



MQGA: Evolutionary Method for Cloud Task Scheduling

Neha Latka, Dr. Sanjay Tyagi

Department of Comp. Sc. & Applications,
Kurukshetra University, Kurukshetra, Haryana, India

Abstract— *Cloud computing has become more popular among researchers from last few decades. Most of the time, the research work is carried out in the area of cloud scheduling especially task scheduling. Task scheduling is one of crucial part in cloud computing. Effective scheduling results in customer satisfaction. Task Scheduling is based on various parameters like cost, makespan, throughput, drop rate and energy. The proposed Multi Queue Genetic Algorithm (MQGA) mainly focuses on minimizing the makespan. The algorithm performs proper utilization of resources and reduces starvation problem. The proposed algorithm has been compared with the existing Multi queue scheduling algorithm.*

Keywords— *Cloud computing, makespan, resource utilization, multi queue genetic algorithm and multi queue scheduling.*

I. INTRODUCTION

Everyone defines the cloud computing in their own ways. 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”. According to IBM [2] “the cloud is the distribution of on-demand computing resources everything from applications to data centres over the Internet on a pay-for-use basis”.

Cloud scheduling plays significant role in cloud computing. In cloud computing user may use hundreds of thousands of virtualized resources which are impossible to assign manually [4]. Hence to assign resources to each task efficiently, scheduling plays an important role in cloud computing. There are many techniques developed from last few decades for effective scheduling, but all the techniques have some problem. The proposed scheduling method has been designed for effective scheduling of tasks. The algorithm mainly focuses on minimizing the makespan. The algorithm also deals with effective resource utilization and reducing starvation problem. Minimizing the makespan also increases the throughput. The testing of the schedule has been done by the cloudSim. CloudSim is a simulation toolkit that provides cloud environment to the researcher.

II. RELATED WORK

The most difficult task in cloud computing is cloud scheduling. The major factors contributing to cloud scheduling are cost, makespan etc. Many researchers have developed various cloud scheduling techniques on the basis of cost, makespan, throughput etc. Wei Neng Chen and Jun Zhang [5] proposed S-CLPSO i.e. discrete version of comprehensive learning PSO to deal with budget constraint, the deadline constraint and the reliability constraint. Budget Constrained Priority Based Genetic Algorithm (BCHGA) was suggested by Amandeep Verma and Sakshi Kaushal [6]. The aim of the algorithm was to schedule workflow applications that improved the total cost of workflow within the user’s defined budget. For multimedia cloud, Xiaoming Nan, Yifeng He and Ling Guan [7] proposed queuing model based resource optimization method for better quality of service and the resource cost. M. Geethanjali et al. [10] proposed Multi-objective Real Time scheduling algorithm (MORSA) for Multi Cloud Environments to achieve minimum cost and deadline constraints of the user tasks. All this techniques are mainly based on minimizing the cost. The above techniques do not deal with makespan, resource utilization and starvation problem. Tri Queue Scheduling (TQS) algorithm was developed by A V. Karthick, Dr. E. Ramaraj and R. Kannan[8] based on Dynamic Quantum Time to reduce the fragmentation problem and starvation within the process. Round robin fashion was used to give equal importance to all jobs based on dynamic quantum time. Fan Zhang et al. [9] proposed a multi-objective scheduling (MOS) algorithm to reduce scheduling overhead time and to reach optimal performance. Husnu S. Narman et al. [11] proposed Dynamic Dedicated servers scheduling algorithm (DDSS) for multiple priority level classes in cloud computing for calculating throughput and drop rate. Multi Queue Job Scheduling algorithm (MQS) was proposed by A V Karthick, Dr. E. Ramraj and R. Ganapthy Subramanian [12] to overcome the fragmentation problem and to reduce the starvation with in the process. These techniques were based on minimizing the schedule time but it can be improved further. MQGA gives better completion time of a schedule as compared to above work.

III. PROPOSED WORK

The proposed Multi Queue Genetic Algorithm (MQGA) requires the schedule generated by multi queue scheduler to further optimize the makespan and efficient utilization of resources. Proper allocation of resources results in the reduction of makespan. The schedule obtained from MQGA algorithm is better than schedule obtained from MQS algorithm. The GA has been implemented on the schedule received from multi queue scheduler. The proposed work mainly focuses on batch of jobs.

