ISBN 978-93-84422-22-6
Proceedings of 2015 International Conference on Image Processing, Production and Computer Science (ICIPCS'2015)
Istanbul (Turkey), June 3-4, 2015 pp. 38-44

Task scheduling in the Cloud Environments based on an Artificial Bee Colony Algorithm

Nima Jafari Navimipour

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
Jafari@iaut.ac.ir

Abstract**:** *Nowadays, Cloud computing has gained vast attention due to its technological advancement, cost reduction and availability. In the Cloud environments, the suitable scheduling of the received tasks over service provides has become an important and vital problem. The scheduling problem in Cloud environments is an NP-hard problem. Therefore, many heuristics have been proposed to solve this problem up to now. In this paper, we propose a new bee colony algorithm to schedule the tasks on service providers in the Cloud environments. The results demonstrated that the proposed algorithm has a better operation in terms of task execution time, waiting time and missed tasks.*

Keywords: *Cloud computing, task scheduling, bee colony, service providers.*

1. Introduction

Cloud computing [1] is a novel infrastructure and very popular phenomenon that focuses on commercial resource provision and allows customers to utilized the computing resources presented by multiple service providers [2, 3]. These infrastructures are represented by services that are not only used but also installed, deployed or replicated with the help of virtualization [4]. It is a model of service delivery and access mechanism where virtualized resources are provided as a service over the Internet [5]. The main goal of Cloud computing is to provide on-demand computing services with high reliability, scalability, and availability in distributed environments [5]. It follows a pay-per-use model and can be dynamically reconfigured to satisfy user requests via on-the-fly virtual resources [6]. In the Cloud computing, several different forms of virtualized resources are dispersed across many locations and networks. Four forms of such virtualized resources are usually well-known: Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Expert as a Service (EaaS). SaaS delivers special-purpose software that is remotely accessible by consumers through the Internet with a usage-based pricing model. IaaS provides hardware, software, and equipment to provide software application environments, again with a usage-based pricing model. PaaS offers a high-level integrated environment to build, test, and deploy custom applications [7]. EaaS provides the transparent access to expert, knowledge and skills of human resources remotely over the Internet [8].

Task scheduler as an NP-Hard problem [9] is an important part of any distributed system like Grid [9-12], Cloud [13-18] and P2P networks [19-23] which assigns jobs to suitable resources for execution. The goal of job scheduler is to minimize the overall execution time of a collection of jobs [24]. The aim of task scheduler in the Cloud environments is to shorten the response time and enhance the service providers' utilization. Due to the nature of scheduling problem (NP-Hard), many heuristics have been proposed to solve this problem up to now. As another hand, artificial bee colony algorithm is a swarm-based optimization technique proposed for solving continuous optimization problems [25]. It has been applied to solve many problems and obtained interesting results [26]. But, to the best of our knowledge, no research of its applications for task scheduling in the Cloud environments has been done up to now. Therefore, in this paper, we propose a task scheduler mechanism to provide the stated aims using an artificial bee colony algorithm.

The rest of the paper is organized as follows: In section 2, the related works and artificial bee colony algorithm are briefly reviewed. In section 3, we propose an artificial bee colony algorithm to schedule the jobs as well as the related algorithms. In section 4 the obtained results are presented. Lastly, conclusion and future works are provided in section 5.

2. Related Work and Background

This section provides a brief overview of related researches and background of the paper, including task scheduling in the Cloud environments and artificial bee colony algorithms.

2.1. Task Scheduling in the Cloud Environments

The scheduling issue in the Cloud environments is more complicated than that in traditional parallel systems, because Cloud systems can have heterogeneity, dynamicity, intermittent presence, and large communication overhead characteristics. In this sub-section, we briefly analysis and summarize the state of the art methods and approaches in the field of task scheduling in the Cloud environments.

Garg, et al. [27] have proposed the near-optimal scheduling policies to exploit heterogeneity across multiple data centers for a Cloud provider. The method is considered a number of energy efficiency factors (such as energy cost, carbon emission rate, workload, and CPU power efficiency) which change across different data centers depending on their location, architectural design, and management system. The have demonstrated that the proposed method can able to achieve on average up to 25% of energy savings in comparison to profit based scheduling policies.

Abrishami and Naghibzadeh [28] have proposed a new QoS-based workflow scheduling algorithm based on a novel concept called Partial Critical Paths (PCP), which tries to minimize the cost of workflow execution while meeting a user-defined deadline. The proposed algorithm recursively schedules the PCP ends at previously scheduled tasks. The results demonstrated that the computation time of the proposed algorithm is low for the cost decreasing and the fair policies, but is much longer for the optimized policy.

