IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 3, Ver. V (May-Jun. 2016), PP 23-27
www.iosrjournals.org

Workflow Scheduling for Public Cloud Using Genetic Algorithm (WSGA)

Dr. D. I. George Amalarethinam[1], T. Lucia Agnes Beena[2]
[1](*Dean of Science & Director MCA, Department of Computer Science, Jamal Mohamed College, Trichy, India)*
[2](*Assistant Professor, Department of Information Technology, St. Joseph's College, Trichy, India)*

Abstract : *Workflow scheduling is a challenging issue in Cloud Computing. Though there are popular schedulers available for workflow scheduling in Grid and other distributed environments, they are not applicable to Cloud. Cloud differs from other distributed environments in resource pool and incurs less failure rate.Workflow scheduling in Cloud has to concentrate on the QoS parameters such as deadline and budget. Most heuristic algorithms are proposed in the literature. But the meta-heuristic algorithm like Genetic Algorithm approach for the workflow scheduling in Cloud is expected to yield optimal results. This paper is an attempt to minimize the execution cost of the workflow using the Genetic Algorithm. The new fitness function is proposed to minimize the cost and the selection, crossover, mutation operators are applied with the arbitrary task graphs given as input. It was observed that the proposed algorithm reduces the cost to the optimal value when compared to the other list heuristic algorithms like HEFT, CFCSC and LBTP for communication intensive graphs.*
Keywords: *Crossover, Fitness function, Genetic algorithm, Mutation, Selection, Workflow Scheduling*

I. Introduction

Cloud computing has become a standardized way of providing IT services delivered through Internet technologies in a pay-per-use and in a self-service way. Cost reduction and Organizational Agility have attracted industries to adopt Cloud Computing. The different types of Clouds provide services for various categories of users. The flexibility of Cloud Services support any individual or organization to go for Cloud Computing. One of the important areas of Cloud application is Workflow scheduling.

Workflow scheduling tries to map the workflow tasks to the Virtual Machines (VMs) based on different functional and non-functional requirements [1]. A workflow consists of a series of interdependent tasks which are bounded together through data or functional dependencies and these dependencies should be considered in the scheduling. However, workflow scheduling in the cloud computing is an NP-hard optimization problem [1] and it is difficult to achieve an optimal schedule. As there are numerous VMs in a cloud, many user tasks are to be scheduled by considering various scheduling objectives and factors. The common objective of the workflow scheduling techniques is to minimize the makespan by properly allocating the tasks to the virtual resources [2].

For scheduling problems, no known algorithm is able to generate an optimal solution within the polynomial time. Thus there is a need to apply stochastic optimization methods to solve the scheduling problem by using random variables. These methods include simulated annealing, swarm algorithms and evolutionary algorithms. The main advantage of genetic algorithms over other traditional optimization methods is that genetic algorithms are parallelizable. Genetic algorithms intrinsically work with many solutions in parallel which enables them to explore the solution space in multiple directions at any time thereby converging faster [3]. This paper attempts to apply Genetic algorithm for the workflow scheduling to optimize the cost and the performance.

II. Motivation

Researchers prefer the evolutionary algorithms to optimize the performance of the workflows rather than applying the heuristic scheduling algorithms and also Cloud Computing task scheduling differ from the other distributed environments viz. Grid computing in resource sharing and cost of resource utilization [4]. Workflow based scheduling is able to efficiently determine an optimal solution for large and complex applications by considering precedence constraints between potential tasks. One of the most challenging problems with workflow scheduling in cloud computing is to optimize the cost of workflow execution. Sindhu et. Al. [5] proposed a bi-objective Genetic Algorithm based scheduler for cloud that optimizes the makespan (application-centric) and average processor utilization (resource-centric). QoS based task scheduling using genetic algorithm for independent tasks was proposed by Jang et. al. [6]. Kaur et. al. [7] proposed a meta-heuristic based scheduling, which minimizes execution time of the independent tasks at heavy loads. Sourav et. al. [8] produces one scheduling algorithm based on Genetic Algorithm to optimize the waiting time of overall system. Ge and Yuan [9] presented a genetic algorithm MGA, optimized the total task completion time, average task completion time with required costs.