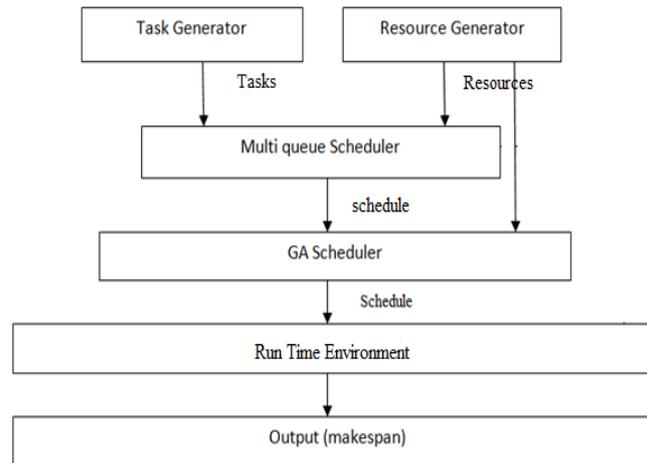


Figure 1 Block diagram of Multi Queue Genetic Algorithm

Figure 1 describes the steps involved in multi queue genetic algorithm. Task generator randomly generates the tasks, and uniquely assigns the task id & mi (million instructions). On the other hand, resource generator generates resources which are static. The tasks and resources are submitted to multi queue scheduler. It sorts the jobs in ascending order and creates three queues i.e. 40% task in small queue, another 40% in medium queue and remaining in large queue and it selects job from each queue. GA scheduler takes schedule from multi queue scheduler and the schedule is optimized. The output will be the makespan that has been calculated after the optimized schedule received from GA scheduler. For the makespan of multi queue scheduler, the schedule is directly sent to the cloud environment

TABLE I DEFINITIONS OF NOTATIONS

N	Number of tasks
T	Task instructions represented in millions
I _f s	Input file size
O _f s	Output file size
R	Number of instructions executed by resources and represented in mips i.e. millions instructions per second
L _q	Size of long queue
S _q	Size of small queue
M _q	Size of medium queue
N _s	Number of tasks taken from small queue
M _s	Number of tasks taken from medium queue
L _s	Number of tasks taken from long queue i.e. always 1
Cr	Crossover rate (0.5)
Mr	Mutation rate (0.15)
E _t	Execution time of tasks on a particular resource
Maxiter	Maximum number of iterations

Pseudo code for Multi Queue Genetic Algorithm

1. Start
 2. Create N tasks randomly and also initialize parameters like I_fs, O_fs and T
 3. Sort the tasks and set them in three queues
 4. $N_s = S_q / L_q$
 5. $M_s = M_q / L_q$
 6. For all the tasks in a queue
 7. Fetch N_s tasks from small queue, M_s tasks from medium queue and L_s tasks from long queue
 8. End
 9. Send schedule to the GA scheduler to get optimal schedule
 10. start simulation on ga scheduler
 11. Calculate makespan
- Stop

A. Schedule through GA Scheduler

The MQS scheduler passes the schedule to GA scheduler. The various genetic operators have been put on the schedule to get the optimized schedule [3].

Steps involved in Genetic algorithm:

The cycle will be repeated until generation count is less than maximum generation value.

- i. Create Population: One member of initial population will be the schedule received from Multi queue scheduler. Other members of population will be generated randomly. The algorithm arranges the tasks randomly which act as a gene in the individual or chromosomes.
- ii. Calculate fitness function: 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.

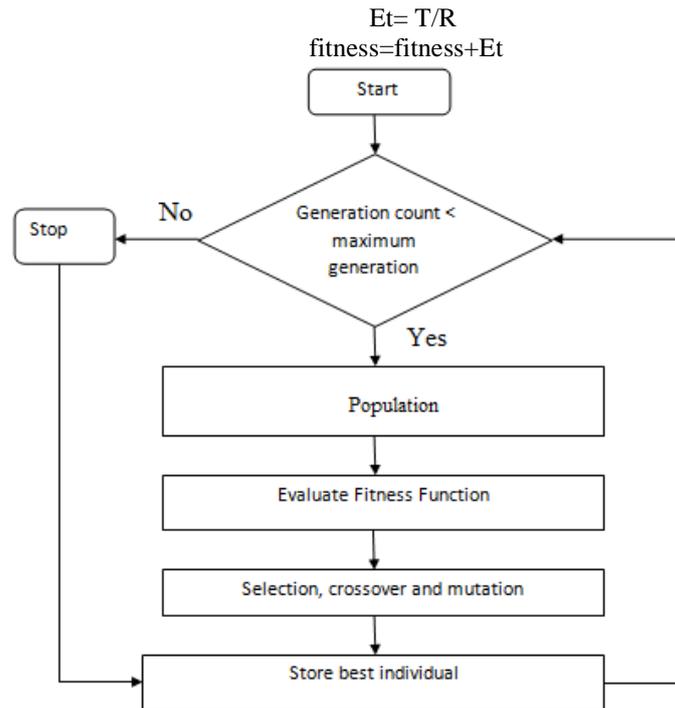


Figure 2 Block diagram of Multi Queue Genetic Algorithm

- iii. Selection: Tournament Selection is performed in which the best individual is stored.
- iv. Crossover: Uniform crossover is performed here and Crossover rate (Cr) is fixed. One value is chosen randomly. If the value is greater than crossover rate it selects value from first individual otherwise it selects from second individual.
- v. Mutation: Mutation is also based on mutation rate (Mr) and the best individual is stored.

