QoS-based Scheduling of Tasks using Bi-Objective Genetic Algorithm in Private Cloud

Revathy K
Master of Technology
Department of Computer Science and Engineering
NSS College of Engineering
Palakkad, India

Sindhu S
Associate Professor
Department of Computer Science and Engineering
NSS College of Engineering
Palakkad, India

ABSTRACT
Cloud is a high performance computing environment in which many users are allowed to utilize the data, storage, computations and services from all around the world. Cloud environment contains heterogeneous collection of systems and is very flexible. Still scheduling of tasks is a major issue in cloud computing environment. Efficient utilization of tasks can be obtained by proper scheduling of all the tasks submitted to the cloud. This paper considers two QoS(Quality of Service) aspects : deadline and makespan for obtaining best schedule of tasks in a private cloud. Bi-objective Genetic Algorithm is used with the aim to minimize the violation of deadline and makespan of tasks.

General Terms
Scheduling, Private cloud

Keywords
Cloud Computing, Deadline, Makespan, Genetic Algorithm

1. INTRODUCTION
Cloud computing can be defined as the set of hardware, storage, networks, services, and many interfaces that combine to deliver computing as a service for high performance computing tasks. Cloud services include the delivery of infrastructure, software and storage over the Internet based on demand from the users [1]. In a cloud computing system workload shift is handled by network of computers that make up the cloud. Hardware and software demands on the user's side decrease. User's computer simply needs to run the cloud computing system's interface software. Cloud computing generally involves delivering hosted services over the Internet. These services are divided into: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) [2]. Three distinct characteristics of cloud service are: it is sold on demand, elastic and the service is fully managed by the provider

Cloud is of two types public and private. Public cloud is the one which sells services to anyone on the Internet. (eg: Amazon Web Services) and private cloud is a proprietary network or a data center that supplies hosted services to a limited number of people. A set of virtualized and dynamically scalable computing power, services, platforms, and storage are delivered on demand to clients over the Internet on a pay as you go basis by several providers. Jobs that arrive into a system are diverse in nature and have independent execution on the virtualized operating system. This makes virtualization as the core characteristic of cloud computing. The resources have to be properly utilized for satisfying the demand.

Scheduling is used for distributing the requested resources among the consumers. Resources are allocated for the received task and in the cloud computing environment, these resources are shared. If the resources are not properly distributed then it will lead to resource wastage [3]. The datacenters uses significant portion of energy. Usage pattern affects amount of energy consumed by auxiliary and computing hardware resources. In other words, resource under-utilization incurs a higher volume of energy. Hence scheduling and virtualization are necessary energy-saving techniques. The problem of task scheduling is known to be NP complete since it is hard to find an optimal solution in polynomial time. Hence many heuristics are proposed for allocating resources to tasks. Some of them are min-min scheduling, max-min scheduling [3], dynamic priority scheduling algorithm (DPSA) [4], Genetic Algorithm (GA) [5] etc…

This paper considers genetic algorithm for task scheduling. The genetic algorithm can be used for getting the best schedule of tasks by utilizing historical data and current state of the system. The objective of this paper is to find the best schedule based on two QoS (quality of service) aspects: makespan and deadline of tasks using Genetic Algorithm. Fitness function of GA is to minimize the makespan and to reduce the violation of deadline of submitted tasks. The rest of the paper is organized as follows: section 2 contains the related works, section 3 explains scheduling using genetic algorithm, section 4 contains simulation and results and section 5 gives the future work.

2. RELATED WORK
Scheduling in cloud are studied using different algorithms in many of the works. Some of them are as follows: In [8] a scheduling model based on multi-objective genetic algorithm (MO-GA) and the research is focused on encoding rules, selection operators, crossover operators and the method of sorting Pareto solutions is proposed. The author in [6] proposed a meta-heuristic based scheduling, which minimizes execution cost and execution time as well. The author in [9] shows virtualization-based server consolidation which has been proven to be an ideal technique to solve the server sprawl problem by consolidating multiple virtualized servers onto a few physical servers leading to improved resource utilization and return on investment. In [10] we can see that the jobs arriving to the cloud system are normally high demanding efficient execution and dispatch at the peak hours. In [11] a Scheduling model based on minimum network delay using Suffrage Heuristic coupled with Genetic algorithms for scheduling sets of independent jobs, here the objective is to minimize the makespan. In [12] the problem of load imbalance and high migration costs is shown. Usually high number of Virtual Machine (VM) migrations and load imbalance occur if the scheduling is performed using the traditional algorithms. The author in [13] mainly focuses on

