A Comparative Analysis of Performance Metrics of Different Cloud Scheduling Techniques

Ravi Kumar
H.C.T.M. Technical Campus, Kurukshetra University, Kurukshetra

Rajender Kumar
Assistant Professor, H.C.T.M. Technical Campus, Kurukshetra University, Kurukshetra

Abstract: With the passage of time, a lot of work has been done in field of cloud computing. Still a lot of advancement can be done in this field. This paper consists of a comparative analysis of three different algorithms i.e. Multi Queue Scheduling Algorithm, Genetic Algorithm and Particle Swarm optimization. The comparison will be done on parameters like makespan and algorithm scheduled time. The result has been concluded that in terms of makespan GA is optimal.

Keywords: Cloud Service Provider (CSP), makespan, execution time, cost etc.

I. INTRODUCTION

According to National Institute of Standards and Technology (NIST) [1] "Cloud computing is a model for implementing convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers storage, applications, and services) that can be rapidly provisioned and discharged with minimal management effort or service provider interaction". Scheduling algorithm is used to schedule the task with maximum estimated gain and execute the task in the queue. The different types of cloud scheduling are [2]: Cloud service scheduling is classified at user level and system level. At user level, it mainly considers the service regarding problems between the provider and the user. At system level, scheduling and resource management is done. Static scheduling permits pre-fetching of required data and pipelining of different stages of task execution. In case of dynamic scheduling, the execution time of the task may not be known and the allocation of tasks is done only as the application executes. In cloud environment, heuristic based scheduling can be done for more optimized results. Real Time Scheduling in cloud environment is done to increase the throughput and to decrease the average response time instead of meeting deadline.

II. LITERATURE SURVEY

The scheduling algorithms have been established in the area of grids and clouds .To meet the large number of users a lot of improvement has been required. For a feasible solution Ravi et al. provide analysis of different algorithms [3]. Wei Neng Chen et al. [4] proposed S-CLPSO which is integrated with seven heuristics to deal with different QoS constraints like the budget constraint, the deadline constraint and the reliability constraint. Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centers was proposed by Li Hongyou et al. [5] to reduce energy consumption in cloud data centers. Yang Xiaoguang et al. [6] worked on Improved Hybrid PSO algorithm which provides better search capability. The Method consists of master-slave mode structure. Budget Constrained Priority Based Genetic Algorithm (BCHGA) was suggested by Amandeep Verma et al. [7] to schedule workflow applications that optimize the total cost of workflow within the user's defined budget. An Improved Xen Credit Scheduler for I/O Latency-Sensitive Applications on Multicores was proposed by Lingfang Zeng et al. [8]. This paper presented an improved Credit scheduler in Xen to facilitate such tasks on multicore platforms. Tristan Glatard et al. proposed controlling the deployment of virtual machines on clusters and clouds for scientific computing in CBRAIN [9]. M. Geethanjali et al. [10] proposed Multi-Objective Real Time Scheduling Algorithm (MORSA) for Multi Cloud Environments. Multi Queue Job Scheduling Algorithm was proposed by A V Karthick et al. [11] to overcome the fragmentation problem and to reduce the starvation with in the process. Amandeep Verma and Sakshi Kaushal [12] proposed Bi-Criteria Priority Based Particle Swarm Optimization Algorithm for Workflow Scheduling in Cloud. BPSO was designed to schedule workflow tasks over the accessible cloud resources that minimized the execution time and the execution cost under the user specified deadline and budget constraints. Eco aware algorithm was proposed by Xiang Deng, Di Wu, Junfeng Shen, and Jian He [13] for eco-aware power management and load scheduling for geographically distributed green cloud centers. Online Multi-Resource Scheduling for Minimum Task Completion Time in Cloud Servers was proposed by Mohammad Javad Norooz Oliaee et al. [14] to provide simple and efficient online scheme for scheduling cloud tasks requesting multiple resources, such as CPU and memory. Rate-Adaptive Scheduling Policies for Network Stability and Energy Efficiency was proposed by Matthew Andrews et al. [15]. Enhancing cloud computing reliability using efficient scheduling by providing reliability as a service was proposed by Abishi Chowdhury and Priyanka Tripathi [16]. Multi Queue Genetic Algorithm proposed by Neha et al. [17] to minimize the makespan. The algorithm performs proper utilization of resources and reduces starvation problem. On the basis of the entire previous algorithm we compare the three algorithms.

