JACET Journal of Advances in Computer Engineering and Technology, 1(4) 2015

TASA: A New Task Scheduling Algorithm in Cloud Computing

Somayeh Taherian Dehkordi[1] , Vahid Khatibi Bardsiri[2]

Received (2015-07-11)
Accepted (2015-10-17)

Abstract — **Cloud computing refers to services that run in a distributed network and are accessible through common internet protocols. It merges a lot of physical resources and offers them to users as services according to service level agreement. Therefore, resource management alongside with task tim has direct influence on cloud networks' performance and efficiency. Presenting a proper scheduling method can lead to efficiency of resources by decreasing response time and costs. This paper studies the existing approaches of task scheduling and resource allocation in cloud infrastructures and assessment of their advantages and disadvantages. Afterwards, a compound algorithm is presented in order to allocate tasks to resources properly and decrease runtime. The proposed algorithm is built according to conditions of compounding Min-min and Sufferage algorithms. In the proposed algorithm, task allocation between machines takes place alternatively and with continuous change of scheduling algorithms. The main idea of the proposed algorithm is to concentrate on the number of tasks instead of the existing resources. The simulation results reveal that the proposed algorithm can achieve higher performance in decreasing response time.**

Index Terms - **Cloud computing, Task scheduling algorithm, Min-Min, Sufferage.**

I. INTRODUCTION

Cloud computing means offering any kind of software or hardware services through Internet space that obey the rules of sufficient payment. The main purpose behind using this technology is to decrease costs and increase efficiency and performance [1]. The main point is how machines and resources are managed beside task and inquiries scheduling policies. Appropriate scheduling makes it possible to manage a large number of inquiries with a specific volume of resources which leads to efficient resources [2]. Different algorithms have been proposed for task scheduling in the cloud environment, that each has their advantages and disadvantages. In an inappropriate scheduling for a sequence of requests may involve a large number of machines, while the optimal scheduling could provide that result with less machines and better management. Therefore, providing an appropriate scheduling would decrease reaction time of tasks, costs and efficiency of virtual machines. In this paper, a new scheduling algorithm is proposed that is more efficient than previous similar algorithms.

According to the studies[3][4][5] it can be concluded that the Sufferage algorithm is suitable for cloud medium and its implementation shows relatively good results of scheduled work in the cloud medium. In addition, based on the studies[5][6], one of the best scheduling algorithms that can be implemented in the cloud computing environment is combination of Min-min algorithm and Max-min algorithm as RASA, an algorithm is aware of the resource that has the minimal delay. Although, the mentioned algorithm has shown acceptable performance, but the high level of inconsistency and diversity

1- Department of Computer Engineering. Kerman Branch. Islamic Azad University. Kerman. Iran (S.taherian2000@gmail.com).

2- Department of Computer Engineering. Kerman Branch. Islamic Azad University. Kerman. Iran (kvahid2@live.utm.my).

of requests in a cloud environment could limit the scope of this algorithm and can make some problems in real environments.

on changes in the structure of this algorithm and combining with Sufferage algorithm can optimize performing parameters in this area. Study the literatures indicate that the desired combination has no background and can be considered as a new method.

Since Sufferage algorithm in previous studies had a better performance than Max-min algorithm[2][8][9], in proposed algorithm has been tried to provide an appropriate combination of Min-min and Sufferage algorithm for task scheduling. In fact, the combined proposed algorithm is a new aware algorithm of tasks that attempt to consider the advantages and disadvantages of previous techniques and scheduling, produce a less average response time.

This article studies the most common task scheduling algorithms in cloud computing. After reviewing cloud computing and its common scheduling algorithms, a compound method using Min-min and Sufferage algorithms is presented to decrease runtime.

The article is organized into five sections. Section two expresses tasks associated to scheduling. In section three, Min-min and Sufferage algorithms are explained. Then, details of the proposed algorithm are given. Assessing performance of the proposed algorithm and comparison of it with RASA, Min-min and Sufferage algorithms is brought in section four. Section five contains the conclusions.