study of various problems and types of scheduling based on the genetic algorithm for cloud workflows. In [14] author shows that to attain maximum utilization of resources the tasks need to be scheduled. The main problem in scheduling is allocating the correct resources to the arrived tasks. In [15] the energy issue in task scheduling particularly on high performance system (HCSs) is investigated. The proposed approach analyzes the problem of scheduling precedence-constrained parallel applications on heterogeneous computing systems (HCSs) like cloud computing infrastructures. There is a novel heuristics based Energy Aware Resource Allocation (EARA) mechanism to allocate the user applications to the cloud resources that consumes minimal energy and incorporating the prioritization mechanism based on the deadline [16]. The consolidation problem can be as a modified bin packing problem and illustrates it with an example. This algorithm aims to find a minimal energy allocation of workloads to servers [17]. In [18] author presents two energy-conscious task consolidation heuristics aiming to maximize resource utilization and explicitly take into account both active and idle energy consumption. The author in [2] define an architectural framework and principles for energy-efficient Cloud computing. There are some main focuses on how to improve the energy efficiency of servers through appropriate scheduling strategies [18]. Taking full consideration of the relationship between the performance and energy consumption of servers, a new energy-efficient multi-job scheduling model based on the Google's massive data processing framework, MapReduce is proposed.

3. SCHEDULING USING GENETIC ALGORITHM

Genetic algorithm is a method of scheduling in which the tasks are assigned resources according to individual solutions (called schedules), and tells which resource is to be assigned to which task. Genetic Algorithm is based on the biological concept of population generation. Genetic Algorithms (GAs) are population-based metaheuristics based on the iterative application of stochastic operators on a population of candidate solutions. At the each iteration, individuals are selected from the population, paired and recombined in order to generate new ones which replace other individuals selected from the population either randomly or according to a selection strategy [6]. The main steps are as follows:

3.1 Initial Population

Set of individuals which are used in Genetic Algorithm to find the optimal solution are called Initial population. Every solution in the population is called as chromosome. Individuals are selected from initial population and some fitness operations are applied on those to form the next generation. In this paper the initial populations contains set of chromosomes containing the mapping of variable length tasks onto different physical machines. The population is encoded as m x n array: inipop[m][n] where m is the number of chromosomes $\{c_1, c_2 \ldots c_n\}$ and n is the number of tasks $\{t_1, t_2, \ldots t_n\}$. The elements of the array are the different virtual machines in which the corresponding tasks run. Each chromosome represents a different schedule.

	t_1	t_2	t_3	t_4	t_5	t_6	t_7
c_1	1	2	5	7	9	7	1
c_2	5	1	9	8	7	5	3

Fig 1: Initial Population

3.2 Fitness Function

Genetic Algorithm evaluates each chromosome by fitness function. Fitness function is used for the measurement of effectiveness of the solution according to the given objective and helps to know which chromosomes retain in the population. Incorrect fitness function may lead to delay in process. The productivity of any individual depends on the fitness value which is the measure of the superiority of an individual in the population. Larger the fitness value better will be the performance. In this paper fitness function is defined as to minimize the makespan and minimizing the violation of deadline. Minimizing the violation of deadline refers to finding a best schedule so as to reduce the number of tasks which take more time than the given deadline. Hence this is a bi-objective GA. The tasks of varying length are given to different VMs in the cloud. Space Shared Scheduling is used. Makespan[m] is the array containing the makespan of m tasks.

<u>Calculation of makespan:</u>

1. Generate a pxq array called etc[p][q] (expected time to complete) for storing the time taken by task p on VM q.
2. For chromosome c=1 to m do
   - 2.1 for task t=1 to n do
     - 2.2.1. r = inipop[c][t]
     - 2.2.2. makespan[m] = makespan[m] + etc[i][r]
     - 2.2.3. increment m

<u>Calculation of deadline</u>

1. input the deadline d.
2. count the number of tasks violating the deadline in each chromosome using inipop and etc matrix.