III. MULTI QUEUE SCHEDULING ALGORITHM

In Multi Queue Scheduling approach, tasks were sorted in ascending order and stored in three queues based on the percentage i.e. 40% in small queue, another 40% in medium and remaining 20% in large queue. One task is taken from each queue and added to the schedule which is then sequentially assigned to resources [11].

A) Pseudo code for Multi Queue Scheduling Algorithm

Ql- size of long queue, Qs- size of small, Qm- Size of medium queue, Sn-Number of tasks from small queue, Sm-Number of tasks from medium queue, Sl- Number of tasks from long queue .

i. Start
ii. Create N tasks randomly and also initialize parameters like Input file size, Output file size and Tasks instruction in millions.
iii. Sort the task and set in three queues
iv. Sn=Qs/Ql
v. Sm=Qm/Ql
vi. For all the tasks in a queue
vii. Fetch Sn tasks from small queue, Sm tasks from medium queue and Sl tasks from long queue
viii. End
ix. Start simulation on MQS scheduler
x. Calculate makespan and Scheduled Time
xi. Stop

IV. GENETIC ALGORITHM

For better allocation of resources and to minimize makespan Genetic Algorithm is used. The various genetic operators have been put on the schedule to get the optimized schedule. The cycle will be repeated until it reaches the termination. Initial population is first created. The algorithm arranges the tasks which act as a gene in the individual or chromosomes. Here fitness is the makespan. Execution time is calculated by the million instructions (mi) of the tasks divided by the million instructions per seconds (mips) of the resources. Fitness function arranges the schedule in such a way that it minimizes the overall execution of the schedule. Tasks are assigned to such resource which will take least execution time. And the process continues for all the tasks in a schedule. If the task get the resource which was earlier assigned then the current resource time is added with new task fitness value. After that tournament selection is performed in which the best individual is stored. Uniform crossover is performed here and the crossover rate is fixed. One value is chosen arbitrarily. If the value is greater than crossover rate it selects value from first individual otherwise it selects from second individual. Mutation is also based on mutation rate. The best chromosome is stored.

A) Pseudo Code for Genetic workflow

i. Start
ii. While (i<maxiter)
iii. Create population
iv. Evaluate fitness function
v. Selection (Tournament)
vi. Crossover (Uniform)

vii. Mutation (Uniform mutation)
viii. Store best individual
ix. End of while Loop
x. Start simulation on GA Schedule
xi. End

V. PARTICLE SWARM OPTIMIZATION

Particle Swarm optimization is local based search. We consider this technique to identify which technique is much better GA or PSO. Location and velocity vector is initialized. Fitness function value is calculated. The execution time for each schedule is the summation of the execution time of each task on a particular resource. Particle velocity and location vector is continuously updated in terms to minimize makespan. Fitness function: Once again fitness value is calculated and the value i.e. minimum will taken into consideration. Update individual optimal value and global value: optimal value and global value is updated. The process continues until reach the termination condition.

A) Pseudo Code for PSO workflow

i. Start
ii. While (i<maxiter)
iii. Initialize location and vector velocity
iv. Evaluate fitness function
v. Update particle velocity and location vector
vi. Calculate fitness function
vii. Update individual global and optimal value
viii. End of while Loop
ix. Start simulation on PSO Schedule
x. End

VI. RESULTS

In a cloud environment, tasks may be of different numbers and parameters are also different that means there will be hundreds of thousands of tasks that have different execution time. Here the tasks are static generated and resources are also static. The simulation has been performed for different number of tasks and executes multiple times for the same number of tasks. In a GA workflow, population size and number of iterations also played an important role.