II. RELATED WORK

Cloud computing refers to a kind of distributed and parallel systems that include a set of virtual computers that are connected to each other. In fact, cloud computing is a computer model that tries to facilitate users' access according to their request of information and calculation resources. This model tries to respond to user needs by the least need to human resources and decreasing costs and increasing time of access time to information [7].

Today, increased efficiency of clouds is undeniable. Proper and optimal scheduling leads to increased efficiency and performance of cloud computing. The dynamic and changeable nature in computing and users' varied inquiries makes scheduling in clouded environments very complicated. Optimum scheduling of the existing tasks is one of the main challenges in heterogeneous calculative systems and more specifically in cloud processing.

In general, the following hypothesis can be used for task scheduling in cloud computing:

Suppose that there m number of machines (resources): VMj(j=0,…,m) and these resources have to process n number of tasks: Ti (i=0,…,n), in this case a scheduler for task i is the allocation of one or more resources in different time lapse for that task. Task scheduling in cloud computing means doing n tasks on m resources and in this problem the most optimal case of doing these n tasks on m resources must be considered so that the total final task achievement time is minimized [8].

In general, variables which are defined for scheduling in cloud computing are shown in table1.

TABLE I
SCHEDULING VARIABLES IN THE CLOUD COMPUTING

Variable	Description
Ti	task i
Rj	resource j
Eij	expected execution time of Ti on Rj
rj	ready time of Rj
Cij=Eij + rj	Completion time of task i on resource j

Two types of scheduling algorithms are surveyed in the following: static and dynamic. Static scheduling in fact indicates the allocation of processes to processors within the compile time or sooner. Dynamic scheduling means that processes are allocated to processors at the beginning of running (allocation while running) and the allocation may be performed again during the implementation. The other difference is that the static scheduler makes decisions only based on process information and static system, while dynamic scheduler considers the current status of the system as well. In cloud networks, either of the above methods can be adopted based on the requirements of and the major goal which is optimization [9]. The purpose of most scheduling algorithms is to decrease makespan, i.e. implementing tasks is supposed to end as soon as possible.

Table2 shows a number of scheduling algorithms in cloud computing environment and Table3 shows some scheduling algorithms of cloud computing environment beside their goals advantages and disadvantages.

TABΛE II
VARIOUS SCHEDULING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT

Algorithm name	Algorithm type	Description
FCFS	**Dynamic**	FCFS is used for Distributed systems such as grids or Cloud computing, and parallel scheduling, and is aiming at the resources with the smallest waiting queue time and is selected for the incoming task. The idea of the FCFS (Frist Come First Service) algorithm in distributed network, was first proposed by Brent and was used by Schwiegelshohn in Grid and tasks scheduling in Cloud computing. This algorithm assigns tasks in order of their arrival to the resource. The purpose of this algorithm is justice [10].
OLB	**Static / Dynamic**	Opportunistic load balancing is famous algorithms and was suggested by Freund and allocated for a particular resource in the future, regardless of the performance of the machine, available and ready [11].
MET	**Static / Dynamic**	Minimum Execution Time algorithm was proposed by Armstrong and recommendation of this algorithm is given best resource to each task which can cause load imbalance between the resources [12].
MCT	**Static / Dynamic**	Minimum Completion Time algorithm is a hybrid algorithm proposed by Michalewicz. This algorithm is composition of two MET and OLB algorithms and has considered the advantages of MET and OLB algorithms and had covered their disadvantages. Each task is allocated to the resource with the shortest Completion time [11].
RR	**Dynamic**	Round Robin algorithm is a non-exclusive scheduler. In this algorithm, the scheduler allocates a fixed unit of time to each process which is called interrupt and then circulates among them. The combination of this algorithm and FCFS algorithm is used in task scheduling in cloud environment. In fact, by achieving to the desired interval, the running task enters the ready queue and the next task is selected based on FCFS algorithm [10].
Max-min	**Dynamic**	In this algorithm, at first, a set of minimum times is selected for each task on the resource. Then the task which requires maximum time to be fulfilled is selected in the time set and is allocated to the related machine. The selected task is removed from the set of tasks and by adding its running time to other tasks on the selected resource, the running time of the other tasks will be updated [12].
RASA	**Static**	It is a hybrid algorithm proposed by Saeed Parsa and Reza Entezari-Maleki and examined the distribution and scalability of distributed systems including grid and Cloud computing. RASA (Resource Aware Scheduling algorithm) algorithm is analyzed by the analysis of two Min-Min and Max-min algorithms and has considered the advantages of Min-Min and Max-min algorithms and had covered their disadvantages. This algorithm takes into consideration a set of tasks. If the amount of tasks is an even number it uses Max-min method in order to allocate resources to tasks and if the amount of tasks is an odd number it uses Min-Min method [13].

