

HOST SCHEDULING ALGORITHM USING GENETIC ALGORITHM IN CLOUD COMPUTING ENVIRONMENT

TARUN GOYAL¹ & AAKANKSHA AGRAWAL²

¹Department of Computer Science, IT Shekhawati Institute of Engineering & Technology, Sikar, Rajasthan, India

²Department of Information Technology, GBPUAT University, Pantnagar, Uttarakhand, India

ABSTRACT

Cloud computing is a paradigm in which IT (information technology) application provide as a service. Cloud computing allows users to utilize the computation, storage, data and services from around the world in commercialize manner. In cloud environment, scheduling is the major issue. Scheduling is responsible efficient utilization of the resources. In this paper, a Scheduling model based on minimum network delay using Suffrage Heuristic coupled with Genetic algorithms for scheduling sets of independent jobs algorithm is proposed, the objective is to minimize the make span.

KEYWORDS: Cloud Computing, Suffrage Heuristic, and Genetic Algorithm

INTRODUCTION

Following distributed computing, parallel computing, grid computing, utility computing, Web 2.0. etc., the computer industry and academia put forward cloud computing model [1], which achieves generalization and commercialization of these previous models in some sense. Cloud computing, the long-held dream of computing, has the potential to transform a large part of the IT industry to makes software even more attractive as a service and shaping the way IT hardware is designed and purchase[3]. No doubts it would increasingly change the way people live and work. Cloud computing can be defined as “a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service providers and consumers”[1].

With the advancement of the Cloud, there are new ways of opening up on how applications can be built on the internet. There are various cloud service providers who are willing to provide large scaled computing infrastructure at a cheaper price and provide the infrastructure services in a very effective manner which the users can scale up or down at will. There are also large scaled software systems such as social networking sites and e-commerce applications gaining popularity today which can benefit greatly by using such cloud services to minimize costs and improve service quality to the end users.

The cloud computing is still at its infant stage and a very new technology for enterprises. Cloud computing is term used to describe both a platform and types of application. As a platform its supplies, configure and reconfigures server's, while servers can be physical machine or virtual machine. On the other hand, cloud computing describes application that are extended to be accessible through the internet and for this large data centers and powerful servers are used to host the web application and web services.

There are some important points in the definition to be discussed regarding cloud computing. Cloud computing differs from tradition computing paradigm as it is scalable, can been capsulated as an abstract entity which provides

different level of services to the clients, driven by economies of scale and the service are dynamically configurable.

As a very new technology for enterprises there are many benefits stated of cloud computing by different researchers which make it more preferable to be adopted by enterprises. Cloud computing infrastructure allows to achieve more efficient use of their IT hardware and software investments. This is achieved by breaking down the physical barrier inherent in isolated systems, automating the management of the group of the systems as an entity. Cloud computing can also be described as the ultimately virtualized system and a natural evolution for data centers which offer automated systems management. The rest of the paper is organized as follows: Section II introduces previous work. Section III introduces the heuristic based scheduling, Section IV introduces Genetic Algorithm. Section V describes about the new Improved Genetic Algorithm which is coupled with Suffrage Heuristic. Section VI having the simulation and result. Finally, the paper was concluded in section VII.

RELATED WORK

Scheduling of tasks is a critical issue in Cloud Computing, and has received a lot of attention in recent years. Various researches have been done in this area. The basic ideas about scheduling in Cloud Computing and scheduling techniques are discussed in [2]. About the suffrage Heuristic discussed in [4]. The Scheduling using Genetic Algorithm and other modified versions of Genetic Algorithms are discussed in [7] up to [9]. We have discussed in this paper scheduling techniques, which are Min-Min, Suffrage and Genetic Algorithm.

HEURISTIC BASED SCHEDULING

Min-Min

Min-Min begins with the set MT (Meta Task) of all unassigned tasks and has two stages. In the first stage, the set of minimum expected completion time for each task in MT is found. In the second stage, the task with the overall minimum expected completion time from MT is chosen and assigned to the corresponding machine. Then this task is removed from MT and the process is repeated until all tasks in the MT are mapped. However, the Min-Min Algorithm first finishes the shorter tasks and then executes the long task.

Suffrage Heuristic

The Suffrage heuristic is based on the idea that better mappings can be generated by assigning a machine or resource to a task that would “suffer” most in terms of expected completion time if that particular machine is not assigned to it. Let the suffrage value of a task or job t_i be the difference between its second earliest completion time and its earliest completion time.

