International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue-01
January 2018

An Improved Cpu Scheduling Approach For Cloud Computing Environment

Jyoti Dahiya & Pooja

[1]Department of Computer Science International Institute of technology & business (IITB) Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat

Jyoti.pm14@gmail.com

[2]Department of Computer Science International Institute of technology & business (IITB), Deenbandhu Chhotu Ram University of Science & Technology (DCRUST), Sonepat

poojachoppra@gmail.com

Abstract— *Cloud computing appears as a new model and its main objective is to provide secure, quick, convenient data storage and net computing facility. An essential requirement in cloud computing environment is scheduling the current jobs to be executed with the given constraints. The scheduler should order the jobs in a way where balance between improving the quality of services and at the same time maintaining the efficiency and fairness among the jobs. In this dissertation we also proposed a new hybrid approach for CPU scheduling in cloud computing environment. The hybrid approach uses Minimum Completion Time of various jobs with Opportunistic load balancing approach on cloud servers. Then we compare the proposed method with the existing approaches in terms of one special metric – throughput.*

Keywords— **Resource Allocation, Resource Scheduling, Virtual Machines**

I. INTRODUCTION

Cloud computing [1] is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing storage, memory, processing and bandwidth. Cloud computing is a comprehensive solution that delivers Information Technology as a service.

Cloud computing is capable to provide massive computing or storage resources without the need to invest money or face the trouble to build or maintain such huge resources. Consumers only need to pay for using the services just like they do in case of other day to day utility services such as water, gas, electricity, etc. Cloud computing is now being used in many applications that are beyond distribution and sharing of resources [2]. The distributed resources are useful only if the cloud resources are scheduled. The optimal scheduler results in high performance cloud computing whereas poor schedulers produce substandard results. Scheduling algorithms are used for dispatching user tasks or jobs to a particular resource or data. Scheduling is a challenging job in the cloud because the capability and availability of resources vary dynamically. The goal of job scheduling is to properly dispatch parallel jobs to slave node machines according to scheduling policy meeting certain performance indexes and priority constraints to shorten total execution time and lower computing cost and improve system efficiency [3].

In spite of the various scheduling algorithms proposed for cloud environment, there is no comprehensive performance study undertaken which provides a unified platform for comparing such algorithms. These algorithms are Round Rubin (RR), Random Resource Selection, Opportunistic Load Balancing and Minimum Completion Time [8][9].

In this paper we also proposed a new hybrid approach for CPU scheduling in cloud computing environment. The hybrid approach uses Minimum Completion Time of various jobs with Opportunistic load balancing approach on cloud servers. Then we compare the proposed method with the existing approaches in terms of one special metric – throughput.

II. JOB SCHEDULING IN CLOUD ENVIRONMENT

Job Scheduling is used to allocate certain jobs to particular resources in particular time. In cloud computing, job-scheduling problem is a biggest and challenging issue. Hence the job scheduler should be dynamic. Job scheduling in cloud computing is mainly focuses to improve the efficient utilization of resource that is bandwidth, memory and reduction in completion time. An efficient job scheduling strategy must aim to yield less response time so that the execution of submitted jobs takes place within a possible minimum time and there will be an occurrence of in-time where resources are reallocated. Because of this, less rejection of jobs takes place and more number of jobs can be submitted to the cloud by the clients which ultimately show increasing results in accelerating the business performance of the cloud.

There are different types of scheduling based on different criteria, such as static vs. Dynamic, centralized vs. Distributed, offline vs. Online etc are defined below [5][6]:

1) Static Scheduling: Pre-Schedule jobs, all information are known about available resources and tasks and a task is assigned once to a resource, so it's easier to adapt based on scheduler's perspective.

2) Dynamic Scheduling: Jobs are dynamically available for scheduling over time by the scheduler. It is more flexible than static scheduling, to be able of determining run time in advance. It is more critical to include load balance as a main factor to obtain stable, accurate and efficient scheduler algorithm.

3) Centralized Scheduling: As mentioned in dynamic scheduling, it's a responsibility of centralized / distributed scheduler to make global decision. The main benefits of centralized scheduling are ease of implementation; efficiency and more control and monitoring on resources.

On the other hand; such scheduler lacks scalability, fault tolerance and efficient performance. Because of this disadvantage it's not recommended for large-scale grids.

4) Distributed / Decentralized Scheduling: This type of scheduling is more realistic for real cloud despite of its weak efficiency compared to centralize scheduling. There is no central control entity, so local schedulers' requests to manage and maintain state of jobs' queue.

5) Pre-Emptive Scheduling: This type of scheduling allows each job to be interrupted during execution and a job can be migrated to another resource leaving its originally allocated resource, available for other jobs. If constraints such as priority are considered, this type of scheduling is more helpful.