RSDC	**Dynamic**	Arash Ghorbannia et al. presented an algorithm of reliable scheduling distributed in Cloud computing. In RSDC (Reliable Scheduling Distributed in Cloud computing) algorithm, a new algorithm is developed with a new technique and by classifying and considering the sweep time of tasks in competence function. Through careful examination and evaluation of previous algorithms, the tasks have been scheduled by parameters associated with a failure rate. Therefore, in the proposed algorithm in addition to previous applied parameters some other remarkable parameters are used according to which different tasks scheduling can be obtained. The task is associated with a mechanism. The large task is divided into small tasks. In order to balance the tasks, their sweep time has been calculated separately. Then all tasks scheduling has been fulfilled in the form of a common task by considering the sweep time and finally the system efficiency and productivity have increased and also its real time has improved compared to the previous algorithm. Ultimately, through the presented mechanism by the proposed algorithm, the total time of Cloud computing has optimized compared to that of previous algorithms [14].
Optimization PSO algorithm	**Dynamic**	It is a new algorithm proposed by Ramezani et al. and they enhanced tasks scheduling in Cloud computing by using Multi-Objective Particle Swarm Optimization algorithm. The objective of this algorithm is to have maximum utilization of resources and to reduce the execution time. In this algorithm, Master-Slave model was used for implementation. Master computer must divide Computational load between slaver computers and Slaver computers will wait to receive their full share of the load of the computational and then will begin their processing. Each Slaver computer must return the result to the Master computer after processing and according to the way specified in scheduling [15].
Optimization workflow scheduling using Ant Colony	**Dynamic**	It is an algorithm proposed by Elayaraja and Dhanasekar for Optimization workflow scheduling with Ant Colony. This algorithm aims to allocate the presented tasks to resources according to their processing capabilities. It is inspired with the social behavior of ants. Based on this algorithm, the quality of service depends on criteria like reliability, time and costs [16].
SA-G Algorithm	**Dynamic**	It is an algorithm proposed by Xu et al. for Optimization task scheduling in Cloud computing with combinatorial two heuristic algorithms, Simulated-Annealing algorithm and Genetic algorithm. The proposed method has two parts: The first part of the genetic algorithm that after stops last iteration and then all the points in the final product obtained with fitness, as input to the Simulated Annealing algorithm [17].

TABLE III
ADVANTAGES AND DISADVANTAGES OF SCHEDULING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT

Algorithm name	Advantages	disadvantages
FCFS	Simple, Simple in implementation, Justly, no starvation	High average waiting time for Entry task with delay, The procedures are very short end of the queue must wait for the queue to be run very large jobs
Max-min	Better makespan compared to other algorithms	Delays in the implementation of small
RR	Non-exclusive, Simple, no starvation, Justly, allocates a fixed unit of time to each process , Less complexity	Needed great accuracy in determining the timescales, Large overhead
RASA	Reduce makespan	Lack of attention to delay caused by the transfer of tasks

III. TASA ALGORITHM

The proposed algorithm of TASA (Task-aware scheduling algorithm), is a method combining optimal features of base Min-min and Sufferage algorithms with the parameter of decreasing total time in cloud computing environment which considers speed in finding the answer as well as offering the most appropriate one.

