

Job Scheduling based on Harmonization between the Requested and Available Processing Power in the Cloud Computing Environment

Elhossiny Ibrahim^{1*}, Nirmeen A El-Bahnasawy¹ and Fatma A Omara²

¹Computer Science and Engineering Department, Cairo University, Menouf 32952, Egypt

²Faculty of Computers and Information, Cairo University, Egypt

Abstract

The Cloud Computing is a most recent computing paradigm where IT services are provided and delivered over the Internet on demand and pay as you go. On the other hands, the task scheduling problem is considered one of the main challenges in the Cloud Computing environment, where a good mapping between the available resources and the users's tasks is needed to reduce the execution time of the users' tasks (i.e., reduce make-span), in the same time, increase the degree of capitalization from resources (i.e., increase resource utilization).

In this paper, a new task scheduling algorithm has been proposed and implemented to reduce the make-span, as well as, increase the resources utilization by considering independent tasks. The proposed algorithm is based on calculating the total processing power of the available resources (i.e., VMs) and the total requested processing power by the users' tasks, then allocating a group of users' tasks to each VM according to the ratio of its needed power corresponding to the total processing power of all VMs.

To evaluate the performance of the proposed algorithm, a comparative study has been done among the proposed algorithm, and the existed GA, and PSO algorithms. The experimental results show that the proposed algorithm outperforms other algorithms by reducing make-span and increasing the resources utilization.

Keywords: Cloud computing; Task scheduling; Particle swarm optimization; Genetic algorithm

Introduction

The Cloud Computing allows the users to use the computational resources and services of the data centers (i.e., machines, network, storage, operating systems, application development environments, application programs) through the Internet [1]. The main feature of the Cloud Computing is that it provides self-provisioning of IT resources [2]. The Cloud Computing architecture is categorized as layers; service model, and deployment model (types). The deployment models are public, private, hybrid, community, and federated, while the Cloud Computing services are divided into Software as a Service (SaaS), Platform as a Services (PaaS), and Infrastructure as a Services (IaaS) [3]. By this classification, the users can easily choose the suitable cloud services and types to fit their requirements according to these services [1,3].

The main advantages of the cloud is that servers and IT equipments are collected in a single place or distributed over different data centers and can be access remotely. So, the Users can access the cloud services through the Internet by using mobile, PC, laptop, PDA, etc. (Figure 1) [4].

In the Cloud Computing, there are some issues should be concerned. The main issues are resource management and task scheduling. Therefore, the main function of the Cloud providers is to provide the services to the users by deploying virtual machines (VMs), and defining scheduling policies to allocate VMs to the users' requests [5].

In this paper, a new task scheduling algorithm has been proposed. The main principle of the proposed algorithm is based on calculating the total processing power of the available resources (i.e., VMs) and the total requested processing power by the users' requests, then allocating a group of the users's requests to each VM according to the ratio of its needed power corresponding to the total processing power of all VMs. To evaluate the performance of the proposed algorithm, a comparative study has been done among the proposed algorithm, the existed GA, and PSO algorithms. The experimental results show that our algorithm

outperforms other algorithms by minimizing the total execution time of the users' requests and increasing the utilization of the resources.

The remainder of the paper is organised as follows. In Section

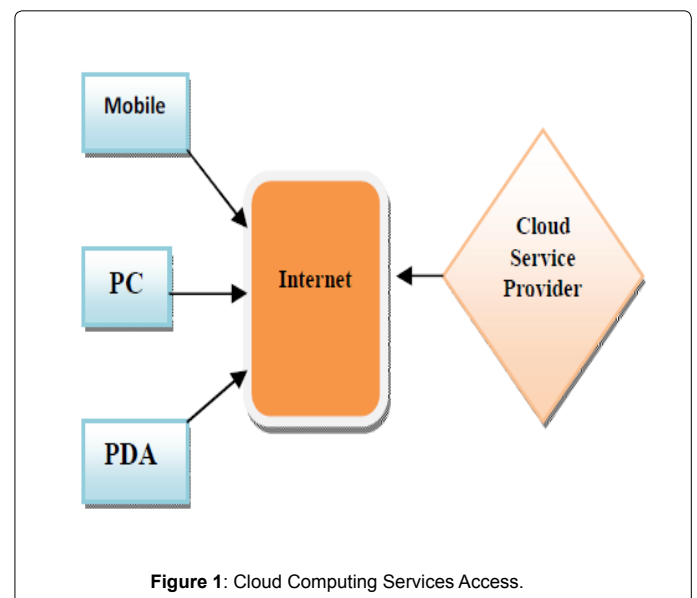


Figure 1: Cloud Computing Services Access.