Agarwal et. al. [10] presented a Generalized Priority algorithm for efficient execution of task and compared the algorithm with FCFS and Round Robin Scheduling. This Algorithm was tested in CloudSim toolkit and result showed better performance compared to other traditional scheduling algorithm. Various scheduling algorithms [11][12][13][14][15][16] were proposed for distributed environment with execution time, speed up and efficiency. Kaur et. al. [17] proposed a meta-heuristic based scheduling, which minimizes execution time and execution cost for independent tasks. A cloud user reaches a Service Level Agreement (SLA) with a cloud provider to process a task. A SLA document includes user requirements like time and budgetary constraints of the task, which indicate acceptable deadline and payable budget of the cloud user [18]. From the literature, it is found that very few algorithms were proposed for workflow scheduling in Cloud. Some Grid workflow management systems, like Pegasus [19] started supporting execution of workflows on Cloud platforms. But it uses Heterogeneous Earliest Finish Time (HEFT) algorithm as the scheduling algorithm which doesn't include cost parameter. Juve et al. [20] found that Cloud is much easier to set up and use, more predictable, capable of giving more uniform performance and incurring less failure than Grid. This background motivates to propose a genetic algorithm based workflow scheduling for the Cloud, which optimizes the cost of executing the workflow in the Cloud.

III. The Proposed Work (WSGA)

Scientific applications are usually represented by workflows. The workflows depict the number of tasks and the data dependencies between the tasks of an application. It is advantageous to use Cloud to execute the complex scientific applications due to its large-size resource pools. One of the most challenging problems in Workflow Scheduling is to optimize the cost of workflow execution. The meta-heuristic scheduling schemes yield the best result when compared to the heuristic algorithms. One of the best meta-heuristic algorithms is Genetic algorithm. A Genetic Algorithm (GA) is a search algorithm which is based on the principle of evolution and natural genetics. It combines the exploitation of past results with the exploration of new areas of the search space [21]. By using the survival of the fittest techniques combined with a randomized information exchange, the best solution is obtained. The experimental setup for this proposed algorithm WSGA was tabulated in Table 1. Figure 1 gives the steps in GA.

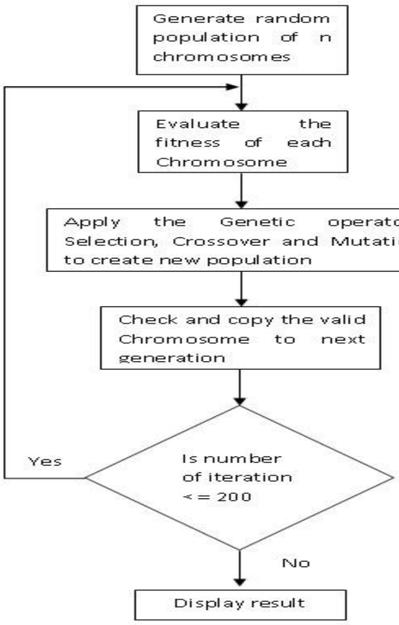

Fig.1: Steps in Genetic Algorithm

1. Chromosome Representation (WSGA)

A GA for a scheduling problem must have the following five components:

- Genetic representation for potential solutions to the problem
- A way to create an initial population of the potential solutions
- An evaluation function that plays the role of the environment, rating solutions in terms of their "fitness"
- Genetic operators that alter the composition of children.

- Termination condition

In workflow scheduling, there are set of tasks to be scheduled. These tasks have to obey the following rules[21]. They are:

- A task's predecessors must have finished their execution before it can start executing
- All tasks within the workflow must execute at least once.

Chromosomes are most commonly represented in binary alphabets {0,1}. The other representations include ternary, integer and real values [21]. Based on these fundamentals, the chromosome for the workflow scheduling is represented as an integer. The scheduling solution is a sequence of set of tasks to be scheduled. Each task is represented by a pair of integers comprising the task and the resource in which the task is executed. For example, (T7, 4) represents the task T7 to be executed on Resource 4. Thus a chromosome consists of n tasks to be executed on m processors.

Table 1: Experimental setup

Population size	20
Selection method	Roulette Wheel
Crossover method	Single Point Crossover
Crossover rate	0.7
Mutation rate	0.1
No. of Iterations	200

2. Initial Population of WSGA

The initial population size is fixed as 20 and is built by generating chromosomes with the list –based heuristic algorithms such as HEFT [22], CFCSC [23] and LBTP [24]. The remaining chromosomes are generated randomly. The initial population is checked for its validity through the fitness function.