Min-min algorithm was proposed in order to minimize resource allocation time for the accomplishment of required tasks. This algorithm aims to decrease pass completion time by early allocation of resources to small tasks it is a dynamic algorithm which is based on minimum completion time and acts as follows. In each stage, a set of task/processor pairs that have the closest expected competition time are selected from tasks and the set of processors. Afterwards, the pair of task/processor which has the littlest expected completion time is selected and then allocated to the subject processor [18].

In the Sufferage algorithm, the least and the one but the least time of task completion is found for each task. Then, the difference between these two values is defined as the expected value. The second step includes allocating the task with maximum value to the machine associated with the least time. The expectation method is based on the idea that solutions can be generated by allocation of a machine to the task which otherwise would have to wait to take over a machine [19].

The proposed algorithm in cloud computing environment is shown in figure 1. Matrix C is formed based on Cij i.e. allocation of task Ti to resource Rj. If the number of system tasks is even, Sufferage strategy is adopted and otherwise, Min-min method is used. In each allocation, as the task is removed from algorithm cycle, task allocation strategy changes rotatory. As an example, if there are three tasks in the system, primarily Min-min method is adopted and the task which has taken the resource is removed from task cycle. In the second step, Sufferage strategy is used and the selectedtask is removed as the resource is allocated to the task.

The third step uses Min-min method. Since Sufferage algorithm had better performance than Max-min algorithm in previous studies [2] [8][9][20] the proposed method tries to present a proper combination of Sufferage and Min-min Algorithms.

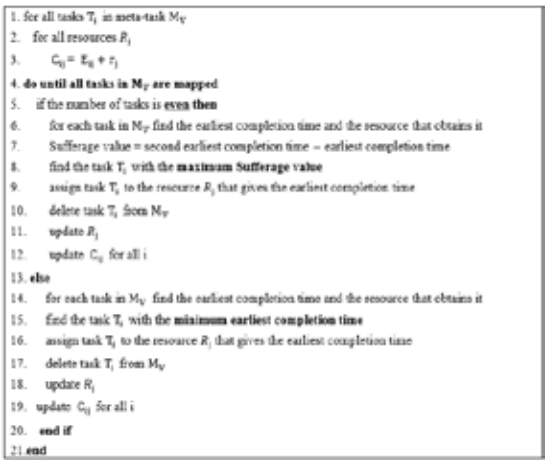

Fig.1. The TASA algorithm.

IV. IMPLEMENTATION AND SIMULATION OF PROPOSED ALGORITHM CLOUDSIM

Cloudism is a simulation software application in cloud environment which has been implemented by Gridbus project team and Grid Lab of Melbourne University. Cloudism can run on Windows and Linux systems and therefore is a Cross-platform tool. This simulator is in fact a Java Library in which there are useful functions for simulation. The newest open-source platform simulation version of Cloudism is Cloudism3.0.

In the simulation environment of Cloudism, after producing a virtual cloud with three machines, Makespan value for the each task is obtained for Min-min, Sufferage and RASA methods. As mentioned earlier, the goal parameter in the proposed algorithm is to decrease Makespan.

The proposed algorithm and Min-min, Sufferage and RASA algorithms are implemented by the following assumptions in the simulator environment:

- Eight number of tasks: T0, T1, T2, T3, T4, T5, T6 and T7 are in meta-tasks (n=8)
- Three number of resources R0, R1 and R2 (m=3)

Table 4 and table 5 show machine and task features based on which the runtime of each task on each machine is calculable. Comparison of the results from the proposed algorithm and existing ones are brought in figures 2 and 3.

