

Comparison of Task Scheduling Algorithms in Cloud Environment

Babur Hayat Malik, Mehwashma Amir, Bilal Mazhar, Shehzad Ali, Rabiya Jalil, Javaria Khalid

Department of CS and IT
The University of Lahore, Gujrat Campus, Pakistan

Abstract—The enhanced form of client-server, cluster and grid computing is termed as Cloud Computing. The cloud users can virtually access the resources over the internet. Task submitted by cloud users are responsible for efficiency and performance of cloud computing services. One of the most essential factors which increase the efficiency and performance of cloud environment by maximizing the resource utilization is termed as Task Scheduling. This paper deals with the survey of different scheduling algorithms used in cloud providers. Different scheduling algorithms are available to achieve the quality of service, performance and minimize execution time. Task scheduling is an essential downside within the cloud computing that has to be optimized by combining different parameter. This paper explains the comparison of several job scheduling techniques with respect to several parameters, like response time, load balance, execution time and makespan of job to find the best and efficient task scheduling algorithm under these parameters. The comparison of scheduling algorithms is also discussed in tabular form in this paper which helps in finding the best algorithms.

Keywords—Task scheduling; algorithms; cloud computing; min-max; genetic algorithm; load balancing; resource utilization

I. INTRODUCTION

In scientific community, Cloud Computing has gained a vast amount of attention. Cloud Computing provides an environment which is more flexible rather than its counterparts. Cloud Computing provides the facility to access the data anywhere from your cloud [1]. Organizations are shifting their businesses toward cloud computing because cloud computing providing resources in large quantity and user/organizations are using resources freely.

Cloud Computing is a model which provide easy access to available resources to cloud users on their demand [2]. Cloud provides a variety of services on the demand of to its user, e.g. dynamically network access, rapid elasticity. The popularity of cloud depends on its performance, manage resource and optimally job scheduling.

This paper is mainly focusing on different task scheduling approaches. Task scheduling can be defined as choosing the most appropriate and suitable resources for the execution of the task. The task can also be defined as user's queries send to the different server, and these queries also accomplished within required time period [8]. Task scheduling works on principle of distributed the task on available resources.

The main objective of scheduling algorithms in

decentralized environment is to extend different task on servers to balance the load, this maximize the utilization of processors and minimize the execution time of user task. The central objective is to schedule available resource according to the available time for its execution. The task may include entering a query, a process that query, accessing the required software and memory [1]. Then data centre classifies user's queries on the requirements made on the service requested and agreement of services.

The user task is appoint to one of the available servers, and result or response of task is sent back to the user. Task by the cloud users are dispatched to available resources for their timely execution is task scheduling [26]. Several task scheduling Algorithms are used to increase the performance of cloud and enhanced throughput of servers. The different parameters of scheduling are used to increase the overall cloud performance [2]. There are several limitations while scheduling a task such as a cost, throughput, time, resource utilization and make span [26]. The main contribution in task scheduling is to minimize cost and time to produce an optimal result which causing to increases the performance of the cloud.

The coming part of the paper will explain classification of scheduling and scheduling process. Section III is comprised of the literature review and Section IV explained the working of several task scheduling algorithms helping to make a brief comparison of different algorithms and discusses the results of this research. After the comparison of scheduling algorithm at last we conclude the best and efficient scheduling algorithm.

II. CLASSIFICATION OF SCHEDULING

Scheduling methods are classified into three main groups: task scheduling, resource scheduling and workflow scheduling. The distribution of virtual resources among Servers (physical machine) is done by Resource Scheduling. To scheduling workflow comprised by an entire job in an efficient order. Task Scheduling is to assigned the task to available resource for its execution. Task Scheduling method is for centralized as well as decentralized structure and also for the homogenous and heterogeneous environment [25]. The paper mainly focuses on the task scheduling algorithms and their comparison. Several Task Scheduling Algorithms are diagrammatically shown in Fig. 1. Allocating resources to any task is considered as task scheduling, and it is the main component of cloud computing. The most significant factor in task scheduling is time and cost is required for its completion.

