



Efficient Cloud Computing Scheduling: Comparing Classic Algorithms with Generic Algorithm

Msc. Masoumeh Alipori^{1,2} and PhD. HAMID HAJ SEYYED JAVADI³

¹ Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Boroujerd, Iran

² Department of Computer Engineering, Boroujerd Branch, Islamic Azad University, Boroujerd, Iran

³ Department of Applied Mathematics, Faculty of Mathematics and Computer Science, Shahed University, Tehran, Iran

E-mail: ^{1,2}alipori.m@gmail.com, ³hamid.h.s.javadi@gmail.com

ABSTRACT

One of the most important indexes of using cloud services is that this technology is far from the user. In cloud computing systems, computing resources are presented as virtual machines. In such a scenario, scheduling algorithm plays a very important role because the purpose of scheduling is tasks efficiency so that time is reduced, and resources utilization can be improved. A user may use hundreds of computing resources in a cloud environment, so it is not possible to perform scheduling manually. This can be done by using classic algorithms whose results have been studied and compared with our proposed algorithm, genetic algorithm. Selecting an appropriate and efficient algorithm for resources scheduling is required due to dynamic feature of resources and various requests of users in cloud technology to increase efficiency. In this research, our purpose is to perform and obtain an optimal scheduling by using genetic algorithm to reach the main purpose of finding an optimum scheduling to execute tasks graph in a multi-processor structure so that total execution time or ending time of the last work unit is minimized.

Keywords: *Cloud Computing, Scheduling Algorithm, Genetic Algorithm, Classic Algorithms.*

1 INTRODUCTION

Evolutionary route of computing is one of the essentials in today's life. In the case, the users of cloud environment try to access it on the basis of their needs. Also, they consider where the service is located or how it is delivered (Pearson, 2009). Cloud computing tries to provide services in network virtual machines dynamically so that users can access to their required application programs all over the world (Charbrashkhar et al,2012). Cloud technology prevents computing resources management remotely. Regardless of the used pieces and the user location, users can access to all services in cloud computing by using a browser (Pashpcosineh and Krit, 20012; Fournal et al, 2011). Selecting an appropriate and efficient algorithm is required to increase efficiency due to dynamic feature of resources and various requests of users in

cloud technology (Mafiami, 2009). In cloud computing, a dynamic and new generation is provided by presenting services in dynamic and networked virtual machines, and users can access to application programs all over the world (Bouya et al, 2009). Due to dynamic feature of resources and requests in cloud computing, allocation of resources to requests is considered as NP-complete problems. In recent years, one of the most important and promising methods used to solve such problems and to find optimal answer for resources scheduling is innovative methods derived from the nature. These methods are similar to natural or social systems. Some of these methods are genetic algorithm, chakoos optimization algorithm, and optimization of particles mass. Nature-oriented algorithm is a global optimization technique based on population. It is widely applied in most optimization fields due to its simple search

mechanism, computing efficiency and complexity and easy implementation (Houmar, 2012). Since immediate execution of processors is considered in cloud computing, a suitable scheduling algorithm can increase the speed and use maximum processing power of resources (Yousefinezhad et al, 2012).

2 CLASSIC ALGORITHMS

Scheduling must be performed in such a way that it can minimize total execution time of program on the basis of tasks time and the relationship between processors. Scheduling problem of tasks graph is a NP-Hard problem (Grace and Johnson, 1979). Hence approaches based on deterministic methods are efficient. Using evolutionary processing algorithm, mostly genetic algorithm, chakoos algorithm and other revolutionary algorithm are efficient to solve this problem due to their undeterministic feature. In previous approaches, tasks graph in multi-processor architecture of non-genetic algorithm has been proposed to solve scheduling problem. In this cases, there are six important non-genetic algorithms involving LAST (Wong et al, 1998; Bechester and Patel, 1989), EFT (Gari and Johnson, 1979), HLEFT (Sih and Li, 1993), EZ(Couria et al, 1999), MCP (Kaffman, 1976; Weo et al , 1990), and LC (Dadhi and Ahmad, 1995). Among these algorithms, MCP is the best non-genetic algorithm to solve this problem so it is used to evaluate and compare other algorithms.

