International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 6839-6845

A Survey on Resource Scheduling Algorithms in Cloud Computing

Shweta Varshney, Sarvpal Singh

Madan Mohan Malaviya University of Technology, Gorakhpur, Uttar Pradesh- 273010, India.

Abstract

Cloud Computing is a distributed computing paradigm that provides computing i.e. processing, storage, services, network, and applications in an abstracted, virtualized, managed, and dynamically demand driven manner using Internet. It offers several distinguished features like virtualization, heterogeneity, measured service, pricing, resource pooling and elasticity. The explosive demand of cloud computing has led to the need of carefully managing the resources that provide services to the users. Resource Scheduling deals with this notion. It refers to the process of appropriate generation of the schedule that decides which tasks will be mapped on to which resources. In this paper, we describe all the important resource scheduling approaches that aim at optimizing the user Quality of Service (QoS) metrics such as cost, makespan, reliability, priority etc. An exhaustive survey on the approaches that tend to improve the user QoS metrics has been conducted. The deterministic, linear and evolutionary approaches to resource scheduling have been described along with the detailed comparison among these approaches.

Keywords: User QoS, Resource Scheduling, Deterministic approaches, Evolutionary approaches.

INTRODUCTION

Cloud Computing is an innovative technology that has brought a revolutionary transformation in the way computing services are delivered. With the proliferating growth of the Internet and the Web, Cloud Computing has changed the way information and communication technology users access resources. It has enabled to drive the focus from local/personal computation to datacenter-centric computation by providing resources dynamically in a virtualized manner via Internet. Cloud Computing transforms the usage of computing as the 5th utility which is charged on pay-per-use provision just as traditional utilities like water, electricity, gas and telephony [1]. Defining Cloud Computing in simple words, it is a blooming technological trend that provides computing i.e. processing, storage, services, network, and applications in an abstracted, virtualized, managed and dynamically demand driven manner using Internet. The resources which provide the services are somewhere placed on the Internet rather than our local system. And all this is provided to end users just like any other utility which they can access anytime, anywhere in the world with the help of Internet. It frees the users from the burden of managing hardware, software, storage and networks by providing them with a pool of virtualized resources according to their need thus, bringing in the concept of elasticity [2]. For such reasons, cloud computing is regarded as analogous to Internet these days.

While Cloud Computing finds its roots from existing paradigms such as Cluster and Grid Computing, it has some distinguished features like virtualization, heterogeneity, measured service and pricing, elasticity and resource pooling. In order to offer such distinguished features, cloud computing faces a lot of challenges like security and privacy, resource scheduling, scalability and fault tolerance, energy efficiency, interoperability etc. It is the task of the resource providers to take care of such challenges while delivering services to the users so as to maintain their confidence over cloud. On one hand, the service providers aim at maximizing their profit and return on investment, while on the other hand, users demand for cheapest, fastest and the most reliable services. In order to fulfil both the ends' demands, it is necessary that a proper resource management mechanism exists. In this paper, we emphasize on resource scheduling which is one of the most challenging problems in cloud from the providers' end as well as the users' end.

The remaining of this paper is arranged in the following way. Section II describes related work. The details about the Resource Scheduling Problem (RSP) are presented in Section III. Section IV provides details about several algorithms that are used to solve the RSP. A brief conclusion of the work along with providing the future directions is given in Section V.

RELATED WORK

In recent years, due to growing popularity of cloud computing, several researchers are attracted towards solving RSPs in cloud computing environment and have given several methods for the same. The concern is towards designing the scheduling algorithms that optimize the user Quality of Service parameters like makespan, execution time, deadline, reliability, response time, migration cost, availability etc. A lot of surveys have been conducted that incorporate different approaches to solve the same. Bala et al. [3] described the need of bringing cloud for executing workflows with the help of giving example of Online Banking System and reviewed about the existing algorithms that solved the problem of workflow scheduling. Salot [4] reviewed about the basic linear approaches that exist for solving RSP. An extensive survey was conducted by Singh et al. [5] over resource management that included resource provisioning as well as resource scheduling. They described various approaches to RSP by categorizing them according to

QoS parameters such as cost, time, profit, priority, SLA, energy etc. Yet another survey was conducted by Wadhonkar [6] who described architecture of cloud computing followed by description of the existing schemes to RSP.

