Task Scheduling in Cloud Computing using Enhanced Genetic Algorithm

K.R.Prasanna Kumar[#1], K.Kousalya[*2]

[#]*Department of IT, Kongu Engineering College*
Tamilnadu, India
[1]krprasanname@gmail.com

[*]*Department of CSE*
Kongu Engineering College,Tamilnadu, India

***Abstract*— Cloud computing is recently a booming area and has been emerging as a commercial reality in the information technology domain. Cloud computing represents supplement, consumption and delivery model for IT services that are based on internet on pay as per usage basis. The scheduling of the cloud services to the consumers by service providers influences the time benefit of this computing paradigm. In such a scenario, Tasks should be scheduled efficiently such that the execution time can be reduced. A meta-heuristic based scheduling using genetic algorithm is introduced, which minimizes execution time. The enhanced genetic algorithm is developed by merging two existing scheduling algorithms for scheduling tasks taking into consideration their computational complexity and computing capacity of processing elements. The genetic algorithm generates the initial population randomly using some random functions whereas the proposed algorithm generates the initial population using the concept of Largest Cloudlet to Fastest Processor and Smallest Cloudlet to Fastest Processor. These enhancements lead to the minimization of makespan and increase the performance as compared to the existing traditional genetic algorithm.**

***Keywords*— Cloud computing, Scheduling, Meta-heuristic, Genetic algorithm**

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 [5]. 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 servers in order to 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.

II. Problem Definition

In the current cloud computing environment, VM resource scheduling only considers the current system condition and ignores the previous state of system which causes the system load imbalance. Number of VM migrations is more when most of the job scheduling takes place. This is largely due to granularity of VM resources and the large amount of data transferred in the migration with suspension of VM service. A scheduling algorithm needs to provide a scheduling strategy to enable effective job scheduling. This is achieved using genetic algorithm, historical data and the current state of the system.

An important challenge for the adoption of cloud computing in the scientific community remains the efficient allocation and execution of tasks. This proposed work is based on finding the optimal solution for scheduling the tasks to the vm’s in the cloud environment. The objective of the proposed work is to find the optimal solution by considering time complexity of the cloud system. Here makespan is considered to be the main parameter that needs to be concentrated while scheduling tasks. The algorithm tries to reduce this makespan by using genetic algorithm. Using this algorithm the makespan is reduced and which in turn automatically increases the performance of the system.

III. Existing

Resource allocation and job scheduling are the core functionalities of cloud computing. These functions are based on adequate information of available resources. The existing system is the traditional genetic algorithm which generates their population randomly. The GA proved to be an excellent algorithm for task scheduling in a non preemptive hard real-time scheduling environment. It also works best when migration is used to move tasks from one processor to another [2]. This task scheduling mechanism can not only meet user's requirements, but also get high resource utilization. But it needs more improvement as this whole algorithm is based on the accuracy of the predicted execution time of each task[4]. Dynamically sharing resources gives rise to resource contention. Cloud computing has some limitations in the existing system in traditional shared computing environment, and becomes a leading trend in distributed computing system. This traditional system can be changed effectively using Genetic Algorithm operators[1].

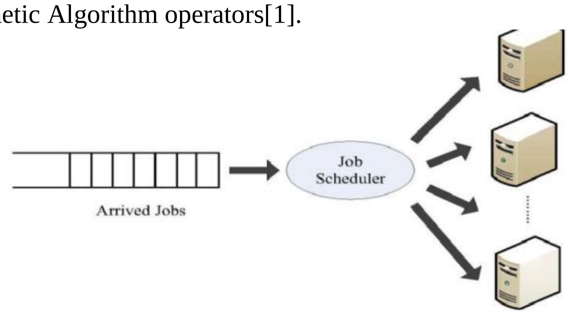

Fig 1. Sample scheduling

IV. PROPOSED WORK

proposed work focuses on optimizing the task scheduling algorithms with meta-heuristic algorithms that is Genetic Algorithm (GA)[6] in a cloud environment. With the combination of SCFP, LCFP and a GA as an optimization method, a new approach called Modified Genetic Algorithm for task scheduling is proposed. MGA is developed by modifying the initial population with LCFP, SCFP and by controlling the stochastic operators of standard genetic algorithm which lead to achieve a very good results and better efficiency of the algorithm than the standard genetic algorithm. This is for single user jobs in which the fitness will be developed to encourage the formation of solutions to achieve time minimization.

A. Generating Initial Population

The first step in the functioning of a GA is the generation of an initial population. It has been recognized that if the initial population to the GA is good, then the algorithm has a better possibility of finding a good solution and that, if the initial supply of building blocks is not large enough or good enough, then it would be difficult for the algorithm to find a good solution. The initial population generation in the proposed system also includes two main concepts.

• LCFP: Longest cloudlet to fastest processor. Here both the vm's and the cloudlets are arranged in descending order and then mapped accordingly.

• SCFP: Shortest cloudlet to fastest processor. Here the vm's are arranged in descending order and the cloudlets are arranged in ascending order and then mapped accordingly.

1) *Initial population generation :* After generating the initial populations with the use of the above mentioned concepts, some random populations are also generated as in the traditional GA. These generated populations are the initial population for our proposed system.