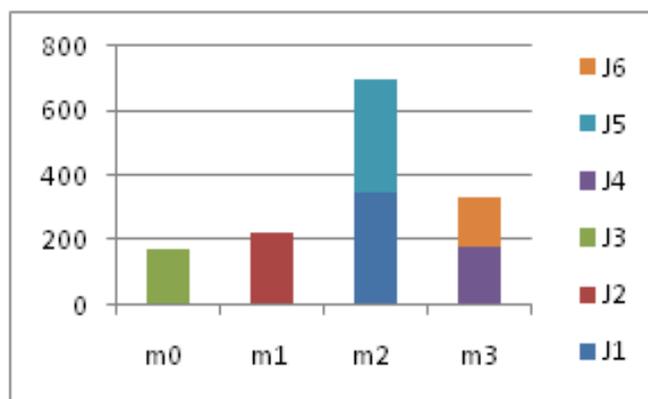


Figure 1: Task Assign under Min-Min Heuristic

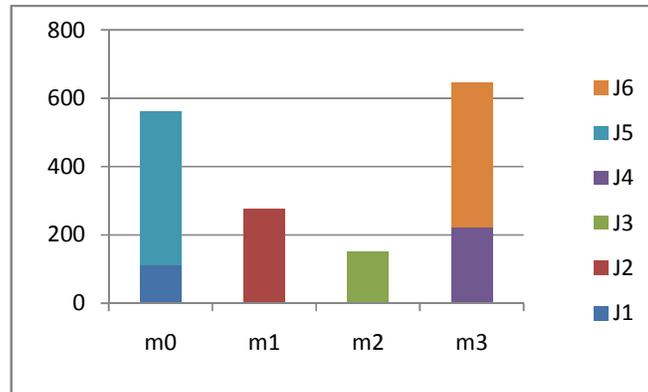


Figure 2: Task Assign under Suffrage Heuristic

Algorithm1: Suffrage Heuristic

Begin Main

Repeat

For each $j \in J_u$ **do**

Find the resource set that achieve the MCT for j

Find the second best completion time for j

Suffrage value = second best value – best value

End

Find the job $j \in J_u$ with the maximum suffrage value

Assign j to its selected resource set and remove j from J_u

Update the resource availability based on the allocation performed in the previous step

Until J_u is empty

End Main

GENETIC ALGORITHM

Genetic algorithm is based on biological concept of generation of the population, a rapid growing area of Artificial intelligence. GA's are inspired by Darwin's theory about Evolution. According to the Darwin "Survival of the fittest". It also a used as the method of scheduling in which the tasks are assigned resources according schedules in context of scheduling, which tells about which resource is to be assigned to which task. Genetic Algorithm is based on the biological concept of population generation [11].

Initial Population

Initial population is the set of all the individuals that are used in the Genetic Algorithm to find out the optimal solution. Every solution in the population is called as an individual. And every individual is represented as a chromosome for making it suitable for the genetic operations. From the initial population the individuals are selected and some operations are applied on those to form the next generation. The mating chromosomes are selected based on some specific criteria.

Fitness Function

The productivity of any individual depends on the fitness value. It is the measure of the superiority of an individual in the population. The fitness value shows the performance of an individual in the population. If the large fitness value, then the performance of an individual is better. Depending on the fitness or function value, whether the individuals are survive or die out. Hence, the fitness function is the motivating factor in the Genetic Algorithm.

Selection

Selection mechanism is used to select an intermediate solution for the next generation based on the survival of the Darwin law. This operation is the guiding channel for the Genetic Algorithm based on the performance. There are various selection strategies to select the best chromosomes e.g. roulette wheel, Boltzmann strategy, tournament selection, selection based on rank, etc.

Crossover

Crossover/hybridizing operation can be achieved by selecting two parent individuals and then creating a new individual tree by alternating and reforming the parts of those parents. Hybridization operation is a guiding process in the genetic algorithm and it boosts the searching mechanism.

Mutation

After crossover Mutation takes place. It is the genetic operator that introduces genetic diversity in the population. Mutation takes place whenever the population tends to become homogeneous due to repeated use of reproduction and crossover operator. It occurs during evolution according to a user-defined mutation probability, usually set to fairly low. Mutation alters one or more gene values in chromosome from its initial state. This can produce the entirely new gene values being added to the gene pool. With this new gene values, the genetic algorithm may be able to produce the better solution than was previously.

Algorithm 2: Genetic Algorithm

Begin Main

[**Initialize**] Produce random population of n chromosomes

[**Fitness**] In the given population, calculate the fitness value $f(x)$ of every chromosome.

[**Selection**] According to the fitness value, select two parent individuals from the population.

[**Crossover**] Generation of the new offspring by reforming the parents by using the crossover probability.

[**Mutation**] Mutate the new child at some positions with the probability of mutation.

[**Accepting**] now the new offspring the part of next generation of population.

[**Replace**] use the new generation as the current generation.

End Main

GENETIC ALGORITHM COUPLED WITH SUFFRAGE HEURISTIC

As discussed in Genetic Algorithm survival of fittest, it means the population that are fit give better generation and if we apply suffrage heuristic for the individual generation, which will be helpful for the better initial population and improved solution as compared to generic Genetic Algorithm.

Algorithm 3: Genetic Algorithm Coupled with Suffrage Heuristic**Begin Main**

Find out the solution by Suffrage Heuristic

[**Initialize**] Produce random population of n chromosomes by using the result of step 1.