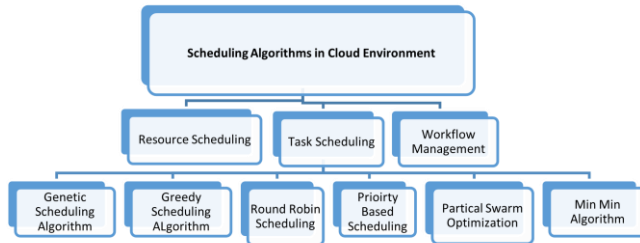


Fig. 1. Classification of scheduling techniques [2].

III. SCHEDULING

Distributed computing environment comprised of several scheduling techniques. These techniques is implemented to cloud environment with applicable parameters. The central purpose of these techniques is to improve the response time of system and the performance of the cloud [3]. Conventional Scheduling techniques are not able to increase the performance, it maximize the cost and execution time as well. Scheduling algorithms which are discussed in this paper are min-min, First Come First Serve, Round Robbin, Genetic Algorithm, Greedy Algorithm, Partical Swarm Optimization, Priority based scheduling, etc. [4], [24].

A. Scheduling Process

Cloud Task scheduling process is generally classified into three stages [7] are shown in Fig. 2:

Resource Dicorvey and Filtering: The cloud service provider discover list of available resource in a secific network and also collect and check their working status.

Selction of Resources: It is the most important stage in task scheduling, is also known as deciding stage. Required resources are selected on specific parameter and according to the requiremt of task execution.

Submission of Task: After slecting the required resource the task is submit to the resource for execution [34].

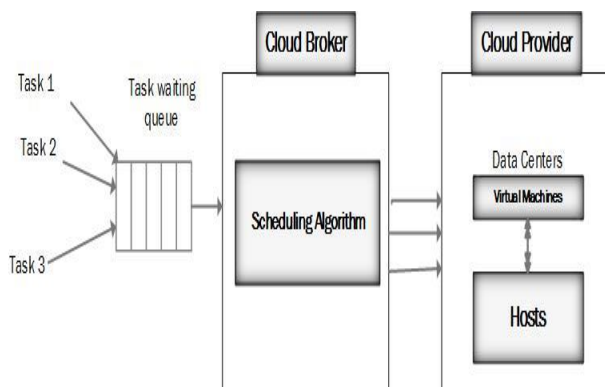


Fig. 2. Task scheduling process [4].

IV. LITERATURE REVIEW

The major issue in task scheduling is the allocation of efficient and available resources to the new task enter by the user. If several tasks arrive at the same time, then dynamically

resource allocation process is become more complex. Therefore, S. Ravichandran and D. E. Naganathan [7] had proposed a new system to get rid from this problem, this system works when a new task is arrived it is sent into the queue for waiting and the job scheduler will easily order each task and allocate resources for their execution. Genetic Algorithm is considered as a best practice in this regard, all the tasks are sent to the queue, scheduler pick the task from the queue allocate resources and execute the task. The central purpose of this system is to minimize the execution time of the task and optimize the resource utilization.

In this paper authors, V. V. Kumar and S. Palaniswami mainly focused on enhancing the efficiency of job scheduling techniques for cloud computing service. They have also purposed an algorithm which optimally utilized the turnaround time by giving high priority to different job for its completion and less priority for termination issues of real-time task [9]

Moreover, a new task scheduling algorithm was purposed by Z. Zheng which is based on genetic algorithm which is termed as Parallel Genetic Algorithm. The main objective of this algorithm is to optimize the cloud scheduling problem mathematically.

Siad Bin Alla and Hicham Bin Alla in [22] have explained a novel based dynamic task scheduling technique which is based on improved genetic algorithm. The working of proposed algorithm is mainly focused to reduce the execution time, effectively improve the throughput and the scalability of the cloud in task scheduling.