A scheduling algorithm to address the major challenges of task scheduling in the Cloud environments is proposed by Choudhary and Peddoju [29]. In this algorithm, the incoming tasks are grouped on the basis of the task requirement like minimum execution time or minimum cost and prioritized. Then, resource selection is done on the basis of task constraints using a greedy approach.

A four-tier architecture for multilingual information resources scheduling in the Cloud environments is proposed by Han and Luo [30] in 2013. It includes user accessing tier, technology supporting tier, resource scheduling tier and resources tier. They proposed a three- layer scheduling model for multilingual information resources in the Cloud computing. The model includes some home managers, some local scheduling agents and a global scheduling agent. The example showed that the proposed scheduling model could improve the performance of the scheduling of multilingual information resources in the Cloud environments.

Kim, et al. [31] have optimized the job scheduling using biogeography-based optimization (BBO). BBO migration is used to change existing solutions and to adapt new good solutions. BBO offers the advantage of adaptive process, which is developed for binary integer job scheduling problem in cloud computing. Experimental results showed that the performance of the proposed methods are better than the considered other methods in the job scheduling problems.

Zeng, et al. [32] have proposed a model which considers data management to obtain satisfactory makespan on multiple data centres. At the same time, their adaptive data-dependency analysis can reveal parallelization opportunities. They introduces an adaptive strategy for workflow applications. It consist of a *set-up* stage which builds the clusters for the workflow tasks and datasets, and a *run-time* stage which makes the overlapped execution for the workflows. Through rigorous performance evaluation studies, they demonstrated that the proposed method can effectively improve the workflow completion time and utilization of resources in a Cloud environment.

Finally, Jafari Navimipour and Sharifi Milani [33] have proposed a new algorithm based on cuckoo search algorithm to schedule the tasks in Cloud computing which is based on the obligate brood parasitic behavior of some cuckoo species in combination with the Lévy flight behavior of some birds and fruit flies. The simulation results demonstrated that when the value of Pa is low, the speed and coverage of the algorithm become very high.

2.2. Artificial Bee Colony Algorithm

The artificial bee colony algorithm as a novel meta-heuristic approach which was developed in 2005 [34]. It is motivated by the intelligent foraging behavior of honey bees and is the simulation of the minimalistic foraging model of honey bee in search process for solving real-parameter, non-convex, and non-smooth optimization problems [35]. In this algorithm, a colony consists of three types of artificial bees including employed, onlooker and scout bees [36].

Employed bee: Each food source is associated with an employed bee and each employed bee tries to detect a new food source in the neighborhood of its current food source. The detected food source is memorized when the nectar amount of the detected food source is higher than the nectar amount of current food source. After completion of the search process, employed bees share their information concerning the nectar amount of food sources with onlooker bees via waggle dance in the dance area.

Onlooker bee: They watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbor around that, she evaluates its nectar amount. An onlooker bee evaluates the information gained from the employed bees and tries to find a new food source in the neighborhood of the selected food based on this evaluated information. Thus, the tendency of onlooker bees is to search around the food sources with high nectar amount; in this way, more qualified food sources can be chosen for exploitation.

Scout bee: The number of scout bees is not predefined in the colony. A scout bee is produced according to the situation of a food source whether it is abandoned or not. Scout bees are translated from a few employed bees, which abandon their food sources through a predetermined number of cycles [37]. The number of scout bees is controlled by the parameter "limit" representing the number of trials before defining a food source as "abandoned". If a food source cannot be improved during the predetermined number of unsuccessful trials being equal to limit parameter, that food source is leaved and a new food source is randomly generated. Exploration and exploitation procedures are carried out together for robust search process in this module [36].

In the artificial bee colony model, a food source denotes a possible solution of an optimization problem and the nectar amount of a food source represents the quality of a solution. As mentioned before, each food source is associated with an employed bee in the colony. Thus, the number of food sources is equal to the number of employed bees. Also, the number of employed bees is equal to the number of onlooker bees. Therefore, the number of food sources is half of the population number.

3. Proposed Method

In the proposed method, the position of a food source represents a possible solution to the scheduling problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution for given scheduling problem. The number of the employed bees is equal to the number of solutions in the population. At the first step, a randomly distributed initial population is generated. After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively. An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise she keeps the position of the one in her memory. After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts [38].

After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively. An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise she keeps the position of the one in her memory. After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts. The steps of the algorithm are:

- Produce initial population.
- While stop criteria is not satisfied, perform steps 3 to steps 6.
- Send the employed bees onto their food sources.
- Send the onlooker bees onto the food sources depending on their nectar amounts.
- Send the scout bees to search possible new food sources.
- Memorize the best food source found so far.

