ORIENTAL JOURNAL OF
COMPUTER SCIENCE & TECHNOLOGY
An International Open Free Access, Peer Reviewed Research Journal
Published By: **Oriental Scientific Publishing Co., India.**
www.computerscijournal.org

ISSN: 0974-6471
March 2017,
Vol. 10, No. (1):
Pgs. 60-65

Workload Consolidation using Task Scheduling Strategy Based on Genetic Algorithm in Cloud Computing

RONAK VIHOL[1]*, DR. HIREN PATEL[2] and PROF. NIMISHA PATEL[1,3]

[1]Sankalchand Patel College of Engineering Visnagar-Gujarat, India.
[2]LDRP Institute of Technology & Research, Gandhinagar-Gujarat, India.
[3]Rai University, Ahmedabad-Gujarat, India.
*Corresponding author E-mail: ronak.vihol1308@gmail.com

http://dx.doi.org/10.13005/ojcst/10.01.08

(Received: February 02, 2017; Accepted: February 15, 2017)

ABSTRACT

Offering “Computing as a utility” on pay per use plan, Cloud computing has emerged as a technology of ease and flexibility for thousands of users over last few years. Distribution of dynamic workload among available servers and efficient utilization of existing resources in datacenter is one of the major concerns in Cloud computing. The load balancing issue needs to take into consideration the utilization of servers, i.e. the resultant utilization should not exceed the preset upper limits to avoid service level agreement (SLA) violation and should not fall beneath stipulated lower limits to avoid keeping some servers in active use. Scheduling of workload is regarded as an optimization problem that considers many varying criterion such as dynamic environment, priority of incoming applications, their deadlines etc. to improve resource utilization and overall performance of Cloud computing. In this work, a Genetic Algorithm (GA) based novel load balancing mechanism is proposed. Though not done in this work, in future, we aim to compare performance of proposed algorithms with existing mechanisms such as first come first serve (FCFS), Round Robin (RR) and other search algorithms through simulations.

Keyword: Cloud computing, Genetic Algorithm, Load Balancing, Task Scheduling.

INTRODUCTION

As information and communication technology (ICT) grows gradually, need for computational resources have also increased than before. Investing in huge amount of IT assets may not be a good alternative for many industries. Answer to this issue is Cloud computing where computing resources are rented rather than owned. Combining the features of grid, cluster and utility computing mingled with virtualization, Cloud computing offers an innovative model of “computing as a utility” wherein computing resources such as processing elements, memory, storage, bandwidth etc. are provisioned on demand over the Internet on pay-per-use basis with elasticity and flexibility. [NIST [1]] The concepts has been so popular over recent years that most of the IT giants including Microsoft, Google, Apple, IBM etc. are not exception to the list of Cloud service providers. Due to enormous acquaintance of the concept, Cloud datacenter are deployed in gigantic number across the globe. Concerns such as security and energy consumption have been the prime motivations for many researchers for the domain of Cloud computing. The issue of energy consumption can be addressed in many directions out of which proper distribution of workload across available resources is one of them.

Workload allocation among available servers can be addressed through efficient load balancing techniques. There are many available load-balancing techniques in existence. For instance Google uses Map-reduce scheduling [Dean *et al.*[2]], Amazon EC2 and Eucalyptus employ simple mechanism like First Fit or First in First out (FIFO). Proficient load balancing scheme improves resource utilization and convene users' requirement in form of service level agreement (SLA). Genetic Algorithm (GA) [Goldberg *et al.* [4]] is a robust heuristic algorithm offering combinatorial optimization. In this paper, we propose a novel load balancing mechanism based on GA to discover the best possible task scheduling sequence in dynamic Cloud environment.

The rest of the paper is organized as follows. Section 2 discusses the related work that has been carried out in the domain over recent years. Section 3 is about our proposal based on GA. We conclude our work in section 4 subsequently listing the references used in section 5.

