

An Efficient Heuristic to Solve a Multi-objective Scheduling Problem in a Cloud Environment

Fatima Hassan and Jihene Kaabi

University of Bahrain, Shakir, Bahrain
fjazeeri2003@hotmail.com; jkaapi@uob.edu.bh

ABSTRACT

Due to the spread usage of cloud computing and the dramatic increase on resources used by cloud providers such as the data centers, the effect of those resources becomes a major concern of many organizations. Since the huge amount of energy used in those data centers has noticed impact on the world. And by the years the usage is increasing dramatically.

On other hand, cloud providers try to not only minimize the energy consumption but also focus on satisfying customers by completing their requests as soon as possible. To solve this problem, providers use scheduling algorithms to get good balance between makespan (the ending time of the last processed job) and energy consumed in cloud when scheduling tasks to the available machines.

The main contribution of this study is to implement a multi objective genetic algorithm to solve a scheduling problem in a cloud environment (MOGAC) that aims to reduce the makespan and the energy consumption was proposed. The algorithm was implemented using Java and was compared to energy aware min-min algorithm earlier proposed in the literature. The proposed algorithm achieves up to 17% and 0.47% reduction in makespan and energy consumption, respectively, in many situations.

Keywords: Cloud Computing, Scheduling Problem, Genetic Algorithm, makespan, energy consumption.

1. INTRODUCTION

Cloud Computing is an attractive way of computing that motivates many organizations to shift to it because of its features like: lowering the cost, minimizing the overhead of maintaining the resources, etc. However, this rapid growth in using the Cloud presents new concerns from Cloud providers' point of view of how to provide customers with good services with the lowest cost. Because of that, different studies and researches have been done to provide best task scheduling methods to achieve customer satisfaction. The aim of this study is to propose an algorithm to find a solution for scheduling problem in a cloud environment minimizing the makespan and the power consumption.

The paper will be organized as follows: Section 2 will study the research area and the available solutions proposed to solve the area's investigated problems. Section 3 will discuss the problem notation and the proposed approach. Then all experimental results will be discussed in section 4. Finally, the conclusion and the future work will be covered in section 5.

2. RELATED WORK

Many researchers focus on scheduling in Cloud recently because of the high impact it can provide for utilizing different resources and achieving some objectives. Many algorithms as well as heuristics were proposed to solve the scheduling problem in Cloud environments while considering different objectives like cost, makespan, or energy minimization. The authors in [1] give out a multi-model estimation distribution algorithm of parallel application that is objects to reduce time and energy consumed. The algorithm proposed was compared with both heuristics and parallel bi-objective genetic algorithm and provides enhanced solutions in term of energy saving and makespan minimization. However, the paper does not consider neither heterogeneity of the cloud nor uncertainty factor. Whereas, the authors in [2] propose an energy aware Min-Min algorithm (EAMM) with the aim of minimizing energy and makespan. The algorithm was tested with to the original Min-Min algorithm, and it achieves better results in most situations. In [3], the authors developed a real time dynamic scheduling system that has a goal to minimize the energy consumed and the makespan for task-based

applications in Cloud. The outcome of the algorithm shows that it can save good amount of energy while respecting the makespan. In [4], the authors proposed an approach that aims to minimize the energy consumed in Cloud data centers. This approach also concerns about both the completion time of the tasks and reliability. Whereas, in [5], the authors proposed a fuzzy dominance sort based heterogenous earliest-finish-time (FDHEFT) algorithm to minimize the cost and makespan of workflows in IaaS Clouds. The algorithm proved its effectiveness compared to ϵ -FDPSO, MOPSO, MOHEFT and NSPSO. Moreover, the authors in [13] proposed a heuristic to minimize energy, time and throughput in Cloud. The results show that it can produce promising results for the three objectives. As well as each single objective compared to other algorithms has better results.

