. Queue Based Job Scheduling Algorithm (QHS)* [8]:** The algorithm is mentioned as QHS algorithm. The algorithm focuses on increasing the efficiency of execution of jobs. The results of various scheduling algorithms such as FCFS, Round Robin, and SJF are compared with proposed algorithm. The performance metrics on which the results are compared are average waiting time and average response time. This algorithm succeeded to optimize the average waiting time, and average response time.

***J. Generalized Priority Based Algorithm (GPA)* [9]:** The GPA mainly focuses to reduce the execution time of tasks. In this algorithm tasks and Virtual Machines (VM) are prioritized according to million instructions per second (MIPS). VM can execute and tasks are prioritized according to length or size of tasks. The VM having highest MIPS value and the task having largest size has highest priority. Tasks having highest priority are scheduled on the VM having highest priority. The results of the algorithm are compared with basic FCFS and Round Robin (RR) algorithms, where the performance of GPA is better than FCFS and RR.

***K. Greedy Based Job Scheduling Algorithm* [10]:** This algorithm focuses on QOS, as the cloud computing is a business oriented service. The goal of the algorithm is to reduce completion time and to give faster solution to scheduling problem. This algorithm classifies tasks based on QOS and then as per the task category, the appropriate function is assigned. The results of the algorithm are compared with other existing algorithm that are, the algorithm based on Berger model and existing scheduling strategy of CloudSim tool.

***L. Improved Priority based Job Scheduling Algorithm using Iterative Method* [11]:** In job scheduling priority is the largest matter since some jobs need to be scheduled first then remaining jobs can be scheduled which can wait. In this paper Improved priority based job scheduling algorithm is proposed in a cloud computing environment. This algorithm has better makespan and consistency than a prioritized round robin algorithm. Analytical Hierarchy Process is used. The proposed algorithm can be optimized further to minimize makespan.

***M. Priority Based Earliest Deadline First Scheduling* Algorithm [12]:** In this method there are two task scheduling algorithms are used, one is Earliest Deadline First and other is priority based scheduling algorithm. This algorithm focuses on resource allocation and memory utilization. The given approach will reduce the execution time of preempted tasks and they can schedule efficiently. This algorithm overcomes the waiting time problem of preempted tasks. The waiting queue is introduced which processes the preempted tasks.

N. Credit Based Scheduling Algorithm [13]: Cloud providers have limited amount of resources, and are thus obliged to strive to maximize utilization. In this paper scheduling algorithm introduced which are based on user priority and task length. Credits are assigned to task length and task priority. And by calculating the credit value the jobs are scheduled to be executed. The proposed algorithm works more efficiently than previous methods. The proposed scheme can be enhanced so as to consider other parameter like a deadline.

***O. Heuristic-PSO based Scheduling Algorithm (H-PSO)[14]* :** This algorithm is based on Particle Swarm Optimization (PSO) technique. This is a static procedure for scheduling. It reduces makespan and improves memory utilization. When this algorithm is compared to other PSO based algorithm, it reduces more makespan. The Heuristic-PSO method increases processing speed and gives an optimal solution.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