3. Fitness function of WSGA

The objective of the fitness function is to evaluate each chromosome in the population. In case of minimization problem, the best fit chromosome will have the lowest numeric value for the objective function. Based on the fitness value the chromosome may be selected for the next generation in the solution set. The objective of WSGA is to minimize the cost of executing the workflow in the Cloud.In GA, the minimization problem should be converted to maximization without sacrificing the optimal solution. Thus the fitness function *f(i)* for the WSGA is given by

$$f(i) = \frac{1}{1+(MS*AVGCOST)} \qquad (1)$$

where MS is the makespan, AVGCOST is the average resource cost of the chromosome i. The fitness value for all the chromosomes in the population is calculated using equation (1).

4. Selection operation of WSGA

The objective of the selection operation is to make duplicate copies of the good solution and eliminate bad solutions in a population, while maintaining the population size. In order to identify the good solution, in this paper, the Roulette wheel method is used. The roulette wheel selection operator maximizes the fitness function. Also Stochastic remainder Roulette-Wheel Selection (SRWS) reduces the variance [25]. In SRWS operator, each solution is first assigned a number of copies equal to the integral part of the expected number. Thereafter, the usual roulette wheel selection operator is applied with the fractional part of the expected number of all solutions to assign further copies.

5. Crossover Operation of WSGA

A crossover operation is applied next to create new solution from the chromosomes of the mating pool. There are number of crossover operations available in the GA literature. In the proposed WSGA, single-point crossover operation is applied. This is performed by randomly choosing a crossing site along the chromosome and by exchanging all the pairs on the right side of the crossing site. In order to preserve some good chromosomes in the population, a crossover probability p_c has to be defined. In WSGA, the crossover probability was 0.7% of the population size. The crossover probability was varied from 0.5 to 0.95, in steps of 0.2 and it was found that 0.7% providing the optimal result. In this paper, the population size is fixed as 20 and 14 chromosomes were selected in each iteration. In selecting the 14 chromosomes for crossover, each chromosome's cost is checked against the average cost. If the chromosome's cost is less than the average cost, that chromosome is selected for crossover operation. This is followed by the mutation operation.

6. Mutation Operation of WSGA

The need for mutation operation is to keep diversity in the population. The mutation probability p_m for the proposed WSGA is 0.1%. Two chromosomes undergo mutation in each generation. The chromosomes for mutation are selected randomly. In each chromosome, a random task is selected and its corresponding resource is altered so that it may lead to lower cost.

7. The Termination of WSGA

Usually a GA is terminated after a certain number of generations or if a level of fitness has been obtained or a point in the search space has been reached [21]. In WSGA, the generations are varied from 50 to 300 in steps of 50 and it was found the optimal solution is attained in the 200^{th} generation.

IV. Results and Discussion

The proposed WSGA is developed in Java in the Netbeans IDE 7.1. The input for the WSGA is the arbitrary task graph generated by a program developed in Java [26]. This program generates the needed virtual machine instance with various speeds randomly. Given the number of tasks to be generated and the number of virtual machines, the program generates the arbitrary task graphs.

Table 2 : Makespan (Sec.)

No. of Tasks	No. of Resources	Algorithms			
		HEFT	***CFCSC***	***LBTP***	***WSGA***
10	3	48.85	39.85	34	**13**
20	4	74.25	57.9	62	**55**
50	7	98.53	101.67	98	**85**
100	10	107.82	115.8	130	**101**
150	12	132	129	128	**114**
200	14	142	126.75	147	**111**

Table 3. Cost ($)

No. of Tasks	No. of Resources	Algorithms			
		HEFT	***CFCSC***	***LBTP***	***WSGA***
10	3	7.7	2.2	2.07	**1.23**
20	4	8.25	5.09	4.74	**3.23**
50	7	14.07	6.46	4.62	**3.81**
100	10	47.72	19.9	22.1	**11.58**
150	12	74.8	31.29	29.45	**14.95**
200	14	96.99	46.59	46.64	**14.5**

From the initial population, after applying the reproduction operations like selection, crossover and mutation for 200 generations, the results were observed for makespan of the arbitrary task graph and the monetary cost for executing the task graph. The task graph size is varied from 10 to 200. Since Cloud follows *pay as you go* formula for the service, the virtual machine instance is charged based on the Google AppEngine [27]. The high speed CPU is costlier than the low speed CPU.