Evolutionary algorithms were also used like particle swarm (PSO), clonal selection or genetic algorithm (GA). In [6], the authors presented a hybrid Particle Swarm (PSO) and a hill climbing algorithms that aim to minimize the makespan in Clouds by maximizing parallelization. The algorithm provides better results comparing to original PSO and HEFT, but it does not consider load balancing issue. The study developed in [8], presented a multi-objective Clonal Selection Algorithm for cloud computing environment. The algorithm has two objectives which are minimizing the makespan and energy consumed. According to the results it can reduce the energy consumption by 10-30% and makespan by 5-25%. In [7] a multi-objective genetic algorithm for Cloud Environment was proposed with the aim of minimizing makespan and energy consumption. The algorithm was tested with different real workload traces and compared to many well-known algorithms. It shows better results in both energy and makespan in most of the cases. The authors in [14] addressed the energy and makespan constraints by implementing a Pareto-based multi-objective discrete ant lion algorithm (PBMO-DALO). The algorithm provides better results when compared to some robust algorithms. However, the number of tasks used in experiment was very small (20-100) to Cloud environment. And used as well in [15] along with GA to reduce time of tasks. The resulted algorithm

achieved up to 64% of time reduction than other well-known algorithms. And this improvement affect the overall cost that reduced up to 11%.

The previous studies showed some good results in solving the scheduling problem in Cloud. However, it is still an open area of research since heuristic and meta-heuristic methods did not produce optimal solution. The proposed algorithm in this study will cover two important factors namely: makespan and energy consumption.

3. THE PROPOSED APPROACH

3.1 Problem Description

In Cloud Environment, a Cloud data center consists of multiple parallel processors each of them has multiple nodes to process the user jobs. However, because of the uncertainty of Cloud environment the number of user jobs is always unknown. Also, the user jobs may consist of different number of tasks. The scheduling problem in a Cloud environment should consider the assignment of the tasks in a job to the available nodes on one of the processors while optimizing some objectives. This study will focus on two objectives namely: minimizing makespan and energy consumption.

According to Graham's three-field notation ($\alpha|\beta|\gamma$) for a scheduling problem presented in [12], the investigated problem is denoted by:

$$P||TEC, Cmax$$

Where P represent identical parallel machines, TEC presents the first objective (the total energy consumed) and $Cmax$ the second objective (the makespan)

In this study, we assume that the number of processors and nodes is known in advance. Each task in a job is characterized by two values namely: ready time and processing time. *Ready time* is the arrival time of a task. Thus, the task cannot start before that time. And the *Processing time* is the running time of each task.

Also, each processor will have a power consumption when it is in computing state and when it is in idle state. Table1 illustrates the notations used in this study.

Table 1. Problem Notations

Notation	Meaning
J	Number of jobs
j	Job index, where $j=1,2,3,\dots,J$
T	Number of tasks in each job
k	Task index, where $k=1,2,3,\dots,T$

P	Number of processors
i	Processor index, where $i = 1, 2, 3, \dots, P$
N	Number of nodes in each processor
n	Node index, where $n = 1, 2, 3, \dots, N$
$Elast_{in}$	Ending time of the last task assigned to node n of processor i .
TC_{in}	Total processing times of the tasks assigned to node n of processor i .
IT_{in}	The idle time of node n of processor i .
$Start_{in}$	The ready time of the first task assigned to node n of processor i .
C_{in}	The energy consumption of node n in processor i when it is computing.
I_{in}	The energy consumption of node n in processor i when it is idle.
P_s	Population size

3.2 Objective Function

Suppose that user task JT_{jk} is assigned to the available node PN_{in} . The makespan for each node can be calculated as in (1):

$$\text{Makespan} = \max (Elast_{in}) - \max (Start_{in}) \quad (1)$$

And the energy consumption can be calculated using as in (2):

$$\text{Energy} = \sum_{in} (TC_{in} * C_{in}) + (IT_{in} * I_{in}) \quad (2)$$

Where $IT_{in} = Elast_{in} - T_{cin}$

The objective of this study is to simultaneously minimize the Makespan and the consumed Energy.

3.3 The proposed algorithm (MOGAC)

In order to find a solution for the problem of task scheduling in Cloud, a Genetic algorithm (GA) was proposed. The Genetic algorithm is a well-known meta-heuristic algorithm that is widely used in solving complex optimization problems. The idea of GA was introduced in [9]. By the years the algorithm proved its effectiveness in giving near optimal solutions in a short time.