[**Fitness**] In the given population, calculate the fitness value $f(x)$ of every chromosome.

[**Selection**] According to the fitness value, select two parent individuals from the population.

[**Crossover**] Generation of the new offspring by reforming the parents by using the crossover probability.

[**Mutation**] Mutate the new child at some positions with the probability of mutation.

[**Accepting**] now the new offspring the part of next generation of population.

[**Replace**] use the new generation as the current generation.

End Main**SIMULATION AND RESULTS**

Cloud Sim leaded by Buyya, allows cloud customers to test their services in repeatable and controllable environment free of cost, and to turn the performance bottlenecks before deploying on real clouds. It can provide a generalized and extensible simulation framework that enables modeling, simulation and experimentation of emerging cloud computing infrastructures and application services. It is designed for studying various resource management approaches and scheduling algorithms in cloud environment. The Cloud Sim toolkit supports both system and behavior modeling of Cloud system components such as data centers, virtual machines (VMs) and provisioning policies of resource. It implements generic application provisioning techniques that can be extended with ease and limited efforts In Cloud Sim, users is modeled by a Datacenter Broker, which is responsible for mediating between users and service providers depending on users' QoS (Quality of Service) requirements and deploys service tasks across Clouds. In our experiments, we have used Cloud Sim as a simulator for checking the performance of our improved algorithm and the Generic Genetic Algorithm. Cloud Sim is an extensible simulation toolkit that enables modeling and simulation of Cloud computing systems and application provisioning environments. We have considered Virtual Machines as resource and Cloudlets as tasks/jobs. We have checked the performance of the algorithm by fixed the number of virtual machines and varied the number of cloudlets. The makes pans that the algorithms produce are shown in figure 3. We have fixed the number of virtual machines as 20 and we are varying the number of cloudlets from 20 to 100 with a difference of 20.

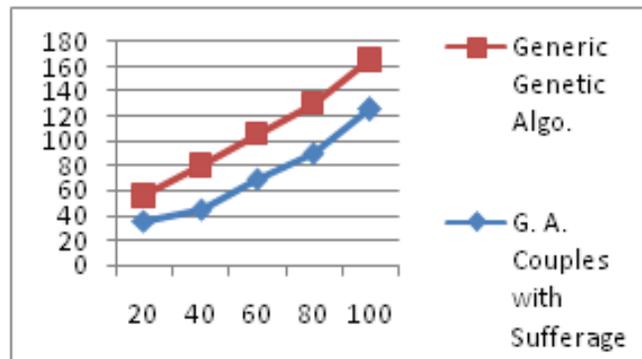


Figure 3: Graph for Make Spans

CONCLUSIONS

We have designed and tested an algorithm which is made by Genetic Algorithm coupled with suffrage Heuristic. The main goal of it, to schedule multiple jobs on multiple machines in an efficient manner such that the jobs take the minimum time for the completion.

REFERENCES

1. Foster, Y Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-degree compared[C]", in *Grid Computing Environments Workshop*, 2008, pp. 1-10.
2. Huang Q.Y., Huang T.L., "An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 673-675, 2010.
3. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", *Technical Report No. UCB/EECS-2009-28*, 2009.
4. E. U. Munir, J. Li and S. Shi, "QoS Sufferage Heuristic for Independent Task Scheduling in Grid," *Information Technology Journal*, Vol. 6, No. 8, pp. 1166-1170, 2007.
5. M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen and R. F. Freund, "Dynamic Matching and Scheduling of a Class of Independent tasks onto Heterogeneous Computing Systems," *Journal of Parallel and Distributed Computing*, Vol. 59, No. 2, pp. 107-131, 1999.
6. E. Munir, J. Li, S. Shi and Q. Rasool, "Performance Analysis of Task Scheduling Heuristics in Grid", in *Proc. 6th Inter. Conf. on Machine Learning and Cybernetics*, 2007, pp. 3093-3098.
7. Yin H., Wu H., Zhou J., "An Improved Genetic Algorithm with Limited Iteration for Grid Scheduling", IEEE Sixth International Conference on Grid and Cooperative Computing, 2007. GCC 2007, Los Alamitos, CA, pp. 221-227, 2007.
8. Guo G., Ting-Iei H., Shuai G., "Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment", IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, Guilin, pp. 60-63, 2010.
9. Wang J., Duan Q., Jiang Y., Zhu X., "A New Algorithm for Grid Independent Task Schedule : Genetic Simulated Annealing", IEEE World Automation Congress (WAC), 2010, Kobe, pp. 165-171, 2010
10. H. Baghban and M. Rahmani, "A Heuristic on Job Scheduling in Grid Computing Environment", in *Proc. 7th Inter. Conf. on Grid and Cooperative Computing (GCC'08)*, 2008, pp. 141-146.
11. Rajkumar Buyya, Rajiv Ranjan, Rodrigo N. Calheiros, "Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities", in *The 2009 International Conference on High Performance Computing and Simulation, HPCS2009*, pp:1-11.