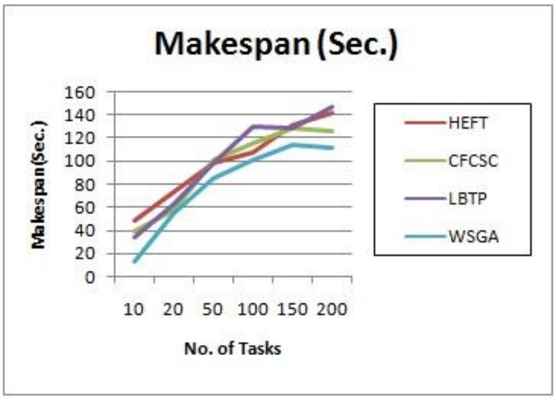

Fig 2: Graphical representation of Makespan

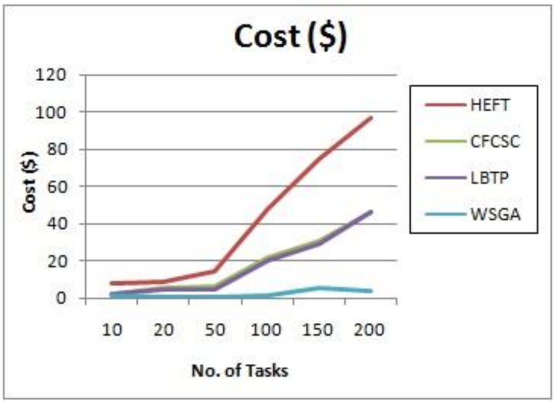

Fig 3: Graphical representation of Cost

The result of WSGA is compared with the heuristic algorithms like HEFT, CFCSC and LBTP. It was observed that cost was reduced to the optimal value. The results are tabulated in Table 2 and Table 3. The graphical representations of the results are shown in Figure 2 and Figure 3.

V. Conclusion

Using the Cloud environment, the scientists can take advantage of executing their workflows with lower cost. Though some workflow scheduling algorithms were proposed for the scientific applications in the distributed environment, they are not suitable for the Cloud environment. This paper, proposed a workflow scheduling algorithm applying the Genetic algorithm to minimize the cost of executing the workflow in the Cloud. The fitness function used in the proposed algorithm selects the appropriate chromosomes for the next generation. The probability of the crossover and the probability of mutation were decided after conducting various experiments with possible values. The termination condition is also finalized as 200 generations by repeating the experiments with different values. This results in cost optimization. Thus the proposed genetic algorithm outperforms the other list scheduling algorithms used in this paper. As a future work, this algorithm will be tested in the *Cloudsim* tool to observe the performance. As this algorithm is tested with arbitrary task graphs, real time workflows can be given as input to the algorithm and its results are expected to be consistent.

References

[1] Mohammad Masdari, SimaValiKardan, ZahraShahi, SonayImaniAzar, Towards workflow scheduling in cloud computing: A comprehensive analysis, *Journal of Network and Computer Applications*, Vol. 66, pp.64–82, 2016.

[2] Rahman M,etal. Adaptive workflow scheduling for dynamic grid and cloud computing environment. *Concurr Comput: Pract Exp2013*, Vol. 25, pp.1816–42, 2013.

[3] Weise, Thomas. "Global optimization algorithms-theory and application.", Self-Published, 2009.

[4] Alkhanak EN, Lee SP, Rezaei R, Parizi RM. Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues. *Journal of Systems and Software*, Vol.11, pp. 1-26, 2016.

[5] Sindhu, S., and Sayan Mukherjee. "A genetic algorithm based scheduler for cloud environment." In Computer and Communication Technology (ICCCT), 4th International Conference on IEEE, pp. 23-27, 2013.

[6] Jang, S.H., Kim, T.Y., Kim, J.K. and Lee, J.S.,. The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation,Vol 5, Issue 4, pp.157-162, 2012.

[7] Kaur, S. and V, erma, A., An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science (IJITCS), Vol. 4, Issue 10, pp.74, 2012.

[8] Banerjee, S., Adhikari, M. and Biswas, U., Advanced task scheduling for cloud service provider using genetic Algorithm. IOSR Journal of Engineering (IOSRJEN), Vol. 2, pp.141-147, 2012.