In [33], author proposed a novelistic approach for task scheduling algorithm M Quality of Service with Genetic Algorithm and Ant Colony "QOS-GAAC" with multi-QOS constraint, in this algorithm author mainly focused on expenditures, security, time-consuming and reliability in the process of task scheduling. This algorithm is the combination of genetic Algorithm and ant colony optimization algorithm. The result represent that this algorithm has great performance in both guaranteeing QOS and resource balancing in task scheduling [27].

Author proposed an optimized algorithm to minimize the cost and bi-objective makespan used by heuristic search techniques for independent tasks scheduling [32]. Integer PSO is a new variant is proposed; the main objective of this variant is to solve the task scheduling problem in cloud. Integral-PSO is an improved and continuous form of Particle swarm optimization.

V. TASK ANALYSIS ALGORITHMS

A. Genetic Algorithm

The most transformative algorithms are the genetic algorithms which are dependent on the concept of natural transformation [14]. This genetic algorithm is promptly emerging in the field of artificial intelligence [6], [3]. It works on the processing of every task as shown in Fig. 3, in which the quality of each task is being processed according to the user requirements unless the user is being satisfied [9]. Darwin's theory introduced the idea of "Survival of the fittest" which basically processes the tasks according to their

allocation to the resources on the base of their fitness value functions [12]. It doesn't process that task as whole rather it evaluates each parameter of that task on basis of fitness value [4] [20]. The generic terms of this algorithm are as follow:

a) Initial Population

In this algorithm there are several number of individuals which operates the tasks in an iterative way and so several number of solutions are being fixed up, such solutions are termed as populations, in every specific iteration. In that population every solution is termed as chromosome. Ten chromosomes are being selected from that population [5]. From this an initial population, ten chromosomes are selected unsystematically [6].

b) Fitness Function

The basic purpose of this function is quality evaluation of each individual in population while depending upon approach of optimization. It is more often dependent on deadline but in few cases it is dependent on the budget constraints [7].

c) Selection

In this process an operator is used known as proportional selection which is used for evaluating the probability and fitness between two algorithms. It identifies that either selected probability or next groups are proportional to fitness of each individual [10].

d) Crossover

Purpose of this process is the selection of best fitted pair of individuals for crossing over and this is not done without the usage of an operator known as single-point crossover operator. The benefit of crossing over is that both sides' portions can be exchanged [6], [10].

e) Mutation

New individuals are not generated in easy way for that purpose; some of gene locus is being substituted by other gene locus values and it is done in the coding series. A very small value (0.05) is chosen as mutation probability [11].

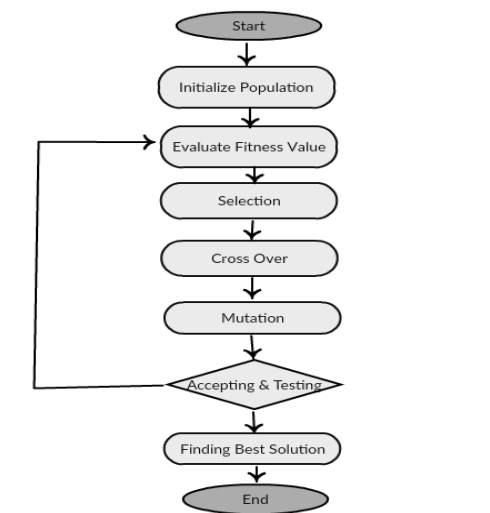


Fig. 3. Working of Genetic Algorithms [5].

B. Greedy Algorithm

Greedy Algorithm is used for solving the problems by making decisions considered best in that particular situation. Working of Greedy Algorithm is explained in Fig. 4. Optimization issues can be easily solved with the help of this algorithm. Though some problems do not seem easy enough with efficient solutions but they are being solved with the help of greedy algorithm with the finest solutions [11]. There are some deviations to the greedy algorithm:

- Pure greedy algorithms
- Orthogonal greedy algorithms
- Relaxed greedy algorithms