MOGAC starts with randomly generating an initial population. The population consists of 100 individuals (chromosomes) that represent potential solutions to the problem. Then the fitness value (C_{max} , TEC) will be calculated for each chromosome using Eq. (1) and Eq. (2) mentioned above. Since there are two objectives, the dominance percentage value defined in [11] will be used. Then the population will be sorted

in ascending order based on the dominance percentage values calculated to each chromosome. The selected chromosomes will go for the crossover step, two-point crossover will be applied to 80% of the population. After that, a mutation process will take place. The random resetting mutation will be used for 50% of the population. And finally, the algorithm will stop after 20 iterations.

Then to test the effectiveness of the proposed algorithm, the EAMM algorithm mentioned in the literature will be used [2].

4. EXPERIMENTAL RESULTS

The algorithm was implemented using IntelliJIDEA (Java language). The experiments were performed on different scenarios to test the effectiveness of the algorithm. Those scenarios were based on three main properties namely: schedule interval of tasks or (ready time), the number of tasks and the number of nodes. The number of machines was fixed to 4 processors in all experiments. Also, it is assumed that each job has only one task, so in the discussion jobs and tasks will consider same. The tasks' processing times were generated using a uniform distribution function in Statistical Package for the Social Sciences application (SPSS). In addition, a real workload traces defined by [10] named HPC2N is used. All those scenarios aim to test the algorithm under different situations and load to make sure it can produce better results when using in Cloud. In order to calculate the energy consumed, the same characteristics used in [7], were adopted in our algorithm to compare the effectiveness of the algorithms. Table 2 shows the parameters used in each scenario.

Table 2. Parameters in Each Scenario

Scenario	Number of Jobs	Number of Nodes/ in each Processor	Schedule Interval (Ready time) (Seconds)
1	Ranged 1000 to 10000	64	40
2	20,000	64	Extracted from [10]

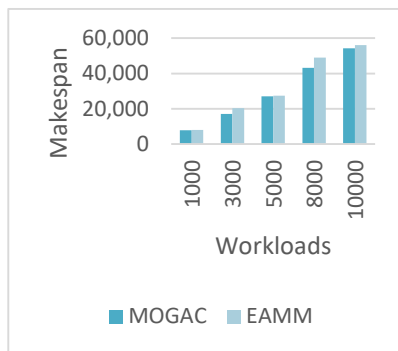


Fig. 1. Makespan Comparison Under Different Workloads

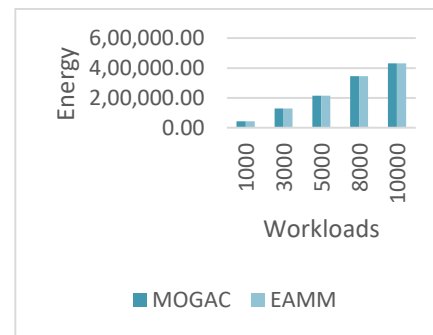


Fig. 2. Energy Comparison Under Different Workloads

As shown in the Figures 3 and 4 for the second scenario, the proposed algorithm achieves better than EAMM in real situations even though the number of

tasks is huge. It was able to improve the makespan and energy by 9% and 0.27% respectively.

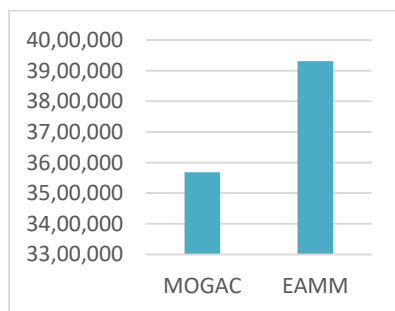


Fig. 3. Makespan Comparison of HPC2N Workload

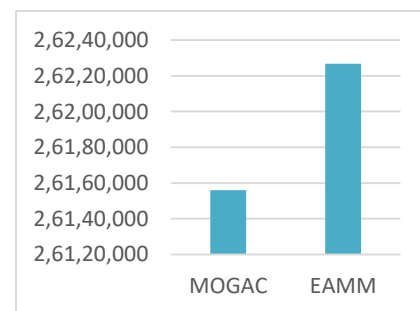


Fig. 4. Energy Comparison of HPC2N Workload

According to the experiments done under different situations, the proposed MOGAC algorithm proved its effectiveness in minimizing the considered objectives namely the makespan and the energy consumption as it gives noticeable reduction compared to a robust algorithm (EAMM). This is an indication that the selection, crossover, and mutation operators are working well to find a near optimal solutions. It's also clear that the algorithm can deal with different situations and different loads without affecting its

effectiveness, so that it will be able to work with large heterogeneous environment like the Cloud.