[9] Ge, J.W. and Yuan, Y.S., October. Research of cloud computing task scheduling algorithm based on improved genetic algorithm. In Applied Mechanics and Materials, Vol. 347, pp. 2426-2429, 2013.

[10] Agarwal, D. and Jain, S., Efficient optimal algorithm of task scheduling in cloud computing environment. preprint arXiv: 2014, 1404.2076, 2014.

[11] Rajveer Kaur and Supriya Kinger. Article: Enhanced Genetic Algorithm based Task Scheduling in Cloud Computing. International Journal of Computer Applications Vol. 101, pp. 1-6, (2014).

[12] Daoud, M.I. and Kharma, N., An efficient genetic algorithm for task scheduling in heterogeneous distributed computing systems. In Evolutionary Computation, CEC 2006. IEEE Congress, pp. 3258-3265, 2006.

[13] Bohler, M., Moore, F.W. and Pan, Y., 1999, May. Improved Multiprocessor Task Scheduling Using Genetic Algorithms. In FLAIRS Conference, pp. 140-146, 1999.

[14] Golub, M. and Kasapovic, S., January. Scheduling multiprocessor tasks with genetic algorithms. In APPLIED INFORMATICS-PROCEEDINGS- 3, pp. 273-278, 2002.

[15] D.I.George Amalarethinam F.Kurus Malai Selvi, Grid Scheduling Strategy using GA (GSSGA), Int.J.Computer Technology & Applications, Vol. 3, Issue 5, pp. 1800-1806, 2012.

[16] Sharma, A. and Kaur, M., An Efficient Task Scheduling of Multiprocessor Using Genetic Algorithm Based on Task Height. Journal of Information Technology & Software Engineering, Vol. 5, Issue 2, pp.1, 2015.

[17] Kaur, S. and Verma, A., An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science (IJITCS), Vol. 4, Issur 10, pp.74, 2012.

[18] Jang, S.H., Kim, T.Y., Kim, J.K. and Lee, J.S., The study of genetic algorithm-based task scheduling for cloud computing. International Journal of Control and Automation, Vol. 5, Issue 4, pp.157-162, 2012.

[19] Ramakrishnan, A., Singh, G., Zhao, H., Deelman, E., Sakellariou, R., Vahi, K., Blackburn, K., Meyers, D. and Samidi, M., Scheduling data-intensiveworkflows onto storage-constrained distributed resources. In Cluster Computing and the Grid, 2007. CCGRID 2007. Seventh IEEE International Symposium on IEEE, pp. 401-409, 2007.

[20] Juve, G., Rynge, M., Deelman, E., Vockler, J.S. and Berriman, G.B., Comparing futuregrid, amazon ec2, and open science grid for scientific workflows. *Computing in Science & Engineering*, *15*(4), pp.20-29, 2013.

[21] Zomaya, A.Y., Ward, C. and Macey, B., Genetic scheduling for parallel processor systems: comparative studies and performance issues. *Parallel and Distributed Systems, IEEE Transactions on*, Vol. 10, Issur 8, pp.795-812, 1999.

[22] Selvarani, S., and G. SudhaSadhasivam. Improved cost-based algorithm for task scheduling in cloud computing. In Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference, pp. 1-5, 2010.

[23] George Aamalarethinam D. I., Lucia Agnes Beena T. Customer Facilitated Cost-based Scheduling algorithm (CFCSC) in Cloud. Procedia Computer Science. Elsevier Publications 46C, pp. 660-667, April 2015.

[24] George Aamalarethinam D. I., Lucia Agnes Beena T. Level Based Task Prioritization Scheduling for Small Workflows in Cloud Environment, Indian Journal of Science and Technology, Vol. 8, Issue 33, pp 1–7, 2015.

[25] Deb, K., An introduction to genetic algorithms. *Sadhana*, *24*(4-5), pp.293-315, 1999.

[26] George Amalarethinam D.I, Joyce Mary G. J. DAGEN – A tool to generate arbitrary Directed Acyclic Graphs used for Multiprocessor Scheduling. *International Journal of Research and Reviews in Computer Science (IJRRCS)*, Vol. 2, Issue 3, pp. 782 – 787, 2011.

[27] Google AppEngine.Available from: URL: *https://cloud.google.com/compute/pricing/04.03.2016*