3.3 Selection

Selection mechanism is used to select an intermediate solution for the next generation. Chromosomes which satisfy both the fitness values together are considered fit. Based on the fitness function every individual is given a rank. By using this method, the individuals that are most fit gains more preference and are selected for next generation. Sort the makespan and deadline violations of all chromosomes. Select the first half populations satisfying both minimum makespan and minimum deadline violation for the next generation.

3.4 Crossover

Crossover operator pairs all the chromosomes and is used to combine two chromosomes to produce next generation chromosomes. New chromosomes are obtained from the mixture of parent chromosomes. In this paper single point crossover is used that is at the locus, remaining alleles are swapped from the parents to others takes place. If the crossover operations are not performed then the new generations resembles their parents. In this paper single point crossover is done based generating two random numbers and if they are equal, crossover is performed.

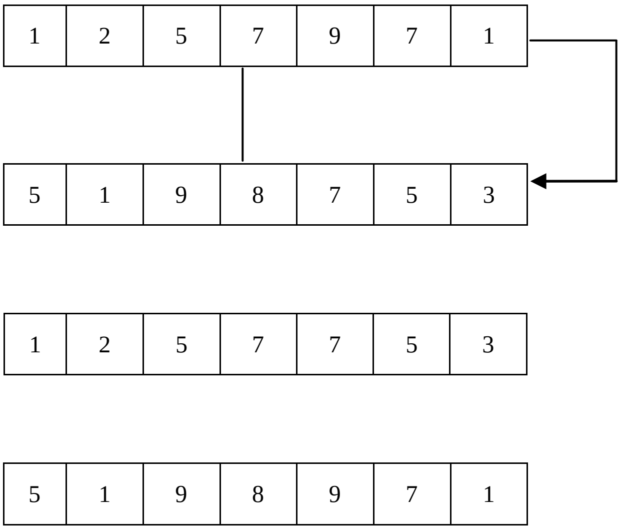

Fig 2: Crossover

3.5 Mutation

After crossover Mutation operation is performed that introduces genetic diversity in the population. This is done whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operator. One or more gene values in chromosome are altered from its initial state. Entirely new gene values being added to the gene pool. Genetic algorithm may be able to produce the better solution with this new gene values, than was previously. Here mutation is performed after every five iterations of GA.

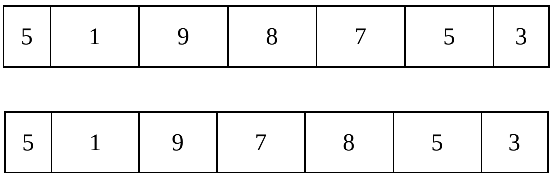

Fig 3: Mutation

3.6 Termination Condition

In this paper GA is iterated until a constant schedule is obtained.

Scheduling Algorithm using GA:

1. Input n number of tasks T = {t_1, t_2... t_n} of variable lengths.
2. Input m number of virtual machines V = {v_1,v_2…v_m} with variable MIPS.
3. Generate an Expected Time to Complete (ETC) matrix and store in array: etc of order p x q.

   p: number of tasks

   q : number of virtual machines(VMs)

   Elements of this array are found by: (Length of the task t_i)/(MIPS of VM v_j).
4. Initializing population.

   4.1. Create array inipop[m][n] representing the initial population

   4.2. for i=0 to m do

   4.2.1. Generate random VM v

   4.2.2. inipop[i][j] = v

   4.2.3. Increment j
5. Check the fitness of each individual.
6. Select the fit individuals for further iteration of GA.
7. Perform the crossover of chromosomes using single point crossover.
7. Mutate the chromosomes to generate new population after every 5 iterations of GA.
8. Repeat steps 5, 6, 7 until the chromosomes contain same schedule.

4. SIMULATIONS AND RESULTS

This paper uses CloudSim as the simulator tool for checking the performance of the algorithm. CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. This toolkit supports both system and behavior modeling of Cloud system components such as data centers, resource provisioning policies and virtual machines (VMs). It implements generic application provisioning techniques that can be extended with ease and limited efforts [7]. This paper considers Virtual Machines as resource and Cloudlets as tasks. With the help of GA and considering minimization of makespan and deadline, efficient schedule of tasks is obtained.