CLOUD RESOURCE SCHEDULING PROBLEM

If we talk about resources, these refer to any component that can provide service to a user in any manner, may be it physical resource (e.g. processor, memory, storage, network elements, workstations etc.) or logical resource (e.g. operating system, energy, throughput, bandwidth etc.). Management of these resources involves both efficient provisioning and scheduling. Resource Provisioning refers to the process of appropriate detection, selection and allocation of resources that are needed to run the tasks and workflows. Resource scheduling is the process of appropriate generation of the schedule that decides which tasks will be mapped on to which resources. Both these procedures closely relate to adhering to Service Level Agreements and satisfying Quality of Service. In this paper, we concentrate on resource scheduling phase and on approaches that are meant to generate effective schedules.

The Resource Scheduling phenomenon is depicted by Fig. 1. Users or brokers on behalf of users submit tasks or workflows on the cloud computing environment using the cloud interface. Now the responsibility lies on the Resource Management System (RMS) to keep track of the status of tasks submitted, the number of resources required, maintaining the SLAs and successful completion of the tasks. Resource Provisioner carefully selects the resources for the tasks. When these resources are successfully provided for the execution, RMS calls resource scheduler. Various interactive components of resource scheduler include QoS monitor, execution manager, request monitor and priority checker.

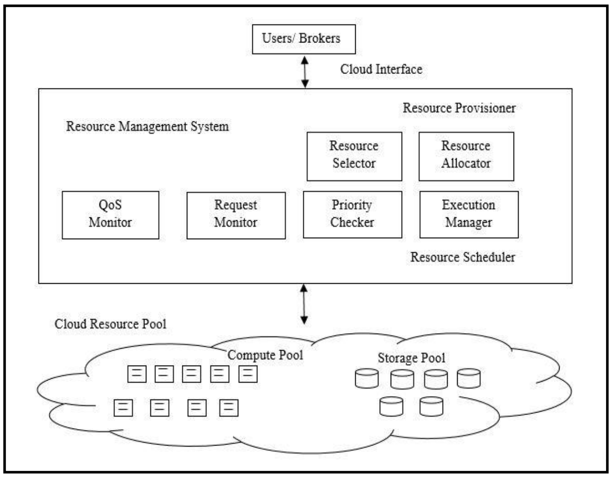

Figure 1. Cloud Computing Resource Scheduling

Resource scheduling algorithms can be classified into several types on the basis of goal which they are fulfilling. Some researchers classify algorithms on the basis of layers upon which they work i.e. scheduling in the Infrastructure Layer, Platform Layer, and Software Layer. Some researchers classify algorithms on the basis of information needed for scheduling i.e. static and dynamic algorithms. Depending on the type of data on which the algorithms work, they can be classified into independent task scheduling and workflow scheduling. Here, workflow refers to a set of interdependent tasks in which the successful completion of all the tasks cause the workflow to complete. Depending on the type of methods used, the scheduling algorithms are classified into deterministic and stochastic algorithms. Whilst in deterministic algorithms, computing is based on certain mathematical rules that cannot be modified and the search is done only in one direction, in stochastic algorithms, heuristic search depends on the judgment of the person concerned to find the solution and random search technique is used with no pre-defined rule. While there are many user QoS attributes based on which the algorithms are classified such as cost, time, reliability, priority, budget, deadline, SLA, there are some parameters from the providers' end also such as energy, bandwidth, efficiency, profit, load balance, resource utilization, cost effectiveness, and negotiation. In this paper, we focus primarily on various linear and evolutionary strategies that focus 6at optimizing the users' QoS attributes.

Coming back to RSP, it can be thought of as a function that maps jobs/tasks (say, we have a set of tasks x_1, x_2, x_3 ..., x_n) to a set of resources (Virtual Machines (VMs)) (say, we have a set of VMs v_1, v_2, v_3 …., v_m) such that the objective function is minimized/maximized. The jobs may be independent of one another or may consist of a sequence of tasks that are interdependent on each other. The objective function depends on user requirements and SLA agreements. The objective function may depend on just one parameter or it may depend on several parameters. The objective function is either weighed according to different parameters or the parameters are to be considered independently. The solution is to be found out within a given time frame. RSP is known to be a NP-hard problem [7] i.e. these problems cannot be solved in linear time frame with the increase in dimensions and complexity. Thus, researchers started using evolutionary approaches such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization to gain a better solution with a better speed of convergence. We firstly describe various linear strategies. Then, we move on for elaborating the nature inspired and metaheuristic algorithms that tend to tackle RSP.