Makespan value is calculated for all algorithms. Simulation has been performed multiple times for same number of tasks for more accuracy and average of all values are considered. The value is different in different run so we calculate average of all values for better accuracy**.** The graph below evaluates the average value of makespan for all three algorithm.

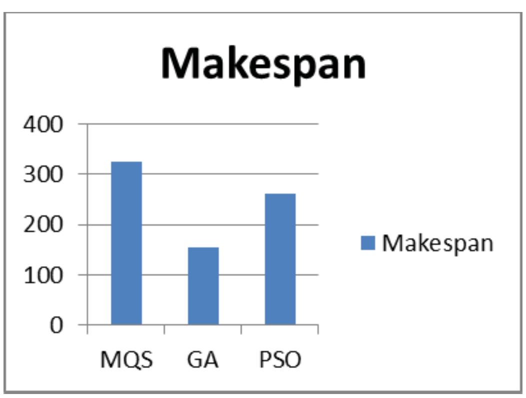

Figure 1. Average makespan for 250 tasks

The figure 2 illustrates the makespan with respect to different number of tasks. It shows that Genetic algorithm gives optimum results if resources are less and tasks are randomly initialized but PSO gives better results than MQS. Population size and number of generations are kept constant. Makespan calculated from Genetic Algorithm lies below the average makespan calculated from MQS algorithm and PSO.

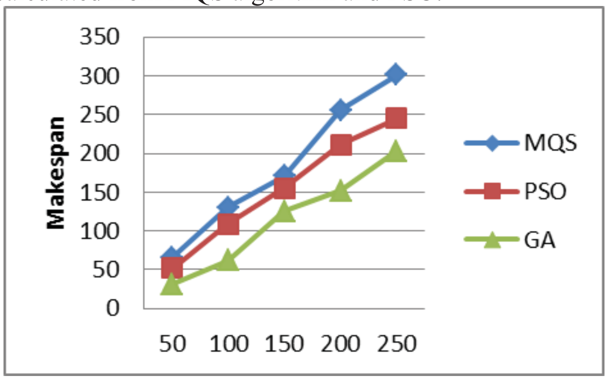

Figure 2. Comparison of MQS, PSO and GA algorithm on the basis of makespan value with respect to varied number of tasks

Simulation has been performed for different number of tasks. As shown in Figure 3. MQS algorithm is better than GA and PSO algorithm in terms of Algorithm Scheduled Time. But GA gives better results in terms of makespan when the GA schedule is run on cloud. The scheduled algorithm time is time taken by the algorithm to generate a schedule. Overall the three algorithms have their own advantages in different aspects.

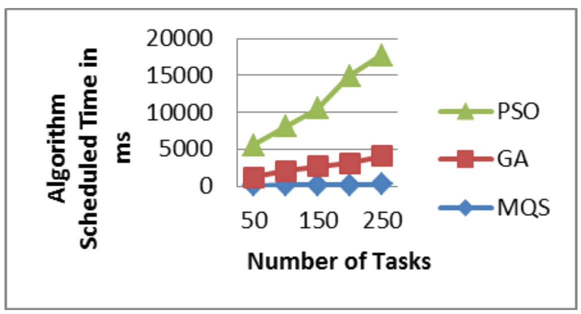

Figure 3. Comparison of MQS, GA and PSO algorithm on the basis of Algorithm scheduled Time in milli seconds with respect to varied number of tasks

VII. CONCLUSION & FUTURE SCOPE

It has been concluded that the schedule generated by GA is optimal. The simulation demonstrates that makespan calculated from GA is better than the makespan calculated from Multi Queue Scheduling Algorithm and PSO. Better makespan is possible only through proper allocation of resources which has been done by GA scheduler. The GA also increases the throughput of the user. In future, the work can be performed on other parameters like cost, drop rate; throughput etc. and any other technique can be performed to get better makespan than Genetic Algorithm and Particle Swarm Optimization.