Table 1. GA parameters

Parameter	Value
No. of cloudlets	5-25
No. of VMs	5
No. of populations	24
No. of iterations of GA	300
Fitness Factors	deadline, makespan
Crossover type	one-point
Mutation type	simple swap
Termination condition	no. of iterations

The lengths of cloudlets and MIPS of the VMs are randomly generated. After the termination of GA we obtain the best schedules for varying number of cloudlets {5, 10, 15, 20, 25}. The makespan and total number of deadline violations of the tasks are minimum. No. of cloudlets vs makespan graph is shown below.

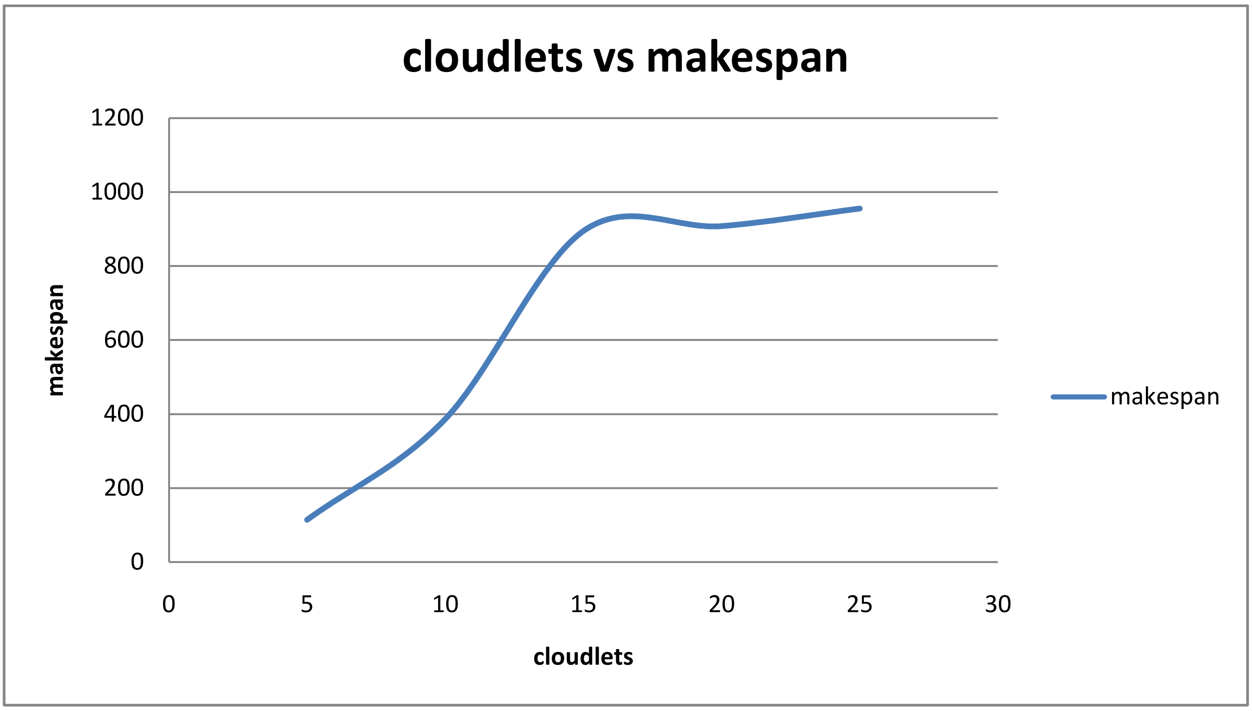

Fig 4: No. of cloudlets vs makespan

5. CONCLUSION AND FUTURE WORK

Scheduling of tasks in cloud by taking into account the QoS(Quality of Service) factors leads to efficient utilization of resources. In this paper the makespan and deadline violation are minimized for all the tasks submitted in the private cloud. In addition to makespan and deadline, maximum utilization of resources can also be added to the fitness factor to generate best schedule of tasks.

6. REFERENCES

[1] Armbrust M, Fox A, Griffith R, Joseph A D, Katz R, Konwinski A, Lee G, Patterson D, abkin A and Stoica I, "A view of cloud computing", Communications of the ACM, Vol. 53, No. 4, 2010, pp. 50-58.