RESOURCE SCHEDULING ALGORITHMS

In earlier days, when cloud computing was a new paradigm, scheduling strategies available for grid and cluster scheduling were used for cloud as well. To name some of them are, First Come First Served (FCFS) where services are served in the sequence in which they enter in the system, Round Robin (RR) scheduling in which a specified time quanta decides the period for which tasks will be executed in one go, Min- Min algorithm in which the task with minimum completion time gets allocated to resource where it takes minimum execution time, Max-Min algorithm, which allocates the task with maximum completion time on resource where it takes minimum execution time and Resource Aware Scheduling Algorithm (RASA) that incorporates Min-Min and Max-Min algorithms. While these algorithms performed well, they did not take into account cloud parameters as well as various other QoS parameters.

A. Deterministic/ Linear Approaches for Resource Scheduling

In this subcategory of work, we discuss various algorithms that made use of linear approaches such as priority, integer programming etc. Table I provides a brief comparison and description about these algorithms. Cao et al. [8] gave an activity based costing approach that says that volume of any program/ job does not determine its complexity , thus volume based cost drivers do not produce accurate results. A better scheme would be to measure the usage cost of every resource (CPU, Memory, I/O) that a job incurs while executing. This would evenly distribute the disturbed costs and would help in bringing accurate costs and more profit. Fang et al. [9] gave a load balancing approach for resource scheduling by considering cloud characteristics like elasticity and flexibility. Their approach worked at two levels: one at scheduling tasks to appropriate virtual machines and second at scheduling virtual machines at appropriate hosts such that the load is evenly distributed at both the levels. The dynamic nature of tasks was considered and migration of virtual machines if required was done according to the tasks' need. They focused on faster response time, thereby reducing makespan and better resource utilization. Here, makespan refers to the time at which the last task submitted executed successfully.

Sindhu et al. [10] came up with two simple algorithms that aimed at reducing the tasks' makespan. The two algorithms were Shortest Cloudlet to Fastest Processor (SCFP) and Longest Cloudlet to Fastest Processor (LCFP). Cloudlets, in cloud environment refer to the tasks submitted. SCFP, as the name suggests, arranges tasks as per their length and arranges the processors as per their processing capabilities. It then, maps the tasks from the sorted task list to the sorted processor list. LCFP works similar except that the tasks are arranged as per their length's decreasing order. It was found from the experiment that LCFP performed better than SCFP and FCFS. Ghanbari et al. [11] used a priority based job scheduling algorithm (PJSC). They used Analytical Hierarchy Process (AHP) which is a multi-attribute and multi- criteria decision model that calculate priority vectors and comparison matrix, the values of which decide the correct task mapping to correct resource. They performed a numerical simulation considering this consistent comparison matrix and aimed at reducing makespan while maintaining priorities.

Li [12] came up with stochastic integer programming model that uses Minimized Geometric Buchberger Algorithm (MGBA, an extension of Grobner Bases) which aims at satisfying various SLA constraints such as cost, throughput and latency. Numerical simulation was conducted and an optimum solution was achieved in reasonable time frame. Wu et al. [13] proposed a QoS based task scheduling with goals of executing the higher priority tasks on resources that take as much minimum time as possible. The priorities are decided according to the special QoS attributes. The algorithm was compared with Min-Min algorithm and Berger Model and the makespan of the proposed approach was found to be better than the former two. Another multi-QoS job scheduling approach was proposed by Abdullah [14] that used Divisible Load Theory (DLT). DLT aims at equally distributing the load among the existing machines so that the total completion time of tasks is minimized. With this approach, the finish time (from users' end) and the total cost spent (from the providers' end) was aimed to optimize and the results were found to improve with increase in number of processors.