6) Non Pre-Emptive Scheduling: It is a scheduling process, in which resources are not being allowed to be re-allocated until the running and scheduled job finished its execution.

7) Co-operative scheduling: In this type of scheduling, system have already many schedulers, each one is responsible for performing certain activity in scheduling process towards common system wide range based on the cooperation of procedures, given rules and current system users.

8) Immediate / Online Mode: In this type of scheduling, scheduler schedules any recently arriving job as soon as it arrives with no waiting for next time interval on available resources at that moment.

9) Batch / Offline Mode: The scheduler stores arriving jobs as group of problems to be solved over successive time intervals, so that it is better to map a job for suitable resources depending on its characteristics.

The figure 1 below shows the general job scheduling process of Cloud computing environment.

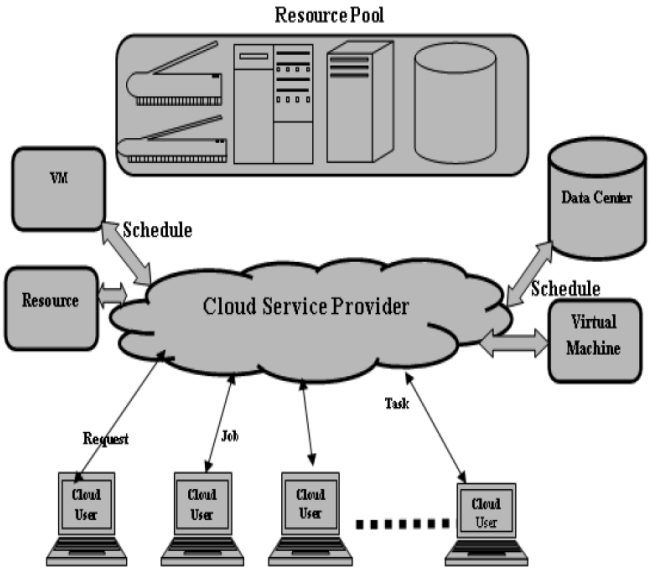

Figure 1: Job Scheduling Process in Clod Computing

III. SCHEDULING CRITERIA

Different CPU scheduling algorithms have different properties and the choice of a particular algorithm may favour one class of processes over another. In choosing which algorithm to use in a particular situation, we must consider the properties of the various algorithms. Many criteria have been suggested for comparing CPU scheduling algorithms [7].

The criteria include the following:

1. CPU Utilization: We want to keep CPU busy as possible.

2. Throughput: If the CPU is busy executing processes then work is being done. One measure of work is the number of processes that are completed per time unit, called throughput. For long processes, this rate may be one process per hour, for short transactions it may be 10 processes per second.

3. Turnaround time: The interval from the time of submission of a process to the time of completion is the turnaround time. Turnaround time is the sum of the periods spent waiting to get into memory, waiting in the ready queue executing on the CPU and doing I/O.

4. Burst Time: Amount of time required for the process for its execution.

5. Completion Time: The time when process completes its execution.

6. Waiting time: The CPU scheduling algorithm does not affect the amount of the time during which a process executes or does I/O, it affects only the amount of time

that a process spends waiting in the ready queue. Waiting time is the sum of period spends waiting in the ready queue.

7. Response time: The time from the submission of a request until the first response is produced. This measure called response time is the time it takes to start responding, not the time it takes to output the response. The turnaround time is generally limited by the speed of the output device.

IV. PROPOSED MODEL

A. Proposed Solution

CPU scheduling has significant contribution in efficient utilization of computer resources and increases the system performance by switching the CPU among the various processes. However, it also introduces some problems such as starvation, large average waiting time, turnaround time and its practical implementation. Many CPU scheduling algorithms are given to resolve these problems but they are lacked in some ways. Most of the given algorithms tried to resolve one problem but lead to others. To remove these problems, we introduce an approach that uses hybrid approach for CPU scheduling in cloud computing environment. The hybrid approach uses Minimum Completion Time of various jobs with Opportunistic load balancing approach on cloud servers. Then we compare the proposed method with the existing approaches in terms of three metrics - throughput, maximum finishing time and the total execution cost. From various experiments we show that our approach works better than existing methods in terms of above metrics.

B. Proposed Algorithm

The steps for performing CPU scheduling on cloud environment using hybrid approach of Minimum Completion Time of various jobs with Optimistic load balancing algorithm are listed below:

1. The initial stage for the proposed algorithm is to establish cloud data centers with multiple CPUs with different processing speed, Primary memory (RAM) and secondary storage (Disk Drives).