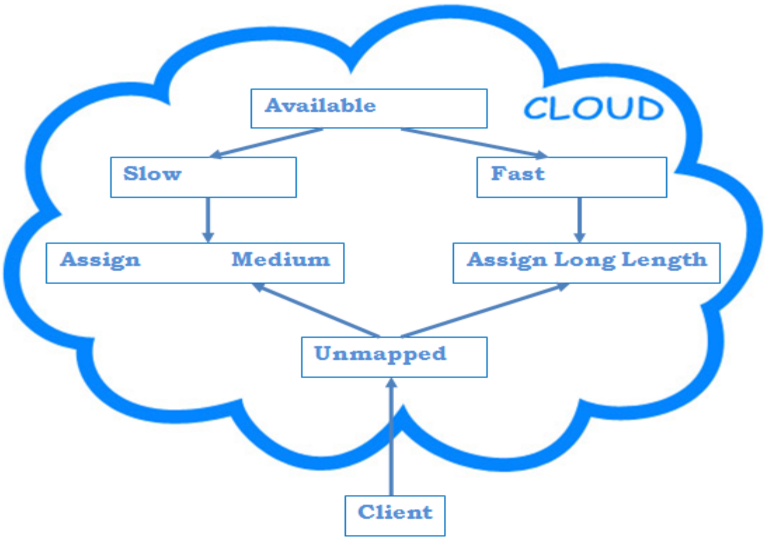
Algorithm	**Environment**	**Scheduling Factor**	**Objective Criteria**	**Advantages**	**Disadvantages**
First Come First Serve	Cloud/Grid Computing	Time	Energy Efficiency	Fair and easy to implement	Non-preemptive algorithm
Shortest Job First Scheduling Algorithm	Cloud Computing	Time	Execution Time	CPU is allocated to the process with least CPU burst time	High waiting time
Round Robin	Cloud Computing	Time	Response time	Utilization of all resources in balanced order, ensures fairness	High power consumption
Genetic Algorithm	Cloud Computing	Cost	Makespan	Handle large search space, Applicable to complex function.	Migration Cost is high
Match-Making Algorithm	Cloud Computing	Energy Consumption	Rank Scheduling Policy	Good distribution of workload into resources	Lack of availability and reliability
Generalized Priority Algorithm	Cloud Computing	Cost	Resource Allocation	Better than FCFS and RR	High power consumption.
Min Min Algorithm	Cloud Computing	Cost and Time	Makespan	Promised the guarantee regarded the provided resources	Large task may have to wait for execution until all tasks gets Finished.
Max Min Algorithm	Cloud Computing	Cost and Time	Makespan	Executes long tasks first, so that it may effect on total response time.	Small task may have to wait for execution until all task gets finished

TABLE 1: COMPARISONS OF VARIOUS EXISTING ALGORITHMS.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

III. PROPOSED SYSTEM ARCHITECTURE



Proposed system divides all available resources into two sets where first set is of the slow resources and second set is of the fast resources. This partitioning is done according to the MIPS speed of each resource. Suppose that there are 10 available resources. The first step is to sort these resources into ascending order by considering their MIPS speed and then add first 5 resources into the first set and remaining 5 resources into the second set. Figure shows the overall view of our solution. When tasks arrived for execution, choose the resource which takes less time for execution among all available resources before assigning any task. If chosen resource is from the first set, then assign average length task to it and if chosen resource is from the second set, then assign the longest task to it. The technique used in proposed algorithm for scheduling tasks is the combination of both Max-Min as well as Enhanced Max-Min. The proposed algorithm minimizes the chances of scheduling a large task to the slow resource with making completion time shorter. Proposed algorithm helps to utilize resources more efficiently and achieve good performance in terms of makespan as compared to existing algorithms.

A.PROPOSED ALGORITHM

Following represents proposed algorithm:
Input: T is task set, R is Resource set with MIPS speed, and N is total no of Resources
Algorithm Proposed (T, R, N)
{
    For ti in T do
    {
        Sort resource set R in ascending order of execution speed in MIPS.
    }
    R_{slow} = R_i form 1 to N/2;
    R_{fast} = R_i form N/2 to N;
    While t_i Ɛ T and T is not empty do
    {
        Find R_i with minimum execution time
        If R_i Ɛ R_{slow} than
            Assign average length task to R_i
        Else
            Assign maximum length task to R_i
        Delete t_i from T
    }
}

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

IV. SIMULATION RESULTS

Table below shows the comparisons of average execution time of three algorithms based on three workflows. Results show that the performance of proposed algorithm is better than FCFS and Max-Min.