When some agitations occur such as bad weather and so on, few constraints in above model are being effected due to which the entire schedule become totally unworkable. The basic purpose of this algorithm was to overcome such problems in each and every step and makes the finest decisions. Its main aim was to get the finest solution and keep on working that schedule unless all the problems are being solved [13]. Due to this optimization of the large problem was divided into small size problems and this helped in identification of solutions in less time [12]. Basic working of Greedy Algorithm is as follows:

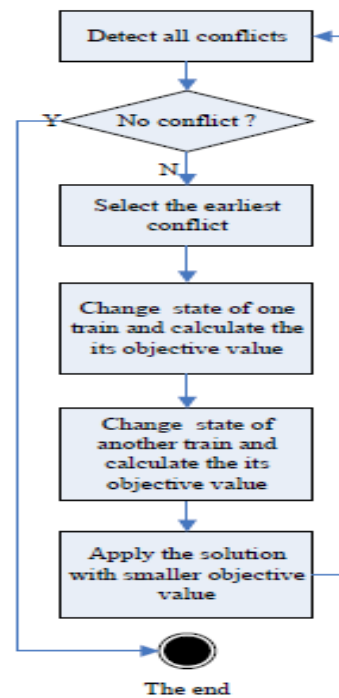


Fig. 4. Working of Greedy Algorithms [12].

Some standards of Greedy Algorithm are as follows [12]:

a) Kruskal's Minimum Spanning Tree (MST)

In this MST is created by selecting edges not collectively but individually. The greedy choice is always selecting the edge of lightest weight because it wouldn't create a cycle in MST.

b) Prim's Minimum Spanning Tree

MST is being created again in it but we manage two sets: set of vertices which are already being added up in MST and the set of vertices which is not added yet. Those edges are selected which are less in weight [11].

c) Dijkstra's Shortest Path

It is very similar to Prim's algorithm. In this the shortest path tree is being built up by every single edge. We manage two sets: set of vertices which are already being added up in MST and the set of vertices which is not added yet [18]. Greedy choice is selection of the smallest weight path.

d) Huffman Coding

Loss-less compression technique is considered as the base of this algorithm. It allocates variable length bit codes to different characters. The Greedy Choice is to assign least bit length code to the most frequent character [11], [12].

C. Priority-based Job Scheduling Algorithm

In Cloud computing, an innovative approach to deal with programming work is presented by Shamsollah Ghanbari and Mohamed Othman by using mathematical measurements [19]. The significance of the job for programming is considered by this algorithm and is called the Algorithm for priority based job scheduling algorithm "PJSC". It is centred as the multiplicative standards decision-making model. There are three levels of priorities in this algorithm that are programming level, resource level and work level which is shown in Fig. 5.

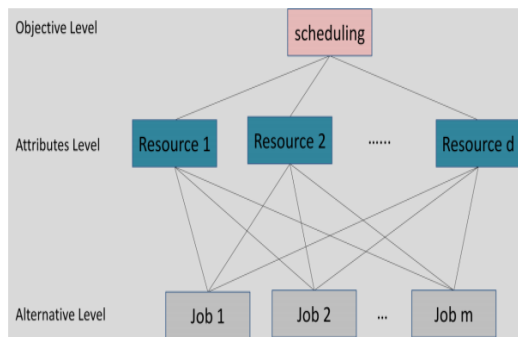


Fig. 5. Three level of priority based scheduling [20].

In this algorithm jobs set are taken as $J = \{J_1, J_2, J_3, J, \dots, J_m\}$ that demands assets in a cloud atmosphere and resources set are taken as $C = \{C_1, C_2, C_3, C_4, \dots, C_d\}$ that is presented in cloud atmosphere as input where $(d \ll m)$. Each job set demands a resource with the required priority. Priority of different jobs set is compared independently [28]. Each job is allocated a resource with the specified priority. Hence, the correlation networks of each activity/job set are computed according to the prospects of retrieving the resources, and the matrix of comparison of the resources is also computed. Now the normal matrix of all jobs with the name Δ is calculated by comparing the each of the job set matrices and priority paths are also calculated [21]. Then the normal resources matrix is calculated, and the name of the matrix is given as γ .