3 GENETIC ALGORITHM

Evolutionary account was presented in 1960 for the first time by Richebery, He studied evolution strategy. Later, his theory was studied and investigated by many researchers. Then, genetic algorithm presented by John Holland in 1975 in Michigan University. Genetic algorithm has been inspired from genetic science and Darwin's theory of evolution, and it is based on the survival of the best or natural selection. A common application of genetic algorithm is using it as an optimization function. Genetic algorithm is a useful tool in terms of pattern detection, feature selection, image perception and machinery learning. This algorithm has been inspired from nature, and it is based on evolutionary principle of survival of the fittest. Although genetic algorithm was proposed after evolutionary strategy algorithm, it is the most well-known method among evolutionary algorithms. In genetic algorithm, population survives according to their desirability. Those having better capability

will have more marriage and reproduction chance. Therefore, children are created with better efficiency after some generations. In genetic algorithm, each person is introduced as a chromosome. The chromosomes are completed in each generation. Chromosomes are evaluated in each generation, and they survive on the basis of their value. Reproduction is considered in genetic algorithm by using host operators and mutation. The superior parents are selected on the basis of fitness function.

In each execution step of genetic algorithms, some points of search space are randomly processed. A sequence of characters is attributed to each point, and genetic operations are applied on these sequences.

Then, the obtained sequences are decoded so that new points are obtained in search space. Their participants in the next step are determined by considering the value of objective function in each point.

Genetic algorithms are oriented and random optimization method, and they gradually move toward the optimal point. It can be said that, in comparison to other optimization methods, genetic algorithm is an algorithm that can be applied for every problem without knowing the problem and without applying any limitations on its variables. Also, it has proved efficiency in finding global optimum. This method can solve complex optimization problems, or it can perceive reliable and valid global optimum.

Genetic algorithm is used to search in large space, and this map is usually used to solve special problems. The results obtained from this algorithm show that it is one of the best map methods (Zhang et al, 2011). When are meta-heuristic and innovative methods used?

Simple problems with too large samples (due to large dimensions of the problem, solving it precisely it costly).

- Simple problems with much real time limitations (dynamic optimization problems).
- Optimization problems with objective functions or time-consuming limitations (high computing cost in precise methods).
- Non-analytical models
- Complex simulation problems

Control problems (such as traffic management in which decisions are in the range of the seconds).

4 THE WAY OF DECODING CHROMOSOME

Each allocation is shown as a string constituting of a solution. Each string is used as a genetic algorithm chromosome. Figure 8-2 shows a view of a chromosome.

Table 1: A sample of chromosome used in genetic algorithm

VM1	VM2	VM3	VMn
Host1	Host2	Host3	Hostn

An appropriate resource has been considered in each solution and for each virtual machine among physical hosts. It shows that each virtual machine is allocated to physical host in related column of the map. Each chromosome involves N genes constituting the chromosome length. The number of genes can be the number of tasks entered to the system or prepared tasks.

Data center of cloud system is responsible for collecting and keeping information related to operation power in resources and its other features (Lois et al, 2009). In the method, a resource is considered for each task to provide a chromosome from the ordered virtual list of available resources, and it is derived from the resource list of cloud system data center. This list is updated to perform some operations like creating initial population, cross over and mutation.

5 SCHEDULING

Scheduling of resources is very important in cloud computing environment. Each user may face with hundreds of virtual resources in executing the task in cloud computing environment. However, allocating the tasks to virtual resources by the user is impossible. Scheduling system controls various tasks in the cloud system to increase the rate of task completion ability of resources and computing power. Cloud computing is highly dependent on virtualization. It can be said that clouds are virtual branches so tasks scheduling in various and heterogeneous physical machines is a crucial task. Scheduling in cloud computing environment has been considered as one of the most important challenges in cloud computing. In cloud computing systems, computing resources are presented as virtual machines. In such a scenario, scheduling algorithm plays a very important role because the purpose of scheduling is the efficiency of tasks to decrease the time and to improve resources utilization. In a cloud environment, the user may

use hundreds of computing resources. Therefore, scheduling cannot be performed manually. An important point of scheduling is keeping information related to resources (Zhang Yan et al, 2011).