***Corresponding author:** Elhossiny Ibrahim, Faculty of Electronic Engineering, Computer Science and Engineering Department, Cairo University, Menouf 32952, Egypt, Tel: +20-2-35676105; E-mail: elhossiny.ibrahim@el-eng.menofia.edu.eg

Received August 14, 2015; **Accepted** September 07, 2015; **Published** September 25, 2015

Citation: Ibrahim E, El-Bahnasawy NA, Omara FA (2015) Job Scheduling based on Harmonization between the Requested and Available Processing Power in the Cloud Computing Environment. J Inform Tech Softw Eng 5: 155. doi:10.4172/2165-7866.1000155

Copyright: © 2015 Ibrahim E, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

II, the related work is illustrated followed by the fundamental of our proposed task scheduling algorithm in Section III. In Section IV, the performance evaluation of the proposed algorithm via Cloudsim simulator and the comparative study are discussed. Finally, in Section V, we will conclude our contributions and point out the future work.

Related Work

Yun et al. [6] have proposed an innovative transaction intensive cost-constraint scheduling algorithm to minimize the cost and the execution time. The simulation results proved that this algorithm could minimize the cost while meeting the user designated deadline. Suraj Pandey et al. [7] have proposed a heuristic task scheduling to minimize the cost of task-resource mapping using particle swarm optimization (PSO) algorithm. According to the simulation results, it is found that the proposed algorithm can achieve as much as 3 times cost savings as compared to BRS, and good distribution of workload onto resources. Ke Liu et al. [8] have presented a Compromised-Time-Cost (CTC) scheduling algorithm. The CTC algorithm considers the characteristics of Cloud Computing to accommodate instance-intensive cost-constrained workflows by compromising the execution time and cost of the users' tasks which are enabled on the fly. The simulation results has demonstrated that the CTC algorithm can achieve lower cost while meeting the user-designated deadline or reducing the mean execution time within the user- designated execution cost. Parsa and Reza [9] have proposed a task scheduling algorithm called Resource-Aware-Scheduling (RASA). It is considered an amalgamation of two traditional scheduling algorithms; Max-min and Min-min. Therefore, the deadline of each task, arriving rate of the tasks, cost of the task execution on each of the resource, and cost of the communication are not considered. The experimental results proved that RASA algorithm outperforms the existing scheduling algorithms in large scale distributed systems. Huang [10] has proposed a workflow task scheduling algorithm based on the genetic algorithms (GA) model in the Cloud Computing environment. The experimental results proved that the efficiency of resource allocation has been satisfied, and in the same time, minimized the completion time. Lei Zhang et al. [11] have proposed a PSO algorithm. This proposed algorithm is similar to the genetic algorithms (GA). The aim of this algorithm is to improve the efficiency of resource allocation and minimize the completion time simultaneously. It is noted that the performance of PSO usually spent shorter time to accomplish the various scheduling tasks and specifies better result comparing to the GA algorithm. Also, they have proved that the PSO algorithm can get better effect for a large scale optimization problem. Cui Lin and Shiyong Lu [12] have proposed an Scalable Heterogeneous Earliest-Finish-Time Algorithm (SHEFT) workflow scheduling algorithm to schedule a workflow elastically on a Cloud Computing environment. The experimental results show that SHEFT is not only outperform several representative workflow scheduling algorithms in optimizing workflow execution time, but also enable resources to scale elastically at runtime. Visalakshi and Sivanandam [13] have presented Hybrid Particle Swarm Optimization (HPSO) method for solving the Task Assignment Problem (TAP) by dynamically schedule heterogeneous tasks on to heterogeneous processors in a distributed setup. The HPSO yields a better result. The experimental result of the HPSO algorithm has proved that the PSO outperforms the GA algorithm. Selvarani and Sudha Sadhasivam [14] have proposed an improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in the cloud. This scheduling algorithm divides all user tasks depending on the priority of each task into three different lists. This scheduling algorithm measures both resource cost and computation performance. Also, it

improves the computation/communication ratio. Yang et al. [15] have highlighted the issue of job scheduling in the Cloud Computing with considering hardware/software failure and recovery. They have proposed a Reinforcement Learning (RL) based algorithm that helps the scheduler to define scheduling decision with fault tolerable while maximizing utilities attained in the long term.