2) *Fitness calculation:* After generating the initial population the fitness of each individual is calculated. Here only the time complexity is considered. So fitness is evaluated using makespan of the system. Makespan is the time difference between the start and finish of a sequence of jobs or tasks.

Fitness (i) = makespan(i)

Where i indicates ith individual in a generation.

3) *Selection:* When the evaluation of each individuals are made, then selection procedure is performed to select the best fitted individual for carrying over them to further generation. This selection procedure use the below formulae to select the individuals.

Selection (i) = fitness (all) / fitness (i)

where i indicates ith individual in a generation.

Using this, the individuals are selected and they are chosen to perform the next level of GM operations.

B. Crossover, Mutation and Generating New Population

1) *Crossover:* The generated initial population is subjected to crossover operation in which some part of one parent is merged with another part of another parent.

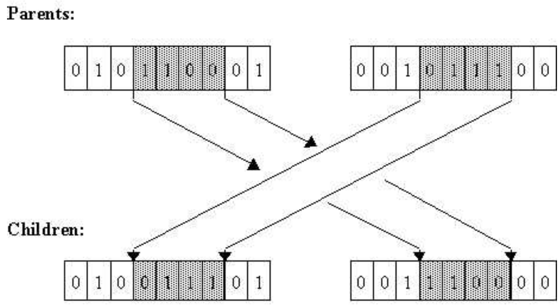

Fig 2. Crossover sample

2) *Mutation:* It is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. Here simple swap is used as mutation operator.

3) *New generation of population:* After performing the crossover and mutation operations on the generated initial population, new offspring are generated. The fitness value of these new children are identified and saved. After calculating fitness, selection operation is performed and the parents for next generation are selected based on the output of selection operation. The process is repeated until the total number of iterations specified. After that the individual with high fitness value is said to the best solution out of the given population.

V. RESULT AND DISCUSSION

The experimental setup is done using cloudsim simulator toolkit. It is widely used by most researchers to simulate the cloud environment [10]. The number of vm's are kept common. The number of cloudlets (jobs) are varied to get the simulation result.

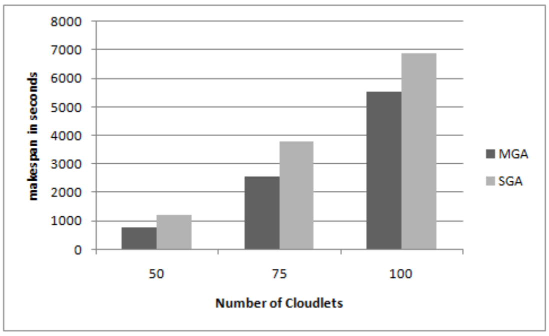

Fig 3. Results

Tested by varying the number of cloudlets in the range of 50, 75 and 100. Figure 3. shows our experiment results. The value of makespan is the most important indicator which reflects the performance of the system. Here the makespan value for 50 cloudlets assigned to 10 VM's is decreased. Similarly for 75 cloudlets and 10 VM's pair the makespan value is reduced by 25.1 seconds in MGA.

VI. CONCLUSION

Clouds enable the users to use utility services. An enhanced genetic algorithm for single user jobs is proposed in which the fitness is developed to encourage the formation of solutions to achieve the time minimization. Experimental results show that, under the heavy loads, the proposed algorithm exhibits a good performance. In future, this work can be further enhanced by supporting dynamic scheduling and priority of jobs for multiple users. The enhancements can also be made in the fitness function by considering the cost of usage or by introducing a user defined selection process which selects the individuals effectively.

REFERENCES

[1] Carlos Alberto Gonzalez Pico and Roger L.Wainwright (1994) 'Dynamic Scheduling of Computer Tasks Using Genetic Algorithms' IEEE World Conference on Computational Intelligence pp. 829-833.

[2] Sandeep Tayal (2011) 'Task Scheduling Optimization for the Cloud Computing systems' international journal of advanced engineering sciences and technologies Vol No. 5, Issue no.2, pp.111-115.

[3] Sasmita Kumari Nayak, Sasmita Kumari Padhy and Siba Prasada Panigrahi (2012) 'A novel algorithm for dynamic task scheduling' Future Generation Computer systems Vol No. 28, pp.708-717.

[4] Isam Azawi Mohialdeen (2013) 'Comparitive Study of Sheduling Algorithms in Cloud Computing Environment,' Journal of Computer Sciences Vol No.9, pp.252-263.

[5] Sih G C, Lee E A, 'A compile-time scheduling heuristic for Interconnection-constraint heterogeneous processor architectures,' IEEE Trans Parallel Distributed System, 1993, pp. 175- 187.

[6] Xiaonian Wua, Mengqing Denga, Runlian Zhanga, Bing Zengb, Shengyuan Zhoua, A Task Scheduling Algorithm based on QoS-Driven in Cloud Computing, Procedia Computer Science Volume, 2013,17, Pages 1162–1169

[7] Introduction to Cloud computing http://www.gartner.com/newsroom/id/1035013

[8] Introduction to genetic algorithm http://www.obitko.com/tutorials/genetic-algorithms/crossover-mutation.php

[9] Exploiting Dynamic Resource Allocation for Efficient Parallel Data Processing in the Cloud http://dsl.cs.uchicago.edu/TPDS_MTC/papers/TPDSSI-2010-01-0012.pdf

[10] Cloudsim http://www.cloudbus.org/cloudsim/