A credit based task scheduling method was proposed by Thomas et al. [15]. They framed algorithms for computing credits of tasks as per their length and priority. Length credit was found by finding difference between actual length of tasks and average length. The priority credit refers to the quotient of priority of every task and its respective division factor. The total credit was the product of length credit and priority credit. The results were compared with credit systems taking length and priority individually and then taking collectively. The collective approach was found to perform better in attaining minimum makespan. Devipriya et al. [16] gave an improved max-min approach of finding the task scheduling solution by considering the completion time of all tasks rather than just execution time of current task i.e. it aimed at changing the Max-Min algorithm by selecting the resource that minimizes overall completion time. This scheme performed better than basic Max-Min algorithm. Lakra et al. [17] designed multi objective task scheduling algorithm that formed a non-dominated list of tasks according to the priorities assigned to the tasks as per QoS. The execution time of the proposed approach was better than FCFS and Priority Scheduling policy. With these deterministic approaches, a better scheduling environment was provided to the cloud computing users and the parameters such as execution time, priority and cost were more focused upon.

Table I. Comparison of Deterministic Approaches used for Resource Scheduling

Algorithm	Objective Criteria	Description	Experimental Environment	Experimental Scale	Results Compared
Activity based Costing Task Scheduling [8]	Cost	Calculates cost of applications on the basis of cost of use of resources (CPU, memory, I/O)	Algorithm explanation	NA	Traditional way of task scheduling
Task scheduling based on Load Balancing [9]	Makespan, Resource Utilization	Considers users' dynamic requirements by scheduling tasks to VMs with lightest load.	Cloudsim	1000 tasks, 100 resources	Grid Environment Load Balancing Algorithm
LCFP, SCFP[10]	Makespan	Considers computational complexity as the basis for making scheduling decision	Cloudsim	10-50 tasks 5 resources	FCFS
Priority Based Job Scheduling [11]	Makespan	Considers three level of priorities- scheduling, resource and job level.	Numerical Simulation	4 jobs 3resources	-
Stochastic Integer Programming [12]	Cost, Throughput, Latency	Applies Grobner Bases Theory to optimize SLA based resource schedule.	Numerical simulation	2-7 tasks 3-6 resources	-
QoS-Driven Task Scheduling [13]	Makespan, Average Latency	Uses several QoS parameters to assign priority to tasks.	Cloudsim	200-2000 tasks 50-100 resources	Min-Min and Berger Model
Cost-based Multi-QoS Job Scheduling [14]	Cost, Time	Uses Divisible Load Theory to distribute tasks evenly on all the resources.	Numerical Simulation	20-100 tasks 10-50 resources	-
Improved Max-Min [15]	Makespan	Improves performance of Max-Min algorithm by considering completion time rather than individual execution time.	Numerical simulation	4 tasks 2 resources	Max-Min Algorithm
Credit Based Algorithm [16]	Makespan	Assigns credit to each task based on its length and priority	Cloudsim	10 tasks 8 resources	Task Length Algorithm, Task Priority Algorithm
Multiobjectiv-e Tasks Scheduling [17]	Average Turnaround Time	Assign QoS values to both tasks and resources. Uses Non-Dominated Sorting to solve multi-objective function.	Cloudsim	3-10 tasks 20-200 resources	FCFS, Priority Scheduling

B. Evolutionary Approaches for Resource Scheduling

As mentioned earlier, scheduling problem is a NP-hard Problem, therefore attaining optimum solution using linear strategies is not much feasible. Thus, researchers have shown much interest towards nature inspired algorithm. The algorithmic complexity of evolutionary approaches has polynomial relationship with the problem scale. In this subsection, we will be describing the most popular evolutionary algorithms like Genetic Algorithms (GA), Ant Colony Algorithms (ACO), Particle Swarm Algorithms (PSO) and BAT algorithm along with the works carried out using such approaches in Cloud Computing environment. Table II provides a brief review about the approaches that use such algorithms in solving RSP.

Every evolutionary approach has three basic phases namely, initialization of population, fitness evaluation, and updation and finding the optimum solution iteratively. Starting with Genetic Algorithms, a chromosome forms a probable solution and the set of initial chromosomes is called population. After forming the initial population, the fitness function is evaluated and on the basis of "survival of the fittest" approach best chromosomes are selected. In the next step, these chromosomes form new offsprings by performing crossover and mutation operations. This process is iterated until the termination condition is met and the optimum solution is achieved.