2. After registering with cloud data center, client can access the resources of that particular cloud data center.

3. Now client can send its jobs for processing on the cloud server. The client sends multiple parameters regarding each job requests such as processing time, capacity of primary and secondary memory.

4. The server processes the requests using minimum completion time scheduling of each job. For this processes are scheduled on the server that satisfying the client's requirement.

5. After that opportunistic load balancing algorithm is applied by the cloud data center to balance the load on various servers.

6. Steps 4 & 5 applied for each allocation and deallocation of servers to multiple clients requests until all the clients are logged out.

Hence the work provides efficient load balancing when required and also the minimum completion time helps to complete the users' tasks on the server that give minimum waiting time and maximum response time.

V. IMPLEMENTATION RESULT

The initial phase of our implementation is to establish cloud data centers with multiple CPUs with different processing speed, Primary memory (RAM). Now client can send its jobs for processing on the cloud server. The client sends number of jobs to be processed and processing time for each job.

The server processes the requests using minimum completion time scheduling of each job. For this processes are scheduled on the server that satisfying the client's requirement. After that opportunistic load balancing algorithm is applied by the cloud data center to balance the load on various servers.

Hence the work provides efficient load balancing when required and also the minimum completion time helps to complete the users' tasks on the server that give minimum waiting time and maximum response time.

Figure 2 shows the client login screen of our implementation.

Type client id and no. of jobs to be processed & press the start button.

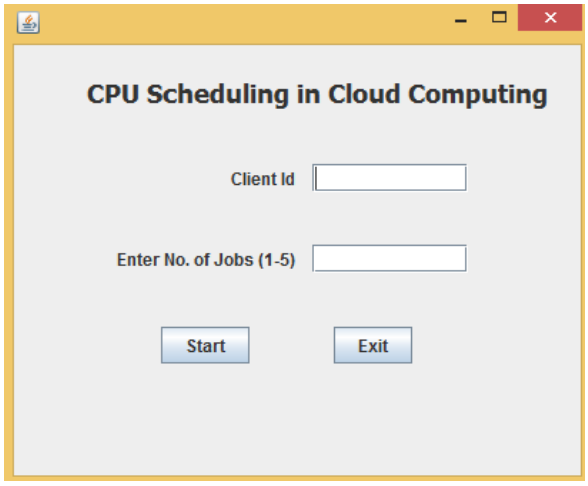

Figure 2: The client login screen of our implementation.

Figure 3 shows the processing time required for each job.

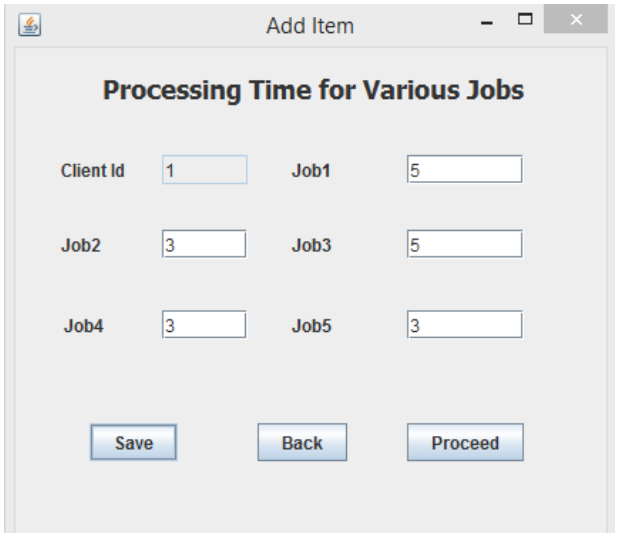

Figure 3: Processing time each job for cline id 1

A message showing the submission successfully will display. Click on proceed button to schedule and allocate jobs as shown in figure 4 below.

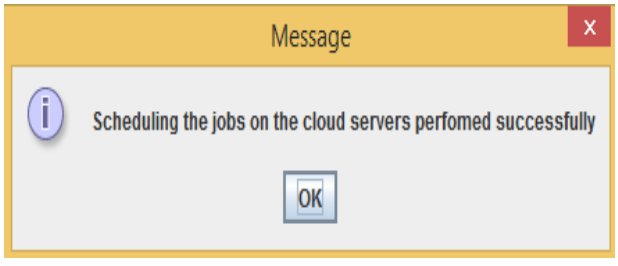

Figure 4: Scheduled & allocated servers for client's jobs

V. RESULTS & DISCUSSIONS

Various performance metrics were taken into consideration in order to measure and evaluate the selected job scheduling algorithms. These metrics include the amount of throughput, maximum finish time and total cost. These performance metrics are the most important and frequently used metrics in the previous works for evaluating the scheduling algorithms in cloud computing environment. In this section we use throughput metrics for comparing our algorithm with previous scheduling algorithms [11][14].