Virtual machines' speed and bandwidth are shown in table4. According to the table, maximum machine speed is 200 and its minimum speed is 50. The most of the bandwidth is related to R1

virtual machine and the minimum bandwidth goes to the third virtual machine R2.

TABLE IV
RESOURCES SPECIFICATION

Resources	Processing speed (MIPS)	Related bandwidth (Mbps)
R0	100	100
R1	50	250
R2	200	50

The volume of syntaxes according to the number of instructions and the required extent of data is given in table 5. As shown in the table, the checked tasks are at most 312 million and at least 21 million instructions. The most data volume related to an instruction is 95 Megabytes.

TABLE V
TASK SPECIFICATION

Task	Volume instructions (MI)	Volume of data (Mb)
T0	256	44
T1	50	95
T2	128	64
T3	69	36
T4	218	59
T5	312	47
T6	21	19
T7	200	51

Runtime for each task on each machine is calculable through formula 1.

$$Eij = [(MI/MIPS) + (Mb/Mbps)] \quad \text{(Formula1)}$$

Total runtime based on the calculated data is given in table 6.

TABLE VI
MAKESPAN FOR THREE RESOURCES & EIGHT TASKS

Algorithm	Makespan
TASA	**4.43**
Min-min	5.83
Sufferage	4.55
RASA	6.67

The number of existing tasks is boosted up to 500 tasks for deeper analysis of the proposed method and the results are illustrated in figure 2.

As it can be seen, the proposed method has given better values for Makespan parameter for 8, 50, 20, 150, 300 and 500 tasks. Bearing in mind that user's requested tasks in cloud environment are increasing, as the number of tasks increases, the proposed algorithm will have more improvements. As well, if the number of requested tasks is small and is rather close to the number of resources, this algorithm will have a performance like that of Sufferage algorithm but yet will have better performance than RASA and Min-min algorithms.

The reason for such an improvement can be found in using Sufferage and Min-min algorithms together because using these two algorithms, which alone have short implementation time, leads to better task runtime in cloud environment.

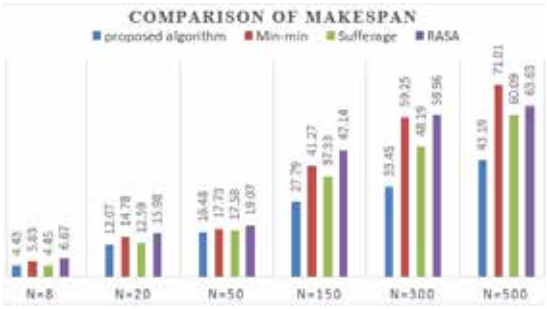

Fig. 2. Makespans for Fixed resources and Varying tasks Makespans of the algorithms.

There is another comparison drawn in figure3 between the proposed method and the existing ones. In order to do the test and more accurately compare the algorithms, the proposed algorithm and Sufferage, Min-min and RASA algorithms

are checked 10 times as task number changed from 10 tasks to 1000 tasks. In order to check the impact of the m=number of virtual machines besides the input tasks, the virtual algorithm was tested for 5, 10, 20 and 30 machines.

The obtained results revealed that the number of virtual machines does not significantly impact on performance of the proposed algorithm. According to the figure, the proposed algorithm has performed better at allocating tasks to resources in comparison with the three other algorithms. In other words, by making the resource allocation rotatory, the proposed algorithm tries to decrease Makespan by better allocation of tasks to resources.

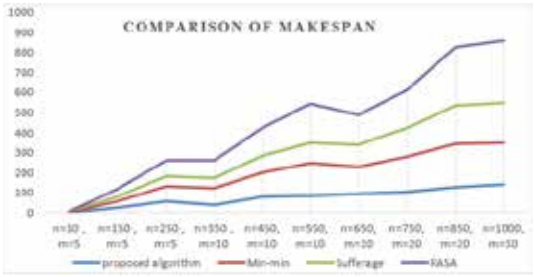

Fig. 3. Makespans for varying resources and Varying task Makespans of the algorithms

Table 7 includes percentage of improvements caused by the proposed algorithm in comparison with other algorithms for various tasks.