Related Work

Before we discuss about the work carried out in the area of task scheduling or workload consolidation in Cloud, we first discuss in brief regarding Genetic Algorithm (GA). GA imitates the principles of natural genetics and natural selection (survival of the fittest) to compose search and optimization processes. Deb *et al.* [3] The objective problem can be classified among two categories viz. maximization and minimization. Fitness function is derived from objective function to be used in successive genetic operations. GA starts with a (initial) population of random strings corresponding to design or decision variables. Further, fitness value is computed for each string. The resultant population is then operated by three main operator's viz. (i) reproduction/selection (ii) crossover (iii) mutation. The outcome of these operators will convert the original population to new population. These new population is further evaluated and tested for terminating condition. The procedure of iteratively operating the population is continued till the terminating condition is satisfied or the iterations are exhausted. The outcome of the process would be the best possible optimum solution for given design or decision variables. Reproduction/selection is generally applied on a population to select good strings from it and to form a mating pool. The elementary idea is to select the strings from population which are above-average (in terms of fitness) and to place them in a mating pool in a probabilistic manner. Roulette-wheel is a simplest way to implement selection scheme. In the crossover operator, new strings are created by exchanging information among strings in the mating pool. Two (parent) strings are selected from the mating pool randomly and some portions of the strings are exchanged using (i) single point or (ii) multi-point crossover to generate offspring's (children strings). Mutation operator changes 1 to 0 and vice-versa in a given string with a small mutation probability. Summarizing, the reproduction operator selects good strings and crossover operator recombines good substrings from good strings together to expectantly create a healthier substring. The mutation operator alters a string locally to hopefully create a better string.

Task scheduling in Cloud can be considered as multi-objective constrained optimization problem. Objectives of task scheduling are to minimize the waiting time period, improve the throughput and enhance the resource utilization. Wang et al.[4] define (a) minimizing job spanning (completion time) and (b) balancing workload as optimization objectives. They further propose three-tier task allocation architecture consisting of task request, resource management and task execution. They also put forward the equations for job's execution time, average execution time, task spanning, total time consumption, under task allocation model. Underload expression, equations for total and average time of a particular node are proposed and using which variance is computed. Subsequently, Job spanning time and load balancing genetic algorithm (JLGA) is proposed which takes into account number of factors such as maximum number of iterations, population scale, number of jobs, weights of total/average job spanning and probability of fitness functions. Authors use greedy initialization of initialization of population. Two fitness functions are proposed using which selection probabilities are computed. Using MATLAB simulation, authors claim improvement in total/average job consumption and balance in system. As part of future research direction, authors suggest to take job priority into account. In addition, authors also find a scope in considering dynamic global adaptive control strategy using GA to overcome the overhead caused due to static factors as population scale, iterations, crossover and mutation operators.

Dasgupta *et al.*[5] propose a load balancing strategy using GA to minimize the make span of given task and compares the results of the strategy with existing approaches such as First Come First Serve (FCFC), Round Robin (RR) and a local search algorithm Stochastic Hill Climbing (SHC) using Cloud Analyst simulator. Authors formulate the problem using two factors viz. (i) jobs and (ii) processing units. Jobs can be defined by the attributes such as type of service required by the job (SaaS, PaaS, IaaS etc), number of instructions present in the job (i.e. length of the job), job's arrival time and worst case completion time of the job. While, processing units are categorized by the factors such as number of instructions that can be executed by the unit per second (MIPS), cost of execution of instruction and delay cost. Further, the fitness function in form of cost function (which is to be minimized) is defined with number of instructions present in job, number of instructions executed by processing element per second and delay cost. Experimentation is carried out with varying number of data centers (1 to 6). Using simulation results, authors claim that the load balancing strategy using GA is efficient compared to other existing techniques mentioned above. Authors have assumed that all the jobs have same priority but they suggest the factor to be considered as future work.

Kumar *et al.*[6] study two scheduling techniques viz. Min-Min and Max-Min. They further explore a standard genetic algorithm and propose an improvement in it. The improved algorithm is tested using the CloudSim simulator. Using empirical results authors claim minimization in make span of a task and improvement in resource utilization.

Hamad *et al.*[7] introduce a GA based task scheduling algorithm for job's allocation and execution. Authors intend to minimize the job's completion time and cost and to improve resource utilization. Tournament Selection GA (TS-GA) has been proposed. Normally (in default GA), solution which is generated after each selection in the population which may satisfy good fitness function may be removed and it is not selected to crossover function. The idea of TS-GA is to retain this solution for the next iteration. Using CloudSim simulation, experimentation results of TS-GA are compared with RR and default GA considering parameters such as job's completion time, cost, resource utilization, speedup and efficiency. The results show improvement in all these parameters using TS-GA. As part future research direction, authors propose the inclusion of dynamic characteristics of VM and more parameters compared to what considered in the paper.