6 METHODS AND COMMENTS

The main assumptions are as follows:

1) Every machine can process several tasks each time (by using virtual machine).

2) Each task is processed just by one machine each time. Hence, one operation cannot be allocated to two resources simultaneously. In this regard, we define fitness function.

$$FF = \sum_{j=1}^n \min(\text{execution times}) + FL$$

3) Tasks are continuously performed, and they must not be stopped.

4) Objective function of the model refers to minimizing time of executing whole project in a way that transposition relations are followed among activities.

The last phase occurring after scheduling is integrating empty spaces resulting from scheduling and repetition, so the response time reduces.

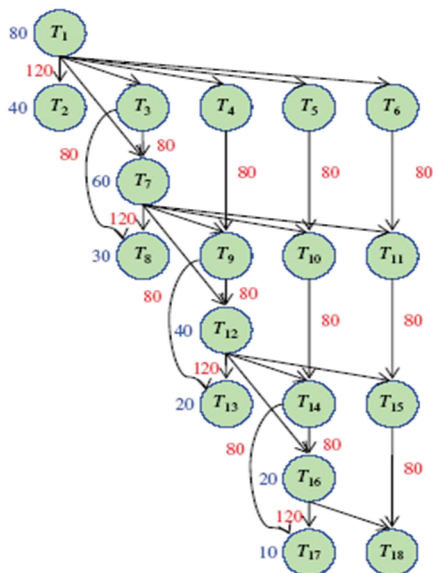
The input can be explained as directed acyclic graph called tasks graph of $G=(V,E)$ [9]. Each node is a member of V set, and it shows a task unit of the program. The weight of these nodes determines execution time of related task unit. Also, this graph involves a set of edges, E, showing prerequisite relations among task units. When there is an edge in the form of (t_i,t_j) , t_j can not initiate the executing until t_i completes its execution. These edges are weighted, and the weight of each edge shows communication cost and sending the message between two task units. This cost of time is considered when two related task units are executed in different processors. If the processors are same, then communication cost will be Zero.

7 THE PROPOSED ALGORITHM

The proposed algorithm performs its tasks by considering middle population rather than random population. In this algorithm, computing the earliest time for executing each node is required for the list of task units' priorities. This parameter is shown by $Ttlevel(n_i)$, and it is the length of longest route from initiation node to n_i regardless of execution time of n_i node. Along with executing this value,

critical path and its length can be determined. This algorithm has been designed in such a way that it tries to make the critical path shorter and to reduce inter-processor communication cost. Communication cost will be Zero among them by executing two task units on a unit processor.

In this section, the proposed algorithm for graph of figure 15 is studied and compared with the algorithms of GA, DCP, MD, DSC and MCP.



Performance Processors				
P1	P2	P3	P4	Average performance processors
76.9%	15.3%	13.4%	7.6%	28.3%

Fig. 1. An example of task graph with 18 tasks

Fig. 2. Gantt graph obtained from MCP algorithm with 4 processors

In figure 2, Gantt chart of algorithms displayed in the table of figure 2 is demonstrated in the task of figure 2 along with processors efficiency in graph of figure 1.