REFERENCES

[1] Mell P., Grance T., (2011), "The NIST Definition of Cloud Computing," NIST Special Publication, pp. 800-145.
[2] Chawla Y., Bhonsle M., (2012), "A Study on Scheduling Methods in Cloud Computing," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), pp. 12-17.
[3] Kumar R., Panghal A., (2016), "Recent Advancement in scheduling algorithm in cloud computing environment," 5(3), IJSRD.
[4] Chen W.N., Zhang J., (2012), "A Set-Based Discrete PSO for Cloud Workflow Scheduling with User-Defined QoS Constraints," IEEE International Conference on Systems, Man, and Cybernetics, pp. 773-778.

[5] Hongyou Li, J. Wang, J.Peng, J.Wang, T. Liu, (2013), “Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centers,” IEEE, pp. 114-124.
[6] Xiaoguang Y., Tingbin C., Qisong Z., (2013), “Research on Cloud Computing Schedule based on Improved Hybrid PSO,” IEEE 3rd International Conference on Computer Science and Network Technology, pp. 388-391.
[7] Verma A., Kaushal S., (2013), “Budget Constrained Priority based Genetic Algorithm for Workflow Scheduling in Cloud,” IET Fifth International Conference on Advent in Recent Technologies, pp. 216-222.
[8] Zeng L, Wang Y, Shi W., Feng D., (2014), “An Improved Xen Credit Scheduler for I/O Latency-Sensitive Applications on Multicores,” IEEE, pp. 267-274.
[9] Glatard T., Rousseau M., Rioux P., Adalat R., Evans A. C., (2014), “Controlling the deployment of virtual machines on clusters and clouds for scientific computing in CBRAIN,” IEEE, 14th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, pp.384-393.
[10] Geethanjali M., Sujana M.J.A.J., Revathi T., (2014), “Ensuring Truthfulness for Scheduling Multi-objective Real Time Tasks in Multi Cloud Environments,” IEEE International Conference on Recent Trends in Information Technology, pp. 1-7.
[11] A.V. Karthick, Dr.E.Ramraj, R.Ganapthy Subramanian, (2014), “An Efficient Multi Queue Job Scheduling for Cloud Computing,” IEEE, World Congress on Computing and Communication Technologies, pp. 164-166.
[12] Verma A., Kaushal S., (2014), “Bi-Criteria Priority Based Particle Swarm Optimization Workflow Scheduling Algorithm for Cloud,” Proceedings of 2014 RAECS UIET Panjab University Chandigarh, pp. 1-6.
[13] Deng A. X., Wu D., Shen J., He J., (2014), “Eco-Aware Online Power Management and Load Scheduling for Green Cloud Datacenters,” IEEE Systems Journal, pp. 1-10.
[14] Norooz Oliaee M., Bechir Hamdaoui B., Mohsen Guizani M., Ghorbel M. B., (2014), “Online Multi-Resource Scheduling for Minimum Task Completion Time in Cloud Servers,” IEEE, IEEE INFOCOM Workshop on Mobile Cloud Computing, pp. 375-379.
[15] Andrews M., Antonakopoulos S., Zhang L., (2014), “Rate-Adaptive Scheduling Policies for Network Stability and Energy Efficiency,” IEEE /ACM TRANSACTIONS ON NETWORKING, pp. 1-10.
[16] Abishi T. Priyanka, (2014), “Enhancing cloud computing reliability using efficient scheduling by providing reliability as a service,” IEEE, pp. 99-104.
[17] L. Neha et al., (2015), “MQGA: Evolutionary Method for Cloud Task scheduling,” 5(6), IJARCSSE.