Liu *et al.*[8] design multi-objective GA (MO-GA) focusing on encoding rules, crossover & selection operators and the method of sorting Pareto solutions to minimize energy consumption and to maximize the profit of service. A functional architecture of job scheduling model has been proposed. Every application (task) is submitted to Cloud with three parameters (i) application's reservation time for VMs (ii) number of VMs required for the application and (iii) deadline after which the application will be considered as fail. The experimentation results depict the improvement in energy consumption, profit and number of failed applications.

Mingxin *et al.*[9] propose dual fitness genetic algorithm (DFGA) to reduce total (task scheduling) completion time and the (scheduling task) average completion time. Using Grid Sim simulator, the results are compared with adaptive genetic algorithm and claimed to be improvement in task scheduling.

METHODS

An effective approach to task scheduling is a long-standing challenge in Cloud. Many conventional approaches are used for scheduling tasks. However, to generate and identify more optimal solutions, we recommend using the genetic algorithm for scheduling tasks. We proposed a new GA method based on weight. First the task and the resources are arranged according to the priority and the weightage respectively.

Algorithm: Weight Based GA
Input: List Task, List Resource
Output: Mapping of Task with Resource

```
for each i in resource R do
$\text{Memory utilization (i)} = \frac{\text{Total memory (i)} - \text{Used memory(i)}}{\text{Total memory (i)}}$

$\text{CPU utilization (i)} = \frac{\text{Total CPU usage of process (i)}}{\text{no. of process (i)}}$

$\text{CPU core (i)} = \frac{\text{Total core (i)} - \text{Used core(i)}}{\text{Total core (i)}}$

$\text{Weight}(W_{R_i}) = \frac{\text{Memory utilization (i)} + \text{CPU utilization (i)} + \text{CPU core (i)}}{\text{Number of resource} * 100}$

Attach $W_{Ri}$ to Resource $R_i$
end for
Sort list Resource R in descending order based on weight $W_R$
    for each i in list Resource R do
        Eff(i)=calEfficiency(i)
    end for
        $\text{avgEff} = \sum_{i=1}^{R} \frac{\text{Eff(i)}}{R}$

for each i in list Resource do
    if avgEff<Eff(i)then
        next Resource list.add(i)
    end if
end for
perform crossover on next Resource list and list Task
perform mutation on next Resource list and listTask

function calEfficiency(Resource r)
        for each task t on resource r do
Task Makespan(t)+= End Time(t)-Start Time(t)
end for
$\text{Node Makespan}(M_N) = \frac{\text{TaskMakespan(t)}}{\text{number of core(c)}}$
Node Efficiency(r)= Node Makespan($M_N$)
                    * Weight (r)

return Node Efficiency(r)
end function
```

where,
R=no.of Resources
W_{Ri})=Weight of Resources
M_{Ti})=Makespan of task i
M_N=Makespan of node
N_i=Number of core of N_i

In our weight based GA, we first calculate weight for each resource. For calculating weight for resource, we compute memory utilization, CPU utilization and number of cores for every resource. The calculated weight is attached to the resource. Then, all the resources are sorted in descending order based on weight. For every resource, we find efficiency which is a function of makespan and weight. If the efficiency is less than average efficiency then we add this resource to next generated list, which is to be passed for next round of genetic algorithm, for further optimization.

Following figure 1 depicts the overall process.

In the flowchart, the first step is to initialize input data likely list of task and list of resources. After that, Task and resources are queued according to the priority and weightage. A selection of the individually according to the job mapping condition and apply crossover, mutation operator's. Finally, we get the optimal solution to the task of suitable resource.