Performance Processors						
P1	P2	P3	P4	P5	P6	Average performance processors
93.4 %	4.3 %	6.5 %	8.6 %	8.6 %	8.6 %	21.66%

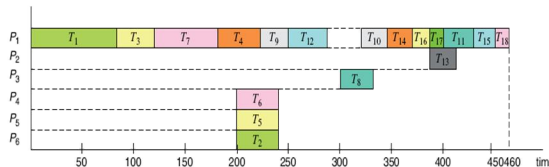


Fig. 3. Gantt graph obtained from DSC algorithm with processors

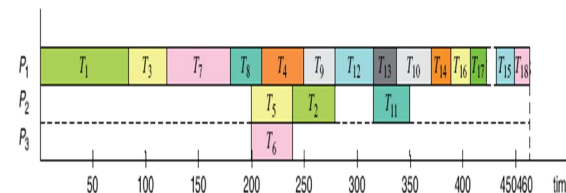


Fig. 4. Gantt graph obtained from MD algorithm with 3 processors

Performance Processors			
P1	P2	P3	Average performance processors
95.45%	15.9%	25%	45.45%

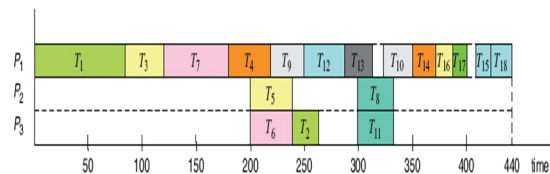
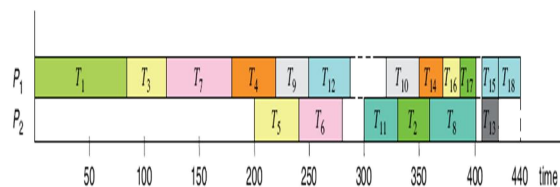


Fig. 5. Gantt graph obtained from DCP algorithm with 3 processors



Performance Processors		
P1	P2	Average performance processors
90.9%	45.45%	68.17%

Fig. 6. Gantt graph obtained from GA with 2 processors

Table 2: The results obtained from MCP, DSC, MD, DCP and genetic algorithms

GA	DCP	MD	DSC	MCP	The algorithm
5	4	3	2	1	Number of
2	3	3	6	3	processors
440	440	460	460	520	End time

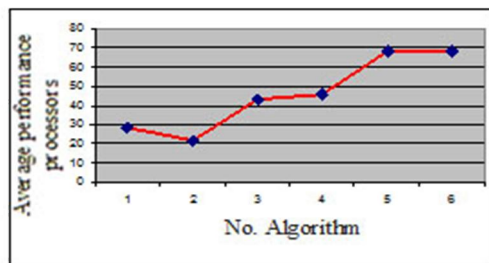


Fig. 7. comparing the efficiency of processors condition

We compare the proposed algorithm with well-known algorithm of MCP and some other algorithm. Also, we consider the graphs in which the result of their scheduling is specified under these algorithms. As it has been shown, the result of presented algorithm is better than mentioned classic algorithms. The proposed algorithm is executed quicker than other algorithms when the number of tasks increases. In addition, the proposed algorithm has more stability, and in most cases, optimal answer is obtained.