We consider *t* tasks to be processed on *c* Cloud nodes. There are some assumptions and constrains as follows: each task has predefined number of operations and a known determined sequence among these operations; each Cloud node is ready at zero time; each task must be processed on one machine at a given time; and each Cloud node can process a new operation only after completing the predecessor operation. Let Ct_i be the completion time of task T_i. Then, the completion time (makespan) is:

$$F = \max \{Cti \mid i = 1, \dots, n\} \tag{1}$$

3.1. Initialization Phase

In this phase, new candidate solutions are produced for each employed bees. At the beginning, the initial *Bs* scout bees are placed randomly in Cloud nodes and *Bs* is the number of scout bees. With the help of the randomization and the insertion-based local search, the initial population is constructed with both diversity and quality.

3.2. Employed Bee Phase

To simplify the process of algorithm, the number of employed bees is set to the same as that of food sources, and each employed bee is applied to each solution in the population, respectively. Also, an employed bee generates a food source in the neighborhood of the current source.

3.3. Onlooker Bee Phase

We firstly provide a combined local search which combines not only the insertion-based local search but also a swap-based local search. Each onlooker bee chooses an employed bee in order to improve its solution. This selection is done according to fitness values of employed bees by roulette wheel ($\frac{Fi}{\sum_{j=1}^{n} fj}$).

3.4. Scout Bee Phase

The abandonment counters of all employed bees are tested. The employed bee, which cannot improve self-solution until the abandonment counter reaches to the limit, becomes scout bee. The scout bee becomes the employed bee. Therefore, scout bees prevent inactivity of employed bee population. So, scout bees with the highest fitness are chosen as “Selected Bee” and the visited sites by them are chosen from neighborhood of Cloud nodes.

4. Experimental Results

In order to test the performance of the proposed method, we use a standard evaluation technique in the Cloud environments. In this section, we investigate whether our approach is effective or not. We present the details of extensive experiments to evaluate the performance of the approach. The simulation is performed using CPU core i5, memory 4GByte and 4MByte cache memory. The results have been obtained with a system simulator programmed in Matlab R2013b[1]. As another hand, generating random DAGs, allows us to evaluate different application graphs. We had evaluated the performance of the proposed algorithm under different

[1] www.mathworks.com

parameters, including different numbers of subtasks. The proposed algorithm is compared with other algorithms in terms of makespan. The number of generated subtasks in a DAG is selected from 1 to 20. The computational and communication costs of the DAG are generated randomly from a range. Fig. 2 shows the makespan value of the proposed algorithm in comparison to Palaniswami and Kumar [39] and El-Sisi and Tawfeek [40] using 20 diferent task graphs.

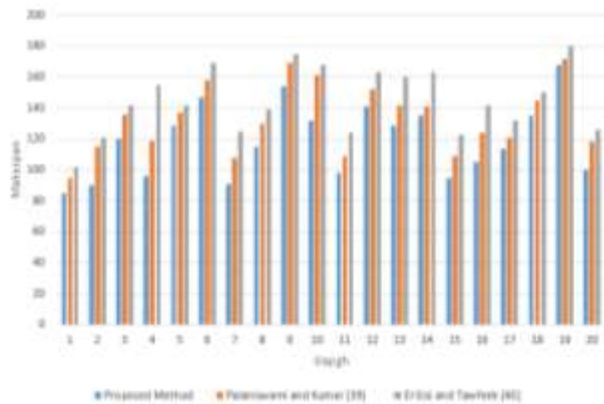

Fig. 2: the makespan value of the proposed algorithm in comparison to Palaniswami and Kumar [39] and El-Sisi and Tawfeek [40]

As we can saw in fig. 2, the makespan of the proposed method is lower than Palaniswami and Kumar [39] and El-Sisi and Tawfeek [40] in different situation.

5. Conclusion and Future Works

Cloud computing is a novel infrastructure and very popular phenomenon that focuses on commercial resource provision and allows customers to utilized the computing resources hosted by multiple service providers. Task scheduling is one of the important issue in the Cloud environments which is received many attentions nowadays because of its high computation cost. Therefore, many heuristics have been proposed to solve this problem up to now. Due to the advantageous of an artificial bee colony algorithm, in this paper, we propose a new artificial bee colony algorithm to schedule the tasks on service providers in the Cloud environments. The results demonstrated that the proposed algorithm has a better operation in terms of task execution time and waiting time in comparison to Palaniswami and Kumar [39] and El-Sisi and Tawfeek [40].