The proposed algorithm is only theoretically tested and validated. As part of next phase future

Flowchart

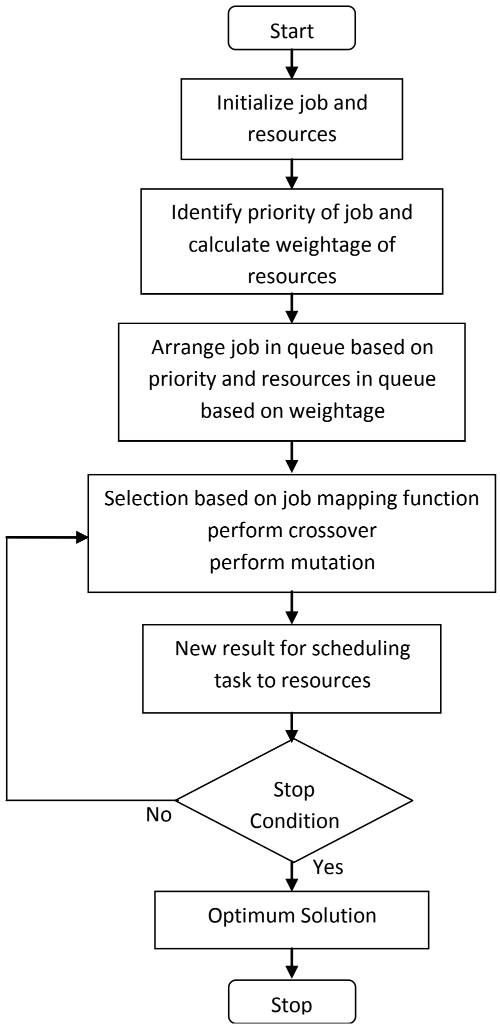
Start
Initialize job and resources
Identify priority of job and calculate weightage of resources
Arrange job in queue based on priority and resources in queue based on weightage
Selection based on job mapping function perform crossover perform mutation
New result for scheduling task to resources
Stop Condition
No
Yes
Optimum Solution
Stop

Fig.1: Flowchart of proposed algorithm

work, we plan to implement of the proposed mechanism in the CloudSim[11] environment using the real-world data set of Planetlab[12]. We aim to consolidate the task scheduling process based on our proposed scheduling algorithm based on GA. The outcome of experimentation will be compared with that of contemporary policies.

CONCLUSION

Task scheduling and efficient resource utilization are few of the main problems in the Cloud computing environment. To address the issues, we use GA for task scheduling problem as well as utilization of resources. We use the fitness function based on a weight of resources and the selection of a higher efficiency of the individual nodes to base ourselves on the mean values of efficiency to obtain the optimized result of the planning tasks. As part of future work, we aim to implement the proposed work on simulated environment and compare the result of the work with existing mechanisms.

REFERENCES

1. P. Mell and T. Grance, “The NIST definition of Cloud Computing”, National Institute of Standard and Technology, Information Technology Laboratory 800-145, 2011
2. Y. Ge and G. Wei, “GA-Based Task Scheduler for the Cloud Computing Systems,” 2010 International Conference on Web Information Systems and Mining, Sanya, 2010, pp. 181-186. doi: 10.1109/WISM.2010.87
3. Kalyanmoy Deb, “Optimizatio for engineering design algorithm and example”
4. Tingting Wang, Zhaobin Liu, Yi Chen, Yujie Xu, Xiaoming Dai, “Load Balancing Task Scheduling Based on Genetic Algorithm in Cloud Computing”, IEEE 2014.
5. Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, “A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing”, First International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA’13).

SPRINGER 2013.

6. Pardeep Kumar and Amandeep Verma, "Scheduling using improved genetic algorithm in Cloud computing for independent tasks" In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (ICACCI '12). ACM, New York, NY, USA, 137-142, 2012.
7. Safwat A. Hamad and Fatma A. Omara, "Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment" International Journal of Advanced Computer Science and Applications (IJACSA), **7**(4), 2016.
8. Jing Liu, Xing-Guo Luo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li, "Job Scheduling Model for Cloud Computing Based on Multi-Objective Genetic Algorithm", *International Journal of Computer Science Issues* (IJCSI) 2013.
9. Wu Mingxin "Research on Improvement of Task Scheduling Algorithm in Cloud Computing", International Journal of Applied Mathematics & Information Sciences **9**, 1, 507-516 (2015), NSP 2015.
10. Rajveer Kaur, Supriya Kinger "Enhanced Genetic Algorithm based Task Scheduling in Cloud computing" *International Journal of computer Application* 2014
11. Calheiros RN, Ranjan R, Beloglazov A, Rose CAFD, Buyya R. CloudSim: a toolkit for modeling and simulation of Cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience 2011; **41**(1):23–50
12. Park K.S., Pai V.S. CoMon: A mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Oper. Syst. Rev. 2006;40:65–74. doi: 10.1145/1113361.1113374