Throughput is the number of executed jobs, which is calculated to study its efficiency in satisfying the jobs dead-lines. The throughput is calculated using equation below:

$$\text{Throughput} \quad J = \sum_{j=J} X_j$$

Where Xj is:

$$X_j = \begin{cases} 1, & \text{job j has finished execution} \\ 0, & \text{Otherwise} \end{cases}$$

Where: j = Job from the list of jobs
J = List of jobs

V. CONCLUSION

Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. An essential requirement in cloud computing environment is scheduling the current jobs to be executed with the given constraints. The scheduler should order the jobs in a way where balance between improving the quality of services and at the same time maintaining the efficiency and fairness among the jobs.

In this work we also proposed a new hybrid approach for CPU scheduling in cloud computing environment. The hybrid approach uses Minimum Completion Time of various jobs with Optimistic load balancing approach on cloud servers. Then we compare the proposed method with the existing approaches in terms of one special metric - throughput. From various experiments we show that our approach works better than existing methods in terms of above metrics.

REFERENCES

[1] Zhang, Q., Cheng, L., and Boutaba, R., "Cloud computing: state-of-the-art and research challenges". Journal of Internet Services and Applications, 1(1): pp. 7-18,(2010).

[2] Sareen, P., "Cloud Computing: Types, Architecture, Applications, Concerns, Virtualization and Role of IT Governance in Cloud", International Journal of Advanced Research in Computer Science and Software Engineering, 3(3): pp. 533-538, (2013).

[3] R.Madhubala, "An Illustrative Study on cloud computing", "International Journal of Soft Computing and Engineering", Vol.1, issue.6, January 2012, pp. 286-290.

[4] Li, B., J. Li, J. Huai, T. Wo and Q. Li et al., 2009, "EnaCloud: An energy-saving application live placement approach for cloud computing environments", Proceedings of the International Conference on Cloud Computing, Sept. 21-25, IEEE Xplore Press, Bangalore, pp: 17-24. DOI: 10.1109/CLOUD.2009.72

[5] Gupta, P.K. and N. Rakesh, 2010. "Different job scheduling methodologies for web application and web server in a cloud computing environment", Proceedings of the 3rd International Conference on Emerging Trends in Engineering and Technology, Nov. 19-21, IEEE Xplore Press, Goa, pp: 569-572. DOI: 10.1109/ICETET.2010.24

[6] Yang, B., X. Xu, F. Tan and D.H. Park, 2011, "An utility based job scheduling algorithm for cloud computing considering reliability factor", Proceedings of the 2011 International Conference on Cloud and Service Computing, Dec. 12-14, IEEE Xplore Press, Hong Kong, pp: 95-102. DOI: 10.1109/CSC.2011.6138559

[7] Li, J., J. Peng and W. Zhang, 2011, "A scheduling algorithm for private clouds", J. Convergence Inform. Technol.", 6: 1-9. Li, J., M. Qiu, J. Niu, W. Gao and Z. Zong et al., 2011.

[8] Sindhu, S. and S. Mukherjee, 2011, "Efficient task scheduling algorithms for cloud computing environment". Commun. Comput. Inform. Sci., 169: 79-83. DOI: 10.1007/978-3-642-22577-2_11

[9] Vijindra and Sudhir Shenai. A, "Survey of Scheduling Issues in Cloud Computing", 2012, ICMOC-2012, 1877-7058, Elsevier Ltd, Doi: 10.1016/j.proeng.2012.06.337, page no: 2881 – 2888.

[10] Neetu Goel, Dr. R.B. Garg, "A Comparative Study of CPU Scheduling Algorithms", International Journal of Graphics & Image Processing |Vol 2|issue 4|November 2012.

[11] Monica Gahlawat, Priyanka Sharma, "Analysis and Performance Assessment of CPU Scheduling Algorithms in Cloud using Cloud Sim", International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 5 – No. 9, July 2013.

[12] Akilandeswari. P and H. Srimathi, "Survey on Task Scheduling in Cloud Environment", I J C T A, 9(37) 2016, pp. 693-698 © International Science Press.

[13] Shridhar Domanal, Ram Mohana Reddy Guddeti, and Rajkumar Buyya, "A Hybrid Bio-Inspired Algorithm for Scheduling and Resource Management in Cloud Environment", IEEE Transactions On Services Computing, VOL. X, NO. X, JULY 2016.

[14] Isam Azawi Mohialdeen, "Comparative Study Of Scheduling, Algorithms In Cloud Computing Environment", Journal of Computer Science, 9 (2): 252-263, 2013.