ALGORITHM / WORKFLOW	FCFS ALGORITHM(MS)	MAX-MIN ALGORITHM(MS)	PROPOSED ALGORITHM(MS)
MONTAGE_50	41.78	14.94	12.88
CYBERSHAKE_50	129.86	101.23	41.59
INSPIRAL_50	1160.84	1132.51	725.96

Table 5.2: Makespan Comparison of Scheduling Algorithms for VM=10.

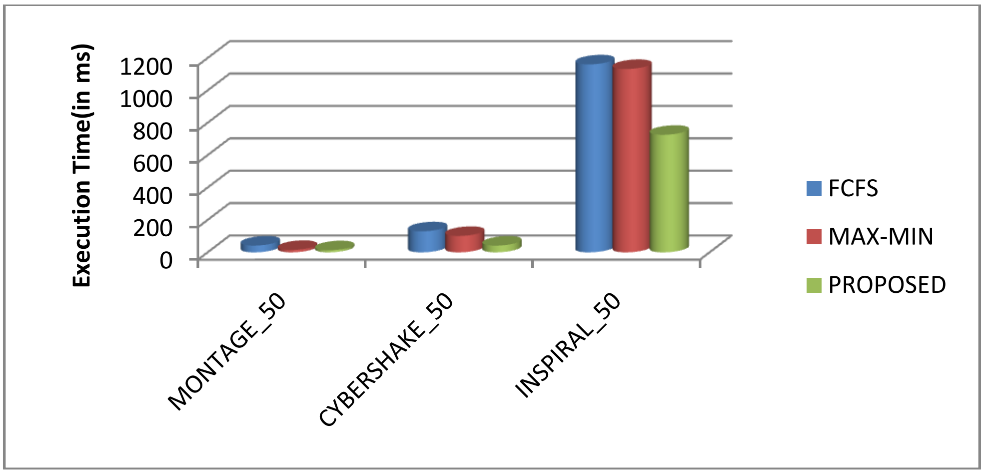


Fig.5.5: Makespan Comparison of Scheduling Algorithms for VM=10.

Performance of proposed algorithm is better than MaxMin and FCFS for workflow Montage50.

Table 5.3: Makespan Comparison of Scheduling Algorithms for VM=20.

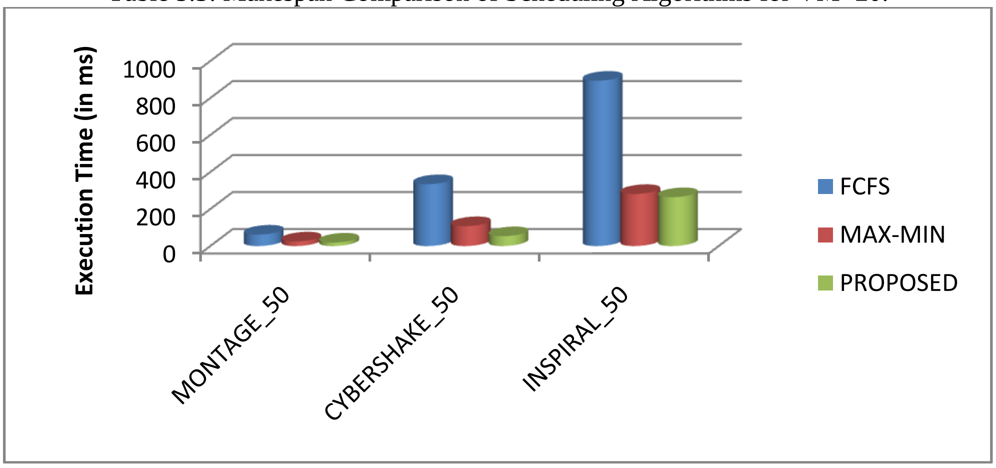


Fig.5.6: Makespan Comparison of Scheduling Algorithms for VM=20.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 5, May 2017