Accordingly, the proposed method has provided more efficiency than RASA, Min-min and Sufferage algorithms which is due to the combined and rotatory use of Min-min and Sufferage algorithms. By using these two algorithms together, in addition to their positive features, they acquire new features that lead to shorter total runtime and eventually decrease in Makespan in comparison with the existing algorithms. According to the table, when resources and tasks are small in number, total runtime deficit among the four algorithms seems trivial and algorithm improvement percentage is less, but as resources and tasks increase in number, this difference grows larger and more improvement is carried out.

TABLE VI
COMPARISON OF IMPROVEMENT PERCENT

Name / Number	Compared to the Min-min	Compared to the Sufferage	Compared to the RASA
8	24%	1%	34%
20	18%	4%	24%
50	8%	6%	14%
150	7%	26%	41%
300	44%	31%	44%
500	39%	28%	32%
average	**23%**	**16%**	**31%**

According to table 7 and diagrams 3 and 4, comparing various numbers of tasks, the proposed algorithm has achieved higher performance than Sufferage, Min-min and RASA algorithms and it is due to rotatory and combined use of Sufferage and Min-min algorithms together.

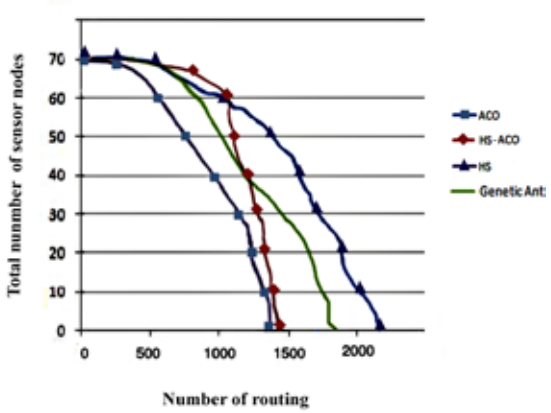

Fig. 3. The life trend of the entire network

V. CONCLUSION

Cloud submitters and cloud users each have different purposes. Submitters intend to increase output by reaching high efficiency levels of resources and users want to access their desired services with the fewest costs in the shortest time. Reaching such goals is challenging since cloud environment is heterogeneous and unpredictable.

Proper task scheduling mechanisms can fulfill users' needs and guarantee submitters' efficacy. In this article, a scheduling algorithm was proposed aiming to decrease total runtime. By combining Sufferage and Min-min algorithms and being inspired by the model presented in RASA algorithm and with the condition of considering the number of tasks instead of the number of resources, the proposed algorithm could cause significant improvement in Makespan parameter. Simulation results revealed that the proposed combined algorithm can offer acceptable and simultaneously comparable performance in large scales. As a future work, it will be attempted to make more improvement in the proposed algorithm which would be based on evolutionary algorithms.

REFERENCES

[1] F. Durao, S.F.J. Carvalho, A. Fonseka and C.V. Garcia, "A systematic review on cloud computing", The Journal of Supercomputing, Springer US, Vol. 68, 2014, pp.1321-1346.

[2] W. Mingxin, "Research on Improvement of Task Scheduling Algorithm in Cloud Computing", Applied Mathematics & Information Sciences An International Journal, Vol.9,2015, pp. 507-516.

[3] T. Ma, Y. Chu, L. Zhao, and O. Ankhbayar, "Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm", IETE Technical Review, Publishing models and article dates explained, Vol.31, 2014, pp.1-16.

[4] R. Kaur, and S. Kinger, "Analysis of Job Scheduling Algorithms in Cloud Computing" , International Journal of Computer Trends and Technology (IJCTT) , Vol.9, 2014, pp.379-386.

