International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

Review: Performance Study of Job Scheduling Methods on Cloud

Rohit Ramesh Kumashi[1], S R Swamy[2]

[1]*Rohit Kumashi, 4th Year Undergraduate, Dept. of Computer Science & Engineering, RV College of Engineering, Bengaluru, Karnataka, India-560059*
[2]*Prof. S R Swamy, Dept. of Computer Science & Engineering, RV College of Engineering, Bengaluru, Karnataka, India-560059*

Abstract - *Cloud computing allows for distributed computing on a large scale and parallel processing. Computing is offered pay as you go service. The performance of tasks submitted by a user to cloud system dictates efficiency of cloud service. Scheduling of user tasks plays important role in improving cloud service performance. Job scheduling is the major scheduling that's performed. The paper gives an elaborate study of different task scheduling methods that are present in the*
cloud environment. This paper discusses parameters of the specified scheduling methods.

Key Words: Cloud computing; Minimum execution time; Job scheduling; Completion time; Cloudsim; Total Cost; Virtual Machines, make span, service-level agreements

1. INTRODUCTION

Cloud computing enables always present, accommodating, *a la carte* access to configurable computing resources containing pool (present in a shared network) that can be untethered with little management effort or interaction of service provider [1]. At present, cloud computing is providing dynamic services viz. data, applications, bandwidth, memory, and services of IT over the internet [2]. The scheduling of tasks dictates reliability and performance of cloud services. Scheduling can be at resource/task/workflow level. This paper focusses on mainly task scheduling algorithms. Users send a task – a request for computing jobs to the data centre; it's a small piece of work that within a given interval of time, should be executed. Job scheduling dispatches the jobs of cloud users, provided to the cloud provider using resources that currently usable.

Overall cloud performance can be increased by performing scheduling on different parameters. Data entry, software access, storage functions or processing can be included in a task. Accordingly, SLA's and requested services can be used by data center to classify tasks. Afterwards the task gets assigned to one available server. The requested task is performed by a server, and user in turn gets a response transmitted back.

Cloud task scheduling is a NP complete problem. In scheduling process of tasks, the jobs are submitted by the users to the cloud scheduler; it later gets the available resources' status and their properties by asking about the cloud information service and thereby allocating various tasks as per requirements of the task on different resources. Cloud scheduler assigns different tasks of users to unique VMs. VMs are assigned in an optimal manner by good scheduling [3]. Fig. 1 illustrates the scheduling model in Cloud Computing environment [16].

CPU utilization, cumulative throughput, and turnaround is improved by a good scheduling algorithm. Task scheduling is performed in different ways on the basis of different parameters considered - it can be allocated statically to different resources at time of compilation or be allocated dynamically at runtime [4][5].

Many adapted scheduling algorithms are put forward in cloud environment to improve performance of the total system like make span, percentage throughput and cost. That said, the complexity of selecting the best one increases as variety of scheduling algorithms is larger.

The paper aims at analysing and investigating 4 job scheduling algorithms under cloud environment, namely, Random Resource Selection, Round Robin, Minimum Completion Time and Opportunistic Load Balancing, based on their ability in guaranteeing fairness amongst the served jobs and providing quality service for the tasks. Additionally, the behaviour study of the scheduling algorithms aids in finding the job scheduling algorithm most appropriate for running jobs in cloud.

Following sections of the paper are presented as follows. Section II reviews works pertained to cloud-based job scheduling. Section III gives a brief analysis and comparison of different Cloud Computing environment task scheduling algorithms. Based on analysis done, Section IV presents various issues and directions for the future, while Section V relates to conclusion of the paper.