Using this algorithm, Zhao et al. [18] formed an objective function to minimize the maximum execution time of any task. The numerical simulation was conducted with 2 tasks and 2 resources and the solution was found in limited time frame.

Kumar et al. [19] came up with providing a better initial population set by replacing the random population set with the population set found by running Min-Min and Max-Min algorithms. This improved the performance of standard genetic algorithm and the makespan for the same set of tasks improved. Yet another improvisation in standard Genetic Algorithm was brought by Gan et al. [20] who incorporated Simulated Annealing (SA) with the process of Genetic Algorithm. Simulated annealing not only helps in finding out a better optimum solution but also aids in reducing the probability of accepting worse chromosomes for next population by controlling the acceptance variable. In this paper, the author has considered weighted sum of five QoS parameters namely, completion time, bandwidth, cost, distance and reliability. The results were better than general Genetic Algorithm.

Coming to Particle Swarm Optimization, its working is similar to Genetic Algorithm. The population set is called as swarm which is a collection of particles. The dimensions of each particle corresponds to tasks. Each particle is a probable solution that decides for movement towards the best position in a particular direction depending on the local best and global best particle position. The particles have got their own velocities and positions. Each particle evaluates its own performance and during this, it is governed by two factors- retrospection and social interaction. The velocities and positions of the particle are updated according to the fitness function and the optimum solution is obtained.

Using PSO, Pandey et al. [21] designed an objective function to minimize the total cost of executing workflows in cloud computing environment. The total cost included execution costs and transmission costs. The results were compared against the Best Resource Selection (BRS) algorithm. And the utilization was found to be much better than the latter approach. Zhan et al. [22] used simulated annealing algorithm with PSO to improve the convergence rate of PSO using the fast searching ability of simulated annealing. This approach performed better than GA, SA and PSO. Another improvement was brought by Guo et al. [23] who updated the velocity equations and embedded PSO in local search and in crossover and mutation operations. Rodriguez et al. [24] proposed a deadline constrained resource scheduling as well as resource provisioning strategy using PSO. They aimed at minimizing the execution cost while meeting the deadline constraint. The PSO algorithm was used as it is. Only the objective to be obtained were with respect to cloud computing environment. Since PSO generates continuous values, a method for discretizing the values of PSO was given by Li et al. [25]. They proposed that resources should be numbered according to their computational speeds rather than just giving random values to them. The results came out to be better than basic PSO algorithm.

Ant Colony Optimization uses ants and pheromone values for finding the optimum solution. The population is initialized according to the pheromone vales. Then, in every generation, the performance pheromone and heuristic information is taken to help get the best resource for each task. Here, task is considered as each ant step. This algorithm is generally used in load balancing strategies. For resource scheduling, Banerjee et al. [26] improved ACO algorithm by modifying the pheromone update mechanism. And the makespan of the tasks executed reduced considerably. When compared with basic ACO algorithm, it performed better. Another modification was made by Wen et al. [27] who incorporated the benefits of PSO with ACO so that local optimum solution is not misinterpreted as optimum solution i.e. premature convergence is avoided. The solution performed better than basic ACO in terms of convergence speed and cost.

Apart from the above algorithms, some more algorithms have been developed recently. One of them is Cuckoo search Algorithm (CSA) that is used in cloud computing environment to solve resource scheduling problem by Navimipour et al. [28] CSA uses the characteristics of the flight behavior of animals which is similar to Levy Flights that can be successfully used in optimization problems. It gave promising results in terms of speed and convergence. BAT algorithm influenced by characteristics of bats was used by Raghavan et al. [29] for solving workflow scheduling problems. The cost of execution was better as compared to BRS algorithm.