IV. RESULT AND DISCUSSION

Both the proposed MQGA and existing MQS algorithm were tested on a CloudSim. Makespan of MQGA was compared with MQS makespan in order to prove the effectiveness of MQGA algorithm. Figure 3 describes the makespan with respect to generation count. Initially the makespan is very high. As the generation count increases, makespan is reduced and after a certain point makespan is constant which is the reduced makespan calculated by MQGA.

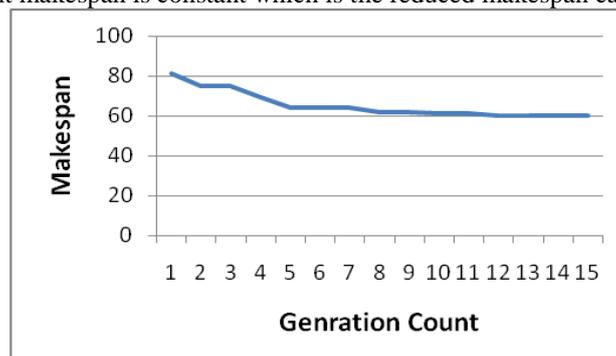


Figure 3 Makespan with respect to generation count

Comparison of makespan calculated by proposed MQGA and existing MQS is shown in figure 4. The results cleared that makespan calculated by MQGA is better than MQS even when the number of tasks is increased. The MQGA is effective for different number of tasks.

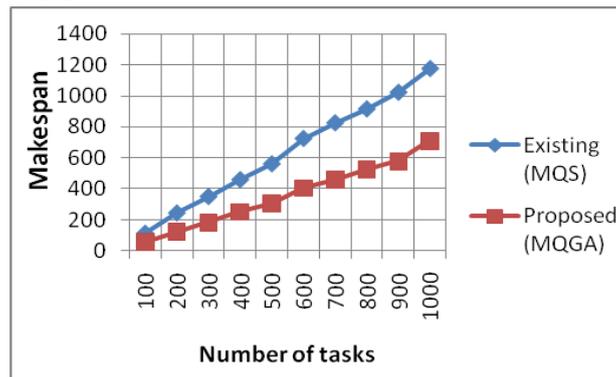


Figure 4 Comparison of Makespan with respect to number of tasks

V. CONCLUSIONS AND FUTURE SCOPE

This paper presented the Multi Queue Genetic Algorithm (MQGA) to schedule applications that minimizes the makespan and also makes efficient utilization of resources. The MQGA algorithm has been compared with the existing MQS algorithm and the effectiveness of Multi Queue Genetic Algorithm was proved.

For further research, the makespan can be reduced by other techniques and the experiment can be performed in a live environment.

REFERENCES

- [1] NIST Definition of cloud computing v15, csrc.nist.gov/groups/SNS/cloud-computing/cloud-defv15.doc.
- [2] <http://www.ibm.com/cloudcomputing/in/en/what-is-cloud-computing.html>
- [3] S.N.Sivanadam, S.N. Deepa, "Introduction to Genetic Algorithms", Springer, 2008.
- [4] Vijendra, Sudhir Shenai, "Survey on cloud scheduling issues in cloud computing", Elsevier, Proeng 2012, pp. 2881-2888.
- [5] Wei Neng Chen and Jun Zhang, "A Set-Based Discrete PSO for Cloud Workflow Scheduling with User-Defined QoS Constraints", IEEE International Conference on Systems, Man, and Cybernetics, 2012, pp. 773-778.
- [6] Amandeep Verma and Sakshi Kaushal, "Budget Constrained Priority based Genetic Algorithm for Workflow Scheduling in Cloud", IET Fifth International Conference on Advent in Recent Technologies, 2013, pp. 216-222.
- [7] Xiaoming Nan, Yifeng He, Ling Guan, "Queuing model based resource optimization for multimedia cloud", Elsevier, 2013, pp. 928-942.
- [8] A.V.Karthick, Dr.E.Ramraj and R.Kannan, "An Efficient Tri-Queue Job scheduling using Dynamic Quantum", International Conference on Green Computing, Communication and Conservation of Energy, 2013, pp. 871-876.
- [9] Fan Zhang, Junwei Cao, Keqin Li, Samee U. Khan, Kai Hwang, "Multi-objective scheduling of many tasks in cloud platforms", Elsevier, 2014, pp. 309-320.
- [10] M. Geethanjali, J. Angela Jennifa Sujana, T. Revathi, "Ensuring Truthfulness for Scheduling Multi-objective s in Multi Cloud Environments", IEEE International Conference on Recent Trends in Information Technology, 2014, pp. 1-7.
- [11] Husnu S. Narman, Md.Shohrab Hossain, Mohammed Atiquzzaman, "DDSS: Dynamic Dedicated Servers Scheduling for Multi Priority Level Classes in Cloud Computing", IEEE ICC, Next-Generation Networking Symposium, 2014, pp. 3082-3087.
- [12] A.V. Karthick, Dr.E.Ramraj, R.Ganapthy Subramanian, "An Efficient Multi Queue Job Scheduling for Cloud Computing", IEEE, World Congress on Computing and Communication Technologies, 2014, pp. 164-166.