5. CONCLUSION AND FUTURE WORK

This study proposes a multi-objective genetic algorithm to solve the scheduling problem in a Cloud environment. The aim of the algorithm is to reduce the

makespan and energy consumed in Cloud datacenters. The algorithm was compared with EAMM algorithm earlier proposed in the literature and showed that it can perform better than EAMM in many situations. For further studies, more objectives will be considered such as: cost, load balance, bandwidth, etc.) Also, other simulations tools such as CloudSim will be used since it supports the simulation for large scale datacenters along with their processors and virtual machines. This simulation will prove the efficiency of the proposed algorithm.

REFERENCES

1. Wu, C., Wang, L.: A multi-model estimation of distribution algorithm for energy efficient scheduling under cloud computing system. *Journal of Parallel and Distributed Computing* (117), 63-72 (2018)
2. Li, Y., Liu, Y., Qian, D.: A Heuristic Energy-aware Scheduling Algorithm for Heterogeneous Clusters. 2009 15TH INTERNATIONAL CONFERENCE ON PARALLEL AND DISTRIBUTED SYSTEMS. (2009)
3. Juarez, F., Ejarque, J., Badia, R.: Dynamic energy-aware scheduling for parallel task-based application in cloud computing. *Future Generation Computer Systems* (78), 257-271 (2018).
4. Wu, T., Gu, H., Zhou, J., Wei, T., Liu, X., Chen, M.: Soft error-aware energy-efficient task scheduling for workflow applications in DVFS-enabled cloud. *Journal of Systems Architecture* (84), 12-27 (2018).
5. Zhou, X., Zhang, G., Sun, J., Zhou, J., Wei, T., Hu, S.: Minimizing cost and makespan for workflow scheduling in cloud using fuzzy dominance sort-based HEFT. *Future Generation Computer Systems* (93), 278-289 (2019).
6. Dordaie, N. and Navimipour, N.: A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. *ICT Express* (2017).
7. Gabaldon, E., Lerida, J., Guirado, F., Planes, J.: Blacklist multi-objective genetic algorithm for energy saving in heterogeneous environments. *The Journal of Supercomputing* 73(1), 354-369 (2016).
8. Jena, R.: Energy Efficient Task Scheduling in Cloud Environment. *Energy Procedia* (141), 222-227 (2017).
9. Holland, J.H. and Holl, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*, University of Michigan Press, Ann Arbor (1975).
10. Feitelson, D. Parallel workloads archive. <http://www.cs.huji.ac.il/labs/parallel/workload>, last accessed 2020, 3, 20.
11. Fonseca, C.M., Fleming, P.J.: Genetic algorithms for multiobjective optimization: formulation, discussion and generalization, *ICGA* (93), 416-423(1993).
12. Graham, R.L., Lawler, E.L., Lenstra, J.K., Kan, A.: Optimization and approximation in deterministic sequencing and scheduling: A survey, *Annals of Discrete Mathematics* (5), 287- 326 (1979).
13. Gu, Y., Budati, C.: Energy-aware workflow scheduling and optimization in clouds using bat algorithm, *Future Generation Computer Systems* (113), 106-112 (2020)
14. Rani, R., & Garg, R.: Pareto based ant lion optimizer for energy efficient scheduling in cloud environment, *Applied Soft Computing* (113), 107943 (2021)
15. Ajmal, M. S., Iqbal, Z., Khan, F. Z., Ahmad, M., Ahmad, I., & Gupta, B. B.: Hybrid ant genetic algorithm for efficient task scheduling in cloud data centers. *Computers & Electrical Engineering* (95), 107419 (2021)