Table II. Comparison of Evolutionary Approaches used for Resource Scheduling

Algorithm	**Objective Criteria**	**Description**	**Experimental Environment**	**Experimental Scale**	**Results Compared**
Genetic Algorithm [18]	Makespan	Uses Genetic Algorithm for task scheduling. Considers time utilization and resource utilization.	Numerical Simulation	2 tasks 2 resources	Traditional algorithm
Improved Genetic Algorithm [19]	Makespan	Uses Min-Min and Max-Min algorithm to initialize population of Genetic Algorithm	Cloudsim	10-40 tasks 10-40 resources	Standard Genetic Algorithm
Genetic Simulated Annealing Algorithm [20]	Multiple QoS parameters	Improves local serach ability of Genetic algorithm by introducing the process of Simulated Annealing	Numerical Simulation	20 tasks 8 resources	Standard Genetic Algorithm
Particle Swarm Optimization [21]	Cost	Considers collective minimization of both execution cost and transmission cost.	Cloudsim	Workflow with 5 tasks 3 resources	Best Resource Selection Algorithm

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 6839-6845

Algorithm	Objective Criteria	Description	Experimental Environment	Experimental Scale	Results Compared
Improved PSO based Task Scheduling [22]	Execution Time	Incorporates the process of simulated annealing in the PSO algorithm to improve convergence.	Cloudsim	50- 400 tasks	GA, SA, ACO, PSO
PSO based Heuristic [23]	Cost	Improves PSO by adding crossover and mutation and SPV.	Cloudsim	25-70 tasks 12-25 resources	Basic PSO
Deadline based Resource Scheduling [24]	Cost	Considers deadline satisfaction as the constraint. Uses basic PSO to optimize cost and time.	Cloudsim	Workflow with 9 tasks 3 resources	SCS and IC-PCP
Renumber Strategy Particle Swarm Optimization [25]	Cost	Uses a renumber strategy for resources to enable a better learning to the particles of PSO algorithm	Cloudsim	Workflow with 9 tasks 3 resources	Basic PSO
Modified Ant Colony Optimization [26]	Throughput	Improves the pheromone updation method of basic ACO algorithm	Google App Engine, MS Mesh	25 tasks 5 resources	Basic ACO
ACO PSO Resource Scheduling [27]	Execution Time	Incorporates ACO algorithm with PSO to avoid premature convergence.	Matlab	50-400 tasks	ACO Algorithm
BAT Algorithm [28]	Cost	Uses a new metaheuristic algorithm called as BAT Algorithm to minimise cost of execution.	Cloudsim	Workflow with 4 tasks 3 resources	BRS algorithm
Cuckoo Search Algorithm [29]	Speed of convergence	Uses a new metaheuristic algorithm called as Cuckoo Search Algorithm to increase speed of convergence.	Matlab	120 tasks 40 resources	-

CONCLUSION

The explosive demand of cloud computing has led to the need of carefully managing the resources that provide services to the users. Resource Scheduling deals with this notion. This paper has described various strategies that aim at tackling the resource scheduling problem. An exhaustive survey on the approaches that tend to improve the user QoS metrics has been conducted. Firstly, the linear strategies to solve resource scheduling problem that exist in literature have been presented in detail and then the evolutionary approaches to solve the same have been described. In each category, we have analyzed the current works with their objectives and methodologies, followed by a detailed comparison among them. Resource scheduling is still an emerging topic in Cloud Computing and much more is left to explore.

REFERENCES

[1] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.

[2] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., & Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing (Vol. 4, pp. 506-522). Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley.

[3] Bala, A., & Chana, I. (2011). A survey of various workflow scheduling algorithms in cloud environment. In 2nd National Conference on Information and Communication Technology (NCICT) (pp. 26-30).

[4] Salot, P. (2013). A survey of various scheduling algorithm in cloud computing environment. International Journal of Research in Engineering and Technology, 2(2), 131-135.

[5] Singh, S., & Chana, I. (2016). A survey on resource scheduling in cloud computing: Issues and challenges. Journal of grid computing, *14*(2), 217-264.

[6] Wadhonkar, A., & Theng, D. (2016, February). A survey on different scheduling algorithms in cloud computing. In Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016 2nd International Conference on (pp. 665-669). IEEE.

[7] Garey, M. R., & Johnson, D. S. (1979). Computers

and intractability: A guide to the theory of np - completeness (series of books in the mathematical sciences), ed. Computers and Intractability, 340.

[8] Cao, Q., Wei, Z. B., & Gong, W. M. (2009, June). An optimized algorithm for task scheduling based on activity based costing in cloud computing. In Bioinformatics and Biomedical Engineering, 2009. ICBBE 2009. 3rd International Conference on (pp. 1-3). IEEE.