The Proposed Algorithm

According to the work in this paper, a new task scheduling algorithm on the Cloud Computing environment has been proposed. The main idea of the proposed algorithm is that allocating the available VMs to the requested tasks according to the following steps:-

1. Calculate the total available processing power of the available VMs (i.e., VM_MIPS).
2. Calculate the total requested processing power by the users' tasks (TASKS_MIPs).
3. Calculate the power factor (PF) of each available resource (VM_i) using equation (1).

$$\text{PF of VM}_i = (\text{Processing Power of VM}_i) / (\text{Total Processing Power of All VMs}) \quad (1)$$

4. Calculate the allotment for each existed resource (VM_i) as equation (2).

$$\text{VM}_i \text{ allotment} = \text{PF} * (\text{total requested processing power (TASKS_MIPs)}) \quad (2)$$

5. Search the requested tasks to find a task or a group of tasks which need processing power equal to or less than the allotment of the VM_i as calculated in step (4) by considering that the difference between the selected tasks processing power and VM allotment to be minimum.

The pseudo code of the proposed algorithm is as follows:

```
Input:
  Number n of cloudlets (i.e., tasks).
  Number m of VMs (i.e., resources).
Output:
  Mapping Scheme for the requested tasks (cloudlets) on the available
  resources (VMs).
// calculate the total processing power of cloudlets
1 for i=1 to n do
2   Define MIPS of the cloudlet
3 endfor
// calculate the total processing power of VMs.
4 for i=1 to m do
5   Define MIPS of the VMs
// calculate the power factor of each VM, and the allotment of each VM
6 for i=1 to m do
7   PF of VMi = processing power of VMi / total processing power
of all VMs
8 VMi allotment = PF * (total requested processing power (TASKS_MIPs)).
9   find the sum of set (VMi allotment).
10 endfor
11   Select a set of cloudlets with the sum of their MIPS equal to/
or less VMi allotment.
12   return the set.
```

Performance Evaluation

To evaluate the performance of the proposed algorithm, a comparative study has been done among the proposed algorithm

and the GA, PSO, and First-Come-First-Serve (FCFS) algorithms by considering two performance parameters; Make-span, and resource utilization.

Experimental environment

The proposed task scheduling algorithm has been written by java programming language using eclipse program in Intel(R) Core(TM) 2 Duo CPU in 2.10 GHZ of processor and 4.00 GB of RAM, through the CloudSim simulator. On the other hands, CloudSim is an extensible simulation toolkit that enables modeling and simulation of Cloud Computing systems and application provisioning environments. The CloudSim toolkit supports both system and behavior modeling of the Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. The Cloudsim Implements generic application provisioning techniques that can be extended with ease and limited efforts [16]. According to the implementation using Cloudsim, the VMs are considered the cloud resources and Cloudlets as tasks/jobs. We have measured the make-span of different algorithms by considering varying the cloudlets while VMs are fixed, as well as, varying VMs.

Cloudlets were generated from a standard formatted workload of a High performance computing center called HPC2N in Sweden as a benchmark. According to this benchmark, Each row in the workload represents a cloudlet where we get the id of the cloudlet from the first column, the length of the cloudlet from the fourth column (the runtime value multiplied by the rating which is defined as 1 MI in CloudSim), and finally the number of the requested processing elements from the eighth column [17].

Experimental results

Two parameters have been used to measure the performance of the task scheduling algorithms; make-span, and utilization ratio. Make-span is the overall completion time needed to assign all the tasks to the available resources (i.e., VMs). Utilization ratio is the average amount of time in which the cloud resources are busy. It ranges from 0 to 1 [4].