V. CONCLUSION AND FUTURE WORK

The proposed work presents new task scheduling algorithm based on resource partitioning to minimize the total length of scheduling and proper resource utilization. This algorithm is implemented using WorkFlowSim simulation tool. Results shows that proposed algorithm done efficient resource utilization and has better makespan than existing scheduling algorithms MaxMin and FCFS. Overall result shows that the proposed algorithm performs better than MaxMin algorithm when it comes to makespan. Proposed algorithm improves the performance of scheduling by 10 percentages as compared to MaxMin. Makespan of proposed algorithm for all workflows such as Montage50, Montage100, CyberShake30, CyberShake50, and Inspiral50 is less than makespan of MaxMin.
In future resource optimization can be performed using extension of proposed algorithm by combining various algorithms to provide hybrid algorithm. Task scheduling algorithm can also be extended for categorizing by type of task.

REFERENCES

[1]. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia, “Above the clouds: A berkeley view of cloud computing,” UCB/EECS-2009-28.
[2]. Kavyasri M N, Dr. B. Ramesh, “Comparative Study of Scheduling Algorithms to Enhance the Performance of virtual machines in Cloud Computing”, 978-1-4673-6725-7/16/$31.00 ©2016 IEEE.
[3]. Mayank Mishra, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo, “Dynamic Resource Management Using Virtual Machine Migrations”, Sep 2012, 0163-6804/12, IEEE Communications Magazine, page no: 34-40.
[4]. Selvarani, S., and G. Sudha Sadhasivam. "Improved Cost-Based Algorithm For Task Scheduling In Cloud Computing." Computational Intelligence And Computing Research (ICCIC), 2010 IEEE International Conference on. IEEE, 2010.
[5]. Jagbeer Singh, Bichitrananda Patra, Satyendra Prasad Singh, "An Algorithm to Reduce the Time Complexity of Earliest Deadline First Scheduling Algorithm in Real-Time System" (IJACSA) International Journal of Advanced Computer Science and Applications, February 2011.
[6]. Ghanbari, Shamsollah, and Mohamed Othman. "A priority based job scheduling algorithm in cloud computing." Procedia Engineering 50 (2012): 778-785.
[7]. Xiao, Jing, and Zhiyuan Wang. "A Priority Based Scheduling Strategy for Virtual Machine Allocations in Cloud Computing Environment." Cloud and Service Computing (CSC), 2012 International Conference on. IEEE, 2012.
[8]. Behzad, Shahram, Reza Fotohi, and Mehdi Effatparvar. "Queue based Job Scheduling algorithm for Cloud computing." International Research Journal of Applied and Basic Sciences ISSN (2013): 3785-3790.
[9]. Agarwal, Dr, and Saloni Jain. "Efficient optimal algorithm of task scheduling in cloud computing environment." arXiv preprint arXiv: 1404.2076 (2014).
[10]. Li, Ji, Longhua Feng, and Shenglong Fang. "An greedy-based job scheduling algorithm in cloud computing." Journal of Software 9.4 (2014): 921-925.
[11]. Patel, Swati J., and Upendra R. Bhoi. "Improved Priority Based Job Scheduling Algorithm in Cloud Computing Using Iterative Method." Advances in Computing and Communications (ICACC), 2014 Fourth International Conference on. IEEE, 2014.
[12]. Gupta, Gaurav, et al. "A simulation of priority based earliest deadline first scheduling for cloud computing system." Networks & Soft Computing (ICNSC), 2014 First International Conference on. IEEE, 2014.
[13]. Thomas, Antony, G. Krishnalal, and VP Jagathy Raj. "Credit Based Scheduling Algorithm in Cloud Computing Environment." Procedia Computer Science 46 (2015): 913-920.
[14]. TAREGHİAN, Shahab, and Zarintaj BORNAEE. "A new approach for scheduling jobs in cloud computing environment." Cumhuriyet Science Journal 36.3 (2015): 2499-2506.

BIOGRAPHY

Priyanka Dhurvey is a Research Assistant in the Computer Science & Engineering Department, Shri ram Institute of Science and Technology, Jabalpur, M.P.