[9] Fang, Y., Wang, F., & Ge, J. (2010, October). A task scheduling algorithm based on load balancing in cloud computing. In International Conference on Web Information Systems and Mining (pp. 271-277). Springer, Berlin, Heidelberg.

[10] Sindhu, S., & Mukherjee, S. (2011). Efficient task scheduling algorithms for cloud computing environment. In High Performance Architecture and Grid Computing (pp. 79-83). Springer, Berlin, Heidelberg.

[11] Ghanbari, S., & Othman, M. (2012). A priority based job scheduling algorithm in cloud computing. Procedia Engineering, 50, 778-785.

[12] Li, Q. (2012). Applying Integer Programming to Optimization of Resource Scheduling in Cloud Computing. JNW, 7(7), 1078-1084.

[13] Wu, X., Deng, M., Zhang, R., Zeng, B., & Zhou, S. (2013). A task scheduling algorithm based on QoS-driven in cloud computing. Procedia Computer Science, 17, 1162-1169.

[14] Abdullah, M., & Othman, M. (2013). Cost-based multi-QoS job scheduling using divisible load theory in cloud computing. Procedia computer science, 18, 928-935.

[15] Thomas, A., Krishnalal, G., & Raj, V. J. (2015). Credit based scheduling algorithm in cloud computing environment. 9 913-920.

[16] Devipriya, S., & Ramesh, C. (2013, December). Improved max-min heuristic model for task scheduling in cloud. In Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on (pp. 883-888). IEEE.

[17] Lakra, A. V., & Yadav, D. K. (2015). Multi-objective tasks scheduling algorithm for cloud computing throughput optimization. Procedia Computer Science, *48*, 107-113.

[18] Zhao, C., Zhang, S., Liu, Q., Xie, J., & Hu, J. (2009, September). Independent tasks scheduling based on genetic algorithm in cloud computing. In Wireless Communications, Networking and Mobile Computing, 2009. WiCom'09. 5th International Conference on (pp. 1-4). IEEE.

[19] Kumar, P., & Verma, A. (2012). Independent task scheduling in cloud computing by improved genetic algorithm. International Journal of Advanced Research in Computer Science and Software Engineering, 2(5).

[20] Gan, G. N., Huang, T. L., & Gao, S. (2010, October). Genetic simulated annealing algorithm for task scheduling based on cloud computing environment. (pp. 60-63). IEEE.

[21] Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010, April). 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 (pp. 400-407). IEEE.

[22] Zhan, S., & Huo, H. (2012). Improved PSO-based task scheduling algorithm in cloud computing. Journal of Information & Computational Science, 9(13), 3821-3829.

[23] Guo, L., Zhao, S., Shen, S., & Jiang, C. (2012). Task scheduling optimization in cloud computing based on heuristic algorithm. *JNW*, *7*(3), 547-553.

[24] Rodriguez, M. A., & Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE transactions on cloud computing, 2(2), 222-235.

[25] Li, H. H., Fu, Y. W., Zhan, Z. H., & Li, J. J. (2015, May). Renumber strategy enhanced particle swarm optimization for cloud computing resource scheduling. In Evolutionary Computation (CEC), 2015 IEEE Congress on (pp. 870-876). IEEE.

[26] Banerjee, S., Mukherjee, I., & Mahanti, P. K. (2009). Cloud computing initiative using modified ant colony framework. World academy of science, engineering and technology, 56(32), 221-224.

[27] Wen, X., Huang, M., & Shi, J. (2012, October). Study on resources scheduling based on ACO algorithm and PSO algorithm in cloud computing. In Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2012 11th International Symposium on (pp. 219-222). IEEE.

[28] Navimipour, N. J., & Milani, F. S. (2015). Task scheduling in the cloud computing based on the cuckoo search algorithm. International Journal of Modeling and Optimization, 5(1), 44.

[29] Raghavan, S., Sarwesh, P., Marimuthu, C., & Chandrasekaran, K. (2015, January). Bat algorithm for scheduling workflow applications in cloud. In Electronic Design, Computer Networks & Automated Verification (EDCAV), 2015 International Conference on (pp. 139-144). IEEE.