[2] Anton Beloglazov, Jemal Abawajyb, Rajkumar Buyya. "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing", Future Generation Computer Systems 28 (2012) 755768.

[3] Saurabh Kumar Garg, Chee Shin Yeo b, Arun Anandasivamc, Rajkumar Buyya ," Energy-efficient Scheduling of HPC Applications in Cloud Computing Environments ", Elsevier, *7 Sep 2009*.

[4] Pardeep Kumar, Amandeep Verma," Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm "International Journal of Advanced Research in Computer Science and Software Engineering, 05/2012; 2(5):111-114..

[5] Zhongyuan Lee, Ying Wang, Wen Zhou, "A dynamic priority scheduling algorithm on service request scheduling in cloud computing", IEEE, 2011.

[6] Shaminder Kaur, Amandeep Verma , "An Efficient Approach to Genetic Algorithm for Task Scheduling in Cloud Computing Environment",***I.J*** Information Technology and Computer Science, 2012, 10, 74-79.

[7] Rodrigo N. Calheiros1, Rajiv Ranjan2, Anton Beloglazov1, C´esar A. F. De Rose3 and Rajkumar Buyya1 "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Published online 24 August 2010 in Wiley Online Library(wileyonlinelibrary.com). DOI: 10.1002/spe.995

[8] Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang4 and Bai-Nan Li5 "Job Scheduling Model for Cloud Computing Based on Multi- Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013.

[9] Siyuan Jing, Kun She, "A Generic Approach to Achieve Optimal Server Consolidation by Using Existing Servers In Virtualized Data Center", International Journal of Computer and Information Engineering 4:3 2010.

[10] Rashmi K S, Suma. V & Vaidehi. M, "Factors Influencing Job Rejections in Cloud Environment", International Joint Conference on "Emerging Intelligent Sustainable Technologies", (EISTCON-2012), Volume–3, 3rd & 4th May 2012, ISBN:978-93-81693- 76-6.

[11] Tarun Goyal and Aakansha Agrawal, "Host Computing Environment using Genetic Algorithm in cloud computing environment",International Journal of Research in Engineering & Technology (IJRET) Vol. 1, Issue 1, June 2013.

[12] Shailesh Sawant, "A Genetic Algorithm Scheduling Approach for Virtual Machine Resources in a Cloud Computing Environment", San Jose State University SJSU ScholarWorks, 2011.

[13] Savitha. P, J Geetha Reddy, " A Review Work On Task Scheduling In Cloud Computing Using Genetic Algorithm", International Journal of Scientific and Technology Research Volume 2, Issue 8, August 2013.

[14] A.Kaleeswaran,V.Ramasamy, P.Vivekanandan," Dynamic Scheduling of Data Using Algorithm in Cloud

Computing", International Journal of Advances in Engineering & Technology, Jan. 2013.

[15] V.Venkatesa Kumar, S.Vidhya, S Palaniswami," A Metaheuristic for Energy Aware Scheduling for Cloud Computing System using Hybrid GA", International Journal of Communications and Engineering Volume 05– No.5, Issue: 02 March2012.

[16] Thamarai Selvi Somasundaram, Kannan Govindarajan, T.D. Rohini, K. Kavithaa, R.Preethi, "A Novel Heuristics based Energy Aware Resour Allocation and Job Prioritization in HPC Clouds", www.annauniv.edu/care, Published online: 2012 .

[17] Shekhar Srikantaiah, Aman Kansal Feng Zhao, "Energy Aware Consolidation for Cloud Computing", 2008.

[18] Xiaoli Wang, Yuping Wang, and Hai Zhu, "Energy-Efficient Multi-Job Scheduling Model for Cloud Computing and Its Genetic Algorithm", Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2012.

[19] M. Mezmaza, N. Melabb, Y. Kessaci b, Y.C. Lee c, E.-G. Talbi b,d, A.Y. Zomayac, D. Tuyttens a, "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems", J. Parallel Distrib.Comput. 71 (2011) 1497–1508.

[20] Anton Beloglazov and Rajkumar Buyya, "Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints", IEEE Transactions on Parallel and Distributed Systems, Vol 24, NO 7, July 2013.