Mak-span results

The experimental results of our proposed algorithm, PSO, GA, and FCFS algorithms are presented in Figures 2-4, using different VMs, as well as, different cloudlets (jobs/tasks). According to the experimental results in Figure 2 with considering 5 VMs, it is found that the performance of our proposed algorithm outperforms PSO,

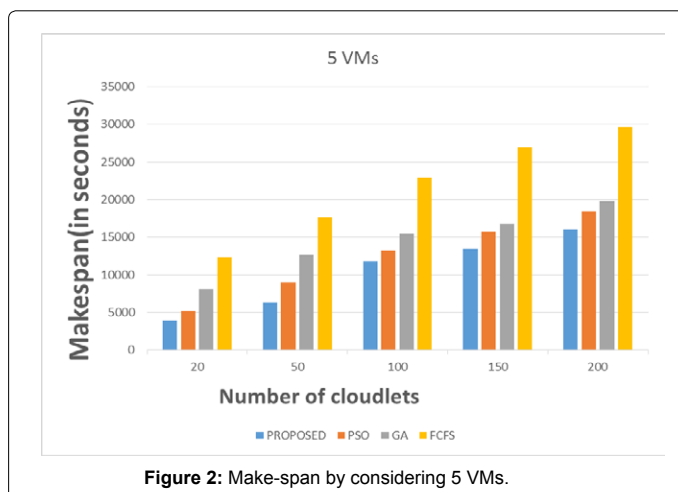


Figure 2: Make-span by considering 5 VMs.

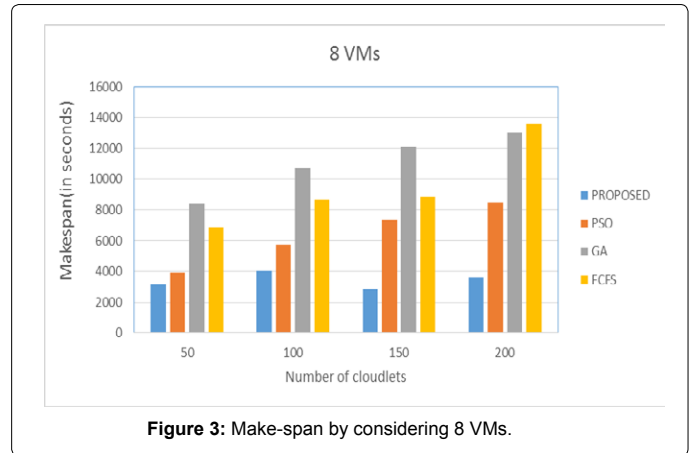


Figure 3: Make-span by considering 8 VMs.

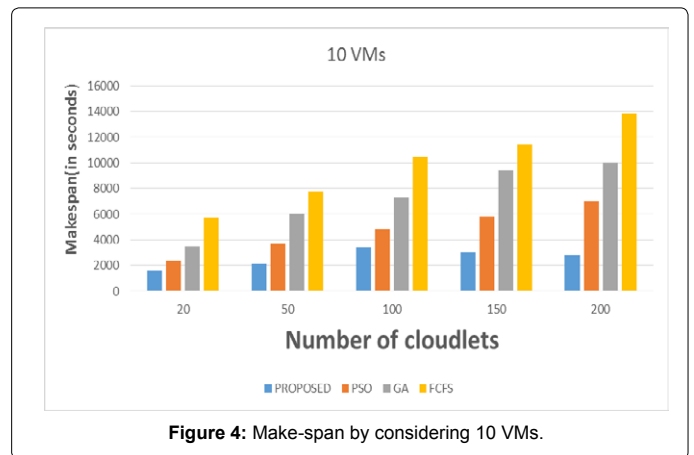


Figure 4: Make-span by considering 10 VMs.

GA, and FCFS algorithms with respect to the total execution time (i.e., makespan) by 59.68%, 71.1%, and 47.2% respectively. The overall performance of the proposed algorithm (i.e., average improvement) is 59.35% relative to the PSO, GA, and FCFS algorithms (Figure 2).

According to the experimental results in Figure 3 with considering 8 VMs, it is found that the performance of our proposed algorithm outperforms PSO, GA, and FCFS algorithms by 53.46%, 30.88%, and 35.97% respectively. The overall performance of the proposed algorithm performance is 40.1% relative to the PSO, GA, and FCFS algorithms (Figure 3).

By considering 10 VMs, the experimental results in Figure 4 proved that the performance of our proposed algorithm outperforms PSO, GA, and FCFS algorithms by 55%, 36%, and 26.46% respectively. The overall performance of the proposed algorithm performance is 39.15% relative to the PSO, GA, and FCFS algorithms.