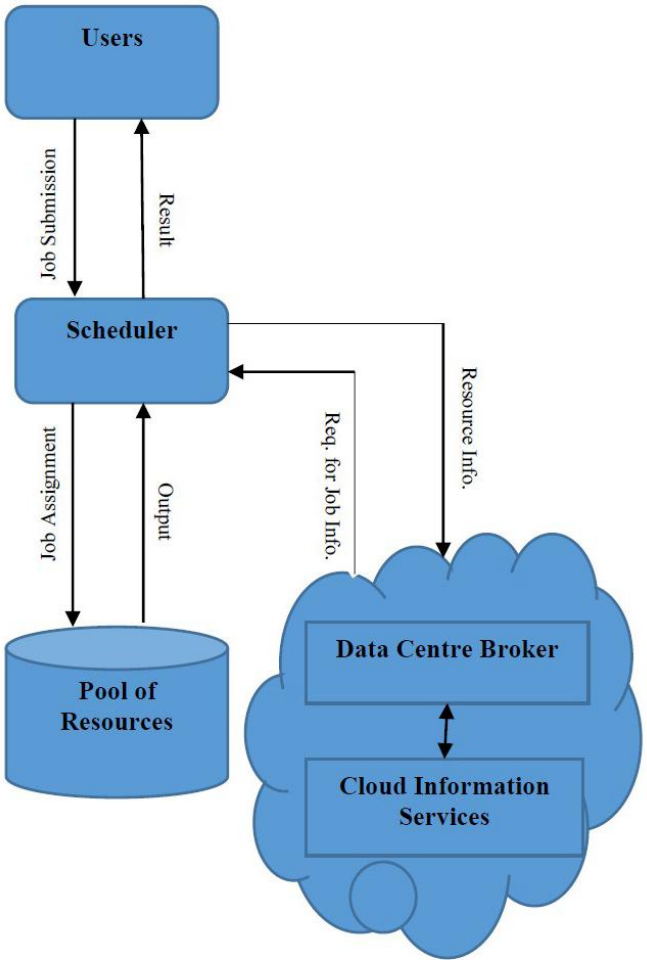

Fig -1: Job scheduling model in a Cloud Computing Environment

2. REVIEW OF RELATED WORKS

Cloud computing-based job scheduling has enticed research. Here job scheduling adopts a paradigm where a job in a cloud computing system is distinguished by factors such as dead-line and the corresponding utility obtained by its completion before deadline, its workload. These are considered in developing a scheduling algorithm that's effective, which is called paradigm of Utility Accrual.

Dynamic live placement of application having considerable energy efficiency on cloud was introduced by Li *et al.* (2009) that was called EnaCloud. A VM encapsulated the application, that supported live migration and scheduling of the applications, saving energy by reducing the quantity of running machines. Additionally, parallel processing of job execution in an environment in cloud has been tackled. In regards to this, they proposed a pre-emptive mechanism of job scheduling that made better cloud resource utilization. Introduction of 2 dynamic scheduling algorithms having feedback for this scheduling mechanism with the least average time of job execution to generate scheduling is also mentioned by them.

Furthermore, there is concentration on data center power consumption problem in 2011 Lin *et al.* Dynamic Round-Robin job scheduling approach reducing consumption of power for scheduling of virtual machines and their consolidation is proposed, that effectively does by attempting to deploy on servers, the virtual machines and migrate virtual machines among servers.

3. SELECTED TASK SCHEDULING ALGORITHMS

This paper considers 4 job scheduling policies in Cloud computing, namely, Opportunistic Load Balancing, Random Resource Selection, Minimum Completion Time, and Round Robin (RR). These are the most frequently and very commonly used Cloud computing-based task scheduling algorithms. Comparison of the algorithms is the main objective of this paper. Details of every job scheduling algorithm is explained by the following sections.

A. Random Resource Selection Algorithm

The selected jobs are assigned to the VMs available in random. The status of the VM under heavy/low load are not taken into consideration.

Therefore, a VM may be selected under heavy load and before the service is obtained, a long waiting time may be required for the job done by it. This algorithm has quite low complexity as no overhead or pre-processing is needed.

B. Minimum Completion Time Algorithm

The VM offering the least time of completion taking current load into account is allocated the selected job in the job scheduling algorithm relating to Minimum Completion Time [6]. The current load and processor speed on every VM determine which VM is used in the scheduling algorithm for minimum completion time. Determining the most appropriate machine to perform the job is determined by a scan of the available VMs. The most suitable VM's execution starts when it receives the incoming dispatched job.

C. Opportunistic Load Balancing Algorithm

The VM having the least load in comparison to other VMs gets the selected job that's dispatched in this strategy. Before sending the job, each VM's current load is scaled [7]. The job is then selected to run on the VM having the minimum load.