6. References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, vol. 25, pp. 599-616, 6// 2009.

[2] S. Ravimaran and M. A. Maluk Mohamed, "Integrated Obj_FedRep: Evaluation of Surrogate Object based Mobile Cloud System for Federation, Replica and Data Management," Arabian Journal for Science and Engineering, pp. 1-16, 2014/03/29 2014.

[3] Y.-T. Lin, M.-L. Wen, M. Jou, and D.-W. Wu, "A cloud-based learning environment for developing student reflection abilities," Computers in Human Behavior, vol. 32, pp. 244-252, 3// 2014.

[4] A. Kertesz, G. Kecskemeti, and I. Brandic, "An interoperable and self-adaptive approach for SLA-based service virtualization in heterogeneous Cloud environments," Future Generation Computer Systems.

[5] B. Rimal, A. Jukan, D. Katsaros, and Y. Goeleven, "Architectural Requirements for Cloud Computing Systems: An Enterprise Cloud Approach," Journal of Grid Computing, vol. 9, pp. 3-26, 2011/03/01 2011.

[6] N. Jafari Navimipour, A. M. Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Expert Cloud: A Cloud-based framework to share the knowledge and skills of human resources," Computers in Human Behavior, vol. 46, pp. 57-74, 2015.

http://dx.doi.org/10.1016/j.chb.2015.01.001

[7] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," in 2008 Grid Computing Environments Workshop, Austin, TX, USA, 2008, pp. 1-10.

[8] N. J. Navimipour, A. H. Navin, A. M. Rahmani, and M. Hosseinzadeh, "Behavioral modeling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources," Computers in Industry, 2015.

http://dx.doi.org/10.1016/j.compind.2014.12.007

[9] N. Jafari Navimipour and L. Mohammad Khanli, "The LGR method for task scheduling in computational grid," in Advanced Computer Theory and Engineering, 2008. ICACTE'08. International Conference on, 2008, pp. 1062-1066.

http://dx.doi.org/10.1109/icacte.2008.24

[10] A. Habibizad Navin, N. Jafari Navimipour, A. M. Rahmani, and M. hosseinzadeh, "Expert Grid: New Type of Grid to Manage the Human Resources and Study the Effectiveness of its Task Scheduler," Arabian Journal for Science and Engineering, 2014.

[11] Q. Yang, H. Wang, W. Hu, and W. Lijuan, "A scheduling model for temporally constrained grid workflow for distributed simulation system on grid," Kybernetes, vol. 39, pp. 1344-1350, 2010.

http://dx.doi.org/10.1108/03684921011063646

[12] N. Jafari Navimipour, A. Masoud Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Resource discovery mechanisms in grid systems: A survey," Journal of Network and Computer Applications, vol. 41, pp. 389-410, 2014.

http://dx.doi.org/10.1016/j.jnca.2013.09.013

[13] Y. Laili, F. Tao, L. Zhang, Y. Cheng, Y. Luo, and B. R. Sarker, "A Ranking Chaos Algorithm for dual scheduling of cloud service and computing resource in private cloud," Computers in Industry, vol. 64, pp. 448-463, 5// 2013.

[14] M. Mezmaz, N. Melab, Y. Kessaci, Y. C. Lee, E. G. Talbi, A. Y. Zomaya, et al., "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," Journal of Parallel and Distributed Computing, vol. 71, pp. 1497-1508, 11// 2011.

[15] S. Pandey, W. Linlin, S. M. Guru, and R. Buyya, "A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, 2010, pp. 400-407.

http://dx.doi.org/10.1109/aina.2010.31

[16] F. Tao, Y. Feng, L. Zhang, and T. W. Liao, "CLPS-GA: A case library and Pareto solution-based hybrid genetic algorithm for energy-aware cloud service scheduling," Applied Soft Computing, vol. 19, pp. 264-279, 6// 2014.

[17] W. Wang, G. Zeng, D. Tang, and J. Yao, "Cloud-DLS: Dynamic trusted scheduling for Cloud computing," Expert Systems with Applications, vol. 39, pp. 2321-2329, 2/15/ 2012.

[18] Z. Wu, X. Liu, Z. Ni, D. Yuan, and Y. Yang, "A market-oriented hierarchical scheduling strategy in cloud workflow systems," The Journal of Supercomputing, vol. 63, pp. 256-293, 2013/01/01 2013.

[19] A. Montazeri, B. Akbari, and M. Ghanbari, "An incentive scheduling mechanism for peer-to-peer video streaming," Peer-to-Peer Networking and Applications, vol. 5, pp. 257-278, 2012.

http://dx.doi.org/10.1007/s12083-011-0121-7

[20] D. C. Erdil, "Simulating peer-to-peer cloud resource scheduling," Peer-to-Peer Networking and Applications, vol. 5, pp. 219-230, 2012/09/01 2012.