Generally, by calculating the average performance of our proposed algorithm, PSO, GA, and FCFS algorithms, it is found that the proposed algorithm outperforms PSO, GA, and FCFS by 57.7%, 51.2%, and 39.89% respectively. And the overall performance of the proposed algorithm is 49.6% relative to PSO, GA, and FCFS algorithms (Figure 4).

Conclusion and Future Work

Task scheduling is one of the main issues in the Cloud Computing. Efficient task scheduling is essential for saving the time and utilizing the resources. In this paper, a task scheduling algorithm has been

proposed for the Cloud Computing environment. The main idea of the proposed algorithm is that allocating the available VMs to the requested tasks by considering the processing power of VMs and tasks. To evaluate the performance of the proposed algorithm, a comparative study has been done among our proposed algorithm, PSO, GA, and FCFS algorithms. The experimental results prove the efficiency of our proposed algorithm by minimizing make-span by 49.6% , in addition, increasing the utilization ratio. The proposed task scheduling algorithm could be further extended by considering the cost, as well as, the memory size. Also, the proposed algorithm could be modified by considering dependent tasks.

References

1. Soror AA, Minhas UF, Abounaga A, Salem K, Kokosielis P, et al. (2009) Deploying Database Appliances in the Cloud. *IEEE Data Eng Bull* 32: 13-20.
2. Jangra A, Saini T (2013) Scheduling Optimization in Cloud Computing. *International Journal of Advanced Research in Computer Science and Software Engineering* 3: 62-65.
3. Furht B, Escalante A (2010) *Handbook of Cloud Computing*. Springer Science+Business Media, LLC.
4. Kaur R, King S (2014) Analysis of Job Scheduling Algorithms in Cloud Computing. *International Journal of Computer Trends and Technology* 9: 379-386.
5. Pandaa SK, Guptab I, Janac PK (2015) Allocation-Aware Task Scheduling for Heterogeneous Multi-Cloud Systems. *Science Direct* 50: 176-184.
6. Yang Y, Liu K, Chen J, Liu X, Yaun D, et al. (2008) An Algorithm in SwinDeW-C for Scheduling Transaction- Intensive Cost-Constrained Cloud Workflows. *Proc. of 4th IEEE International Conference on e-Science*, Indianapolis, USA.
7. Pandey S, Wu L, Guru S, Buyya R (2010) A Particle Swarm Optimization (PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments. *Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, Perth, Australia.
8. Ke Liu, Hai Jin, Chen J, Liu X, Yuan D, et al. (2010) A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on a Cloud Computing Platform. *International Journal of High Performance Computing Applications* 24: 445-456.
9. Parsa S, Maleki RE (2009) RASA: A New Task Scheduling Algorithm in Grid Environment. *World Applied Sciences Journal (Special Issue of Computer and IT)* 152-160.
10. Huang J (2014) The Workflow Task Scheduling Algorithm Based on the GA Model in the Cloud Computing Environment. *Journal of Software* 9: 873-880.
11. Zhang L, Chen Y, Sun R, Jing S, Yang B (2008) A Task Scheduling Algorithm Based on PSO for Grid Computing. *International Journal of Computational Intelligence Research* 4: 37-43.
12. Lin C, Lu S (2011) Scheduling ScientificWorkflows Elastically for Cloud Computing. *IEEE 4th International Conference on Cloud Computing*, Washington.
13. Visalakshi P, Sivanandam SN (2009) Dynamic Task Scheduling with Load Balancing using Hybrid Particle Swarm Optimization. *International Journal of Open Problems in Computer Science and Mathematics, ICSRS Publication* 2: 476-488.
14. Selvarani S, Sadhasivam GS (2010) Improved Cost-based Algorithm for Task Scheduling in Cloud Computing. *IEEE International Conference, Computational Intelligence and Computing Research*, India.
15. Yang B, Xu X, Tan F, Parket DH (2011) An Utility- based Job Scheduling Algorithm for Cloud Computing Considering Reliability Factor. *Proceedings of the 2011 International Conference on Cloud and Service Computing*, IEEE Xplore Press, Hong Kong.
16. Singh S, Kalra M (2014) Task Scheduling Optimization of Independent Tasks in Cloud Computing Using Enhanced Genetic Algorithm. *International Journal of Application or Innovation in Engineering & Management* 3: 286-291.
17. U. University (2006) *The HPC2N Seth log*.