D. Round Robin Algorithm

The strategy has to decide the size of quantum and so the scheduler has an additional load, resulting in low throughput and longer average waiting time. The load is equally distributed to all resources in the cyclic strategy [8]. A node is allocated 1 VM in a cyclic fashion by the scheduler by employing this algorithm. In regard of process scheduling,

the round robin scheduling is analogous the round robin deployed on the cloud. Once a VM is assigned to a task, the job scheduler takes that task, completing it and moving on to the next task. Till all the VMs are allocated at least one node is repeated, later the first task is again returned to by the scheduler. Hence, in this case, to move on to the next task, the scheduler doesn't wait for node resources to be exhausted. As an example, if there are four jobs and there is to be scheduled four VMs, 1 VM is allocated to each task, given there are sufficiently available resources for every node to run the VMs.

4. EXPERIMENTATION AND ANALYSIS

Various criteria of evaluation are used for examining the performance of the described approaches. Cloudsim simulator is used to test and implement all scheduling strategies.

CloudSim is an extendable platform for simulating and generating graphical output of results expressed by the simulation via charts and tables by the person analysing it. This is desirable to summarize efficiently, the enormous statistics collected from simulation of task scheduling strategies. Crucial patterns of the output can be identified by a presentation that delivers a suitable result and helps the associated parameters to be correlated between from the given strategies. Simulation yields measures of statistics [9].

A. Performance Metrics

Evaluating the selected task algorithms for scheduling by measuring their merits is obtained via some metrics of performance [10]. They could be the likes of amount of percentage throughput, make span and total scheduling cost [11]. Earlier methods for scheduling employed these commonplace, occasional and performance related metrics to examine strategies for scheduling in a cloud-based application. Further explanation of the metrics of performance are elucidated in the following paragraphs.

a) Make span

It is the overall duration to complete every job in a queue of tasks. The make span should always be aimed to be kept at a minimum by an efficient and great method for scheduling [12]. The corresponding expression is given by:

$$\text{Makespan} = \text{Max}\ \{FT_j \mid \forall j \in J\} \quad \text{.....eqn(i)}$$

Wherein:
FTj = The time to finish job j belonging to job list J,
j = Job present in job list,
J = job list.

b) Total Cost

$$\text{Total Cost (TC)} = P_j * PC + \left(\sum_{f \in Fin_j} \text{Size}(f) + \sum_{f \in Fout_j} \text{Size}(f) \right) \times TrC \quad \text{.....eqn(ii)}$$

Wherein:
f = 1 file,
TC = Cost of scheduling all jobs,
Pj = The time to process job j,
PC = Cost of processing job j,
TrC = The transfer costs involved between the input files (Finj), and the output files (Fout j).

c) Throughput

Throughput can be explained as overall quantity of jobs that finished execution, being measured to evaluate its efficiency to satisfy the dead-lines of the jobs [13].

Calculate the throughput using:

$$\text{Throughput} \quad J = \sum_{j \in J} X_j \quad \text{.....eqn(iii)}$$

Where Xj is:

$$X_j = \begin{cases} 1, & \text{job j has finished execution} \\ 0, & \text{Otherwise} \end{cases} \quad \text{..... eqn(iv)}$$

Wherein:
j = Job present in job list,
J = the job list.

B. Results and Analysis

For each of the experiments (for make span, total cost, and throughput) different parameters values expressed in differing scenarios are considered during simulation as depicted in Table 1.

a) make span experimental analysis

This experiment views from the execution time perspective, the assignment of jobs to resources quality.

Table -1: Experiment parameters

Value of the Parameter	Value
Quantity of Jobs	50-1000
Number of VMs	100
Processing cost	7.64 INR
Transmission cost	7.64 INR

Table -2: Legend for the graphs depicted in figures 2, 3 and 4:

Dark blue	Random
Purple	Round Robin
Yellow	Opportunistic
Light blue	Min. completion time

From the second figure, in comparison to other strategies, the least value of make span is achieved in all scenarios by minimum completion time. This explanation of this fact can be attributed to fact that method of minimum completion time selects that VM that most appropriately can respond rapidly, thus executing the specified job and generating output required by the user. Note an observation wherein as total number of jobs rises, time of make span rises for minimum completion time. Five thousand five hundred seconds is approximately the time needed to run fifty jobs- this is the make span time, while the algorithm required about nine thousand seconds of make span at the time the job quantity reached a thousand jobs. The largest time is taken by the random resource selection strategy and hence it is the worst performing, this can be attributed to the fact that such a method attempts in a random manner to distribute job containing set over the virtual machine and the constraints of job are not taken into consideration.