[21] F. Luo, H. Jin, X. Liao, and Q. Zhang, "Scheduling in an unstructured peer-to-peer-based high performance computing system," Kybernetes, vol. 41, pp. 1209-1215, 2012.

http://dx.doi.org/10.1108/03684921211275225

[22] J. Rius, F. Cores, and F. Solsona, "Cooperative scheduling mechanism for large-scale peer-to-peer computing systems," Journal of Network and Computer Applications, vol. 36, pp. 1620-1631, 11// 2013.

[23] N. Jafari Navimipour and F. Sharifi Milani, "A comprehensive study of the resource discovery techniques in Peer-to-Peer networks," Peer-to-Peer Networking and Applications, pp. 1-19, 2015.

[24] N. Jafari Navimipour, A. Masoud Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Job scheduling in the Expert Cloud based on genetic algorithms," Kybernetes, vol. 43, pp. 1262-1275, 2014.

http://dx.doi.org/10.1108/K-02-2013-0018

[25] M. S. Kiran, H. Hakli, M. Gunduz, and H. Uguz, "Artificial bee colony algorithm with variable search strategy for continuous optimization," Information Sciences, vol. 300, pp. 140-157, 4/10/ 2015.

[26] P. Fränti and O. Virmajoki, "Iterative shrinking method for clustering problems," Pattern Recognition, vol. 39, pp. 761-775, 2006.

http://dx.doi.org/10.1016/j.patcog.2005.09.012

[27] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya, "Environment-conscious scheduling of HPC applications on distributed Cloud-oriented data centers," Journal of Parallel and Distributed Computing, vol. 71, pp. 732-749, 6// 2011.

[28] S. Abrishami and M. Naghibzadeh, "Deadline-constrained workflow scheduling in software as a service Cloud," Scientia Iranica, vol. 19, pp. 680-689, 6// 2012.

[29] M. Choudhary and S. K. Peddoju, "A dynamic optimization algorithm for task scheduling in cloud environment," International Journal of Engineering Research and Applications (IJERA), vol. 2, pp. 2564-2568, 2012.

[30] Y. Han and X. Luo, "Hierarchical Scheduling Mechanisms for Multilingual Information Resources in Cloud Computing," AASRI Procedia, vol. 5, pp. 268-273, // 2013.

[31] S.-S. Kim, J.-H. Byeon, H. Yu, and H. Liu, "Biogeography-based optimization for optimal job scheduling in cloud computing," Applied Mathematics and Computation, vol. 247, pp. 266-280, 11/15/ 2014.

[32] L. Zeng, B. Veeravalli, and A. Y. Zomaya, "An integrated task computation and data management scheduling strategy for workflow applications in cloud environments," Journal of Network and Computer Applications, vol. 50, pp. 39-48, 4// 2015.

[33] N. Jafari Navimipour and F. Sharifi Milani, "Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm," International Journal of Modeling and Optimization, vol. 5, 2015.

[34] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department2005.

[35] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of global optimization, vol. 39, pp. 459-471, 2007.

http://dx.doi.org/10.1007/s10898-007-9149-x

[36] C. Ozturk, E. Hancer, and D. Karaboga, "A novel binary artificial bee colony algorithm based on genetic operators," Information Sciences, vol. 297, pp. 154-170, 2015.

http://dx.doi.org/10.1016/j.ins.2014.10.060

[37] C. Zhang, J. Zheng, and Y. Zhou, "Two modified Artificial Bee Colony algorithms inspired by Grenade Explosion Method," Neurocomputing, vol. 151, pp. 1198-1207, 2015.

http://dx.doi.org/10.1016/j.neucom.2014.04.082

[38] P. Mansouri, B. Asady, and N. Gupta, "The Bisection–Artificial Bee Colony algorithm to solve Fixed point problems," Applied Soft Computing, vol. 26, pp. 143-148, 2015.

http://dx.doi.org/10.1016/j.asoc.2014.09.001

[39] S. Palaniswami and V. Kumar, "A Dynamic Resource Allocation Method for Parallel DataProcessing in Cloud Computing," Journal of computer science, vol. 8, pp. 780-788, 2012.

http://dx.doi.org/10.3844/jcssp.2012.780.788

[40] A. B. El-Sisi and M. A. Tawfeek, "Cloud Task Scheduling for Load Balancing based on Intelligent Strategy," International Journal of Intelligent Systems and Applications (IJISA), vol. 6, p. 25, 2014.

http://dx.doi.org/10.5815/ijisa.2014.05.02