[5] S.V.Nandgaonkar, and A.B. Raut, "A Comprehensive Study on Cloud Computing", International Journal of Computer Science and Mobile Computing, a Monthly Journal of Computer Science and Information Technology,Vol.3,2014, pp.733-738.

[6] R. Mittal, and k. Soni, "Analysis of Cloud Computing Architectures", International Journal of Advanced Research in Computer and Communication Engineering,Vol.2, 2013, pp.2087-2091.

[7] T. Buchert, C. Ruiz, L. Nussbaum, And O. Richard, "A survey of general-purpose experiment management tools for distributed systems", Future Generation Computer Systems, Vol.45, 2015, pp. 1-12.

[8] T. Mathew, K.C. Sekaran, and J. Jose, "Study and analysis of various task scheduling algorithms in the cloud computing environment" , Advances in Computing, Communications and Informatics (ICACCI), International Conference, 2014, pp. 658- 664.

[9] S. Nagadevi, K. Satyapriya, and D. Malathy, "A Survey on Economic Cloud Schedulers for Optimized task scheduling", International Journal of Advanced Engineering Technology, Vol.5, 2013, pp. 58-62.

[10] L. Tripathy, and R.R. Patra, "Scheduling in Cloud Computing", International Journal on Cloud Computing: Services and Architecture (IJCCSA), Vol. 4, 2014, pp. 21-27.

[11] C.W. Tsai, and J.P.C. Rodrigues, "Metaheuristic Scheduling for Cloud: A Survey", Systems Journal, IEEE, Vol.8, 2013, pp. 279-291.

[12] R. Nallakumar, N. Sengottaiyan, and S.Nithya, "A Survey of Task Scheduling Methods in Cloud Computing". International Journal of Computer Sciences and Engineering, Vol.2, 2014, pp. 9-13.

[13] S. Parsa, and E.R. Maleki, "RASA: A New Task Scheduling Algorithm in Grid Environment", World Applied Sciences Journal 7 (Special Issue of Computer & IT), 2009, pp. 152-160.

[14] A. Ghorbannia, M. Javanmard, M. Barzegar, and M. Khosravi, "RSDC (Reliable Scheduling Distributed in Cloud Computing) ", International Journal of Computer Science, Engineering and Applications (IJCSEA). Vol.2, 2012, pp. 1-16.

[15] F. Ramezani, L. Jie, and J. Hussain, "Task Scheduling Optimization in Cloud Computing Applying Multi-Objective Particle Swarm Optimization", Springer-Verlag Berlin Heidelber, 2013, pp. 237-251.

[16] J. Elayaraja, and S. Dhanasekar, "A Survey on workflow scheduling in cloud using ant colony optimazation", International Journal of Computer Science and Mobile Computing, Vol.3, 2014, pp. 39- 44.

[17] X. Xu, N. Hu, and W.Q. Ying, "Cloud Task and Virtual Machine Allocation Strategy Based on Simulated Annealing-Genetic Algorithm", Applied Mechanics and Materials, Applied Science, Materials Science and Information Technologies in Industry, 2014, pp. 391-394.

[18] M.G. Huang, and Z.Q. Ou, "Review of Task Scheduling Algorithm Research in Cloud Computing", Advanced Materials Research, Progress in Applied Sciences, Engineering and Technology , 2014, pp. 3236-3239.

[19] k. Gupta, and M. Singh, "Heuristic Based Task Scheduling In Grid", International Journal of Engineering and Technology (IJET), Vol.4, 2014, pp. 254-260.

[20] H. Suo, H. Yan, "Research on Resource Scheduling in Cloud Computing: Issues and Solutions", Applied Mechanics and Materials, Numerical Methods, Computation Methods and Algorithms for Modeling, Simulation and Optimization, Data Mining and Data Processing, 2014, pp. 1801-1804.