8 REFERENCES

- [1] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. In: Future Generation Computer Systems, vol. 25(6), pp. 599–616. Elsevier Science, Amsterdam (2009)
- [2] Yashpalsinh Jadeja, Kirit Modi, "Cloud Computing - Concepts, Architecture and Challenges" International Conference on Computing, Electronics and Electrical Technologies [ICCEET], 2012.
- [3] Tim Forell, Dejan Milojevic, Vanish Talwar, Hewlett-Packard, "Cloud Management: Challenges and Opportunities", IEEE, IEEE International Parallel & Distributed Processing Symposium [ICCEET], 2011.
- [4] Zhongyuan Le1, Ying Wang, Wen Zhou, "A dynamic priority scheduling algorithm on service request scheduling in cloud computing", International Conference on Electronic & Mechanical Engineering and Information Technology, 2011.
- [5] Humar Kahramanli (2012), "A Modified Cuckoo Optimization Algorithm for Engineering Optimization", International Journal of Future Computer and Communication, Vol 1(2), pp 199-201.
- [6] R. Maggiani, "Cloud Computing is changing how we communicate," presented at the IEEE International Professional communication Conference, 2009.
- [7] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 1979.
- [8] Y. K. Kwok and I. Ahmad, Benchmarking and Comparison of the Task Graph Scheduling Algorithms, Proceedings of the 12th International Parallel Processing Symposium, April 1998, pp. 531-537.
- [9] J. J. Hwang, Y. C. Chow, F. D. Anger and C. Y. Lee, Scheduling Precedence Graphs in Systems with Interprocessor Communication Times, SIAM J. on Comp., Vol. 18, No. 2, Apr. 1989, pp. 244-257.
- [10] G. C. Sih and E. A. Lee, A Compile-Time Scheduling Heuristic for Interconnection-Constrained Heterogeneous Processor Architectures, IEEE Trans. on Parallel and Distributed Systems, Vol.4, No. 2, Feb. 1993 pp. 75-87.
- [11] J. Baxter and J. H. Patel, The LAST Algorithm: A Heuristic- Based Static Task Allocation Algorithm, Proc. of Int'l Conf. Parallel Processing, Vol. 2, Aug. 1989, pp. 217-222.
- [12] E. G. Coffman, Computer and Job-Shop Scheduling Theory, Wiley, New York, 1976.
- [13] M. Y. Wu and D. D. Gajski, Hyper-tool: A Programming Aid for Message-Passing Systems, IEEE Trans. on Parallel and Distributed Systems, Vol. 1, No. 3, Jul. 1990, pp. 330- 343.
- [14] M. K. Dhodhi, and I. Ahmad, A Multiprocessor Scheduling Scheme Using Problem-Space Genetic Algorithms, Proc. IEEE Int. Conf. Evolutionary Computation, 1995.
- [15] Chandrashekhar S. Pawar and R.B.Wagh.(2012). A review of resource allocation policies in cloud computing. World Journal of Science and Technology, 2(3):165-167.
- [16] Person.S. (2009). Taking Account of Privacy when Designing Cloud Computing Service. IEEE Computer Society Washington. DC, USA, pp 44-52.

- [17] M. Y. Wu and D. D. Gajski, Hyper-tool: A Programming Aid for Message-Passing Systems, *IEEE Trans. on Parallel and Distributed Systems*, Vol. 1, No. 3, Jul. 1990, pp. 330-343.
- [18] A. Gerasoulis and T. Yang, A Comparison of Clustering Heuristics for Scheduling DAGs on Multiprocessors, *J. Parallel and Distributed Computing*, Vol. 16, No. 4, Dec. 1992, pp. 276-291.
- [19] Y. K. Kwok and I. Ahmad, Benchmarking and Comparison of the Task Graph Scheduling Algorithms, Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong, March 17, 1999.
- [20] B. Shirazi, M. Wang and G. Pathak, Analysis and Evaluation of Heuristic Methods for Static Scheduling, *J. Parallel and Distributed Computing*, No. 10, 1990, pp. 222-232.
- [21] E. S. H. Hou, N. Ansari, and H. Ren, A Genetic Algorithm for Multiprocessor Scheduling, *IEEE Trans. on Parallel and Distributed Computing*, Vol. 5, No. 2, February 1994, pp.113-120.
- [22] R. C. Correa, A. Ferreira and P. Rebreyend, Scheduling Multiprocessor Tasks with Genetic Algorithms, *IEEE Trans. Parallel and Distributed Systems*, Vol. 10, No. 8, 1999, pp 825-837.
- [23] M. K. Dhodhi, and I. Ahmad, A Multiprocessor Scheduling Scheme Using Problem-Space Genetic Algorithms, *Proc. IEEE Int. Conf. Evolutionary Computation*, 1995.
- [24] Y. Kwok and I. Ahmad, "Static Scheduling Algorithms for Allocating Directed Task Graphs to Multiprocessors, *ACM Computing Surveys*, Vol. 31, No. 4, 1999, pp. 406-471.