b) Total cost experimental analysis

This analysis discusses the effect of job quantity upon the entire cost incurred when the jobs given to each of the VMs is executed by each one of them respectively. From figure 3 it is evident that the number of jobs assigned highly influences the total cost in all methods for scheduling tasks. The highest cost is produced by minimum completion time overall in comparison to rest of the specified scheduling methods in this paper, owing to the strategy achieving the biggest quantity of jobs received. The method that yielded the next highest cost was opportunistic load balancing scheduling algorithm. The strategy trumping all others in total cost (being lesser) was the Random Resource Selection method. Despite that this algorithm achieves similar cost when stacked up against the strategy of Round Robin in the scenario of jobs executed being around five hundred to about seven hundred. The implication is that the number of executed jobs has strong and intuitive relation to the total cost. Additionally, the total cost highly depends upon the number of executed jobs.

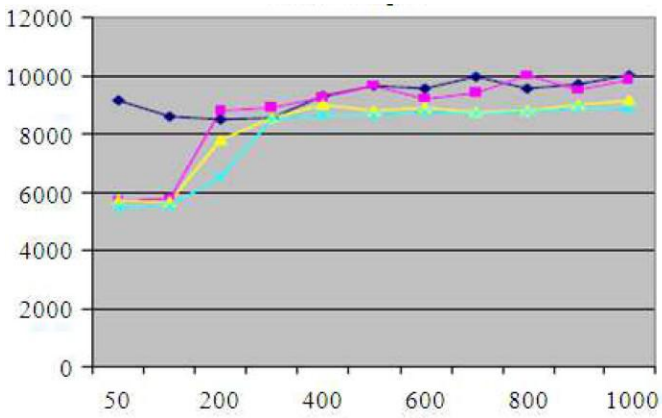

Fig -2: make span (in sec) vs no. of Jobs.

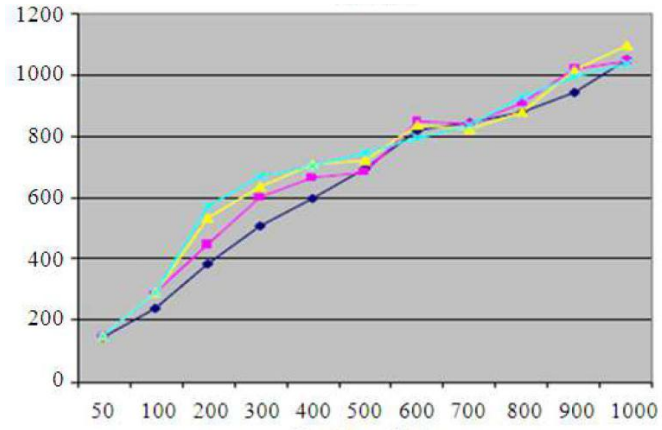

Fig -3: Total cost vs no. of Jobs

c) Percentage of Throughput experimental analysis

Note from figure 4 that other scheduling algorithms are thoroughly outperformed by the minimum completion time in all the considered cases; this can be attributed to the fact that the most appropriate VM capable of accomplishing the

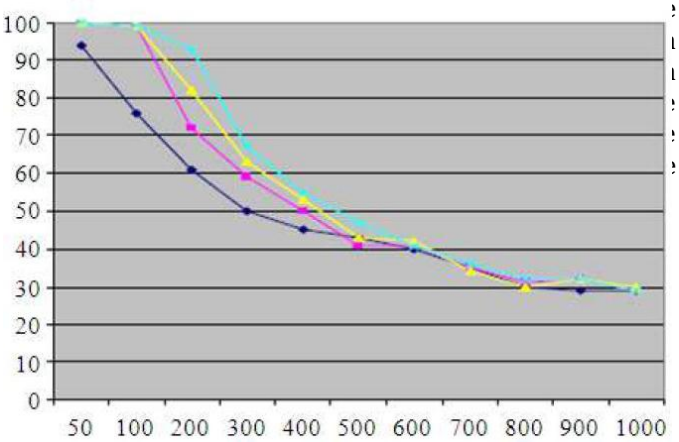

Fig -4: % of throughput vs no. of jobs

5. CONCLUSION AND FUTURE WORK

It is shown, based on the simulation results that some of the scheduling algorithms are beneficial to be used in Cloud

computing. A single algorithm that provides superior performance with respect to various types of quality services does not exist. This is because, a job scheduling algorithm needs to be chosen based on its ability to ensure good quality of service with reasonable cost and should maintain fairness by fairly distributing the available resources among all the jobs and respond to the constraints imposed by the users.

So, an algorithm is needed ensuring an improvement in being available and reliable in cloud related application. Since job scheduling is challenging under multi-cloud environment, an idea exploiting human inspired optimization [14][15] could be devised; additionally, having self-adaptive characteristics. This proposed method of brain storming with people with an addition of being accommodating(adaptive) by itself, will rank the individuals adaptively based on the improvement in the solutions. A consequence of this would be that there would be an improvement in suggestion of relevant solutions to then allow for required updating to be performed. Keeping this in mind, the scheduling method shall be executed. Allocating jobs for non-homogenous cloud's resources would then be expressed as brain storm process encoding. The resulting scheme related to the job schedule will be pitted against various constraints related to job performance like make span, completion of job, rate of utilization of resources. A combined objective model will be derived and subjected to solve job scheduling problem using the proposed optimization algorithm. This can then be compared with existing algorithms.

ACKNOWLEDGEMENT

I would like to acknowledge my guide, Prof. S R Swamy for giving invaluable suggestions while writing the paper.

REFERENCES

[1] D. Peter Mell, Timothy Grance "The NIST definition of Cloud Computing (September, 2011)", Accessed on May, 2014.

[2] Vijendra and Sudhir Shenai. A, "Survey of Scheduling Issues in Cloud Computing", 2012, Elsevier Ltd.

[3] R.Buyya, C.Yeo, S.Venugopal, J.Broberg, and I.Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation computer systems, vol.25, no.6, pp.599– 616, 2009.

[4] L.M.Vaquero, L.Rodero Merino, J.Caceres, and M.Lindner, "A break in the clouds: towards a cloud definition," ACMSIG COMM Computer Communication Review, vol.39, no.1, pp.50–55, 008.

[5] Yang, B., X. Xu, F. Tan and D.H. Park "An utility based job scheduling algorithm for cloud computing considering reliability factor" Proceedings of the 2011 International Conference on Cloud and Service Computing, IEEE Xplore Press.

[6] Minimum Completion Time for Power-Aware Scheduling in Cloud Computing, IEEE, Nawfal A. Mehdi; Ali Mamat; Ali Amer; Ziyad T. Abdul-Mehdi,2011.

[7] M. Mäkeläinen, Z. Khan, T. Hänninen and H. Saarnisaari, "Algorithms for opportunistic load balancing cognitive engine," 2014 9th International Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM), Oulu, 2014, pp. 125- 130.

[8] Silberschatz, Abraham; Galvin, Peter B.; Gagne, Greg (2010). "Process Scheduling". Operating System Concepts (8th ed.). John Wiley & Sons (Asia). p. 194. ISBN 978-0-470-23399-3. 5.3.4 Round Robin Scheduling.

[9] Jinhua Hu, Jianhua Gu, Guofei Sun Tianhai Zhao "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment" 2010 IEEE.

[10] Sindhu, S. and S. Mukherjee "Efficient task scheduling algorithms for cloud computing environment" 2011.

[11] Lianying ZHOU, Xingping CUI, Shuyue WU, An Optimized Loadbalancing Scheduling Method Based on the WLC Algorithm for Cloud Data Centers, Journal of Computational Information Systems, 2013.

[12] Gao, S., Zhang, W. An efficient approach to simplify the calculation of makespan in permutation flow shop scheduling problem. Int J Adv Manuf Technol 35, 325–332 (2007). https://doi.org/10.1007/s00170- 006-0716-y.

[13] https://www.cl.cam.ac.uk/teaching/1516/OpSystems/pdf/04- Scheduling.pdf

[14] Ramezani, F., J. Lu, and F.K. Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization", International Journal of Parallel Programming, vol.42, no.5, pp.739-754, 2014.

[15] Shi Y. "Brain Storm Optimization Algorithm", In: Tan Y., Shi Y., Chai Y., Wang G. (eds) Advances in Swarm Intelligence. ICSI 2011. Lecture Notes in Computer Science, vol 6728. Springer, Berlin, Heidelberg, 2011.

[16] https://www.nicepng.com/ourpic/u2q8u2e6o0a9e6w7_scheduling-model-in-cloud-computing-environment-scheduling-in/