Now the PVS (priority vector of S) is calculated in this algorithm and S is stated as a set of jobs. The matrix Δ is multiplied with the matrix γ which is resulted as PVS. Now the highest ranked job is chosen, and resource is allocated to that job. Job's list is upgraded with the time, and the programming procedure proceeds until the point that all jobs are planned in a suitable resource [33]. The trial/experiments come about show that the calculation of the algorithm has the rational complexity. There are additionally a few issues identified with this calculation, for example, completion time, consistency and complexity [19], [21].

D. Round Robin

The round robin is a simple example of load balancing technique. A round robin technique was designed to divide scheduling time among all scheduled task equally, in which all tasks get in queue list, and each task gets an equally small unit of time as clearly explained in Fig. 6. The major concern of RR is to focus on dividing load to all resources equally [14]. A cyclic approach is applied in round robin. The scheduler picked a task and assigned to the controller and after time expires of the first task then move to next task [17]. This is the cyclic approach in which all task assigned to the controller at least once and then scheduler again pick up the first task again.

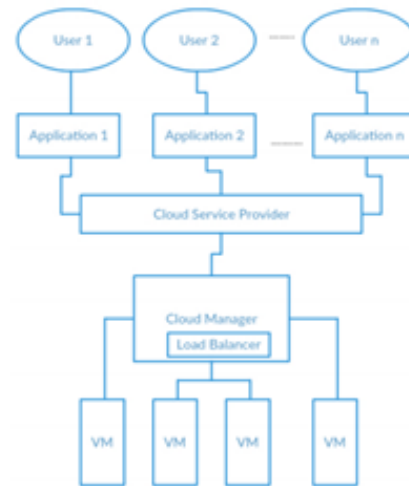


Fig. 6. Load Balancer Round Robin [10].

The load balancing and response time are much better compared to other algorithms. The working of Round Robin in cloud environment is same as round robin in process scheduling [15]. In round robin, each task has an equal opportunity to be chosen [28].

E. Min-Min Algorithm

This scheduling technique works on strategy in which task has minimum execution time is selected for all task. This algorithm starts when a set of all jobs are not assigned and continue to execute until the whole set of the job is empty [16]. In Min-Min jobs having the greater time or long task may not be considered first and the task having greater time will always follow the short job. In this algorithm completion

time of all tasks is computed then job having smallest completion time is scheduled on resource [16].

The formula for calculating completion is as follows:

$$T_{finish} = T_{exe} + T_{start} \quad (1)$$

T_{finish} is finish time, T_{exe} is expected execution time, and T_{start} is starting time of the task

$$T_{comp} = \max(T_{finish} = T_{exe} + T_{start}) \quad (2)$$

The Job which is mapped to resource first after its completion is deleted and the process repeated until all task is mapped. In-Min causes all set of tasks executed get a longer time and unbalanced load even in some cases long task cannot be considered [23].

F. Particle Swarm Optimization (PSO)

This is meta-heuristic population-based algorithm exhilarated by social manners of fish schooling and bird flocking [29]. The algorithm contains set of particles, and each particle depicts a solution for the problem in given search space which is then used to approach convenient solutions [19]. This algorithm is initialized by a set of random particles and then finding a best solution in problem space. In PSO we use iteration to find out each particles position which is referred as Personal best P_b achieved by particle i and global best P_g . Position found by neighbour particle i . Equations to update particles velocity and position after finding both global personal values are [22].

Equation (1):

$$v_i^{t+1} = \omega v_i^t + c1r1(p_i^t - x_i^t) + c2r2(p_g^t - x_i^t) \quad (3)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (4)$$

Where particles velocity and position in dimension d of the i^{th} particle are represented by v_i^t and x_i^t respectively. The PSO parameter ω , $c1$ and $c2$ should be considered properly to increase the efficiency of algorithms. This helps in finding out best solution in short computing time [30].

Once all the tasks are queued in a cloud environment, the optimization algorithm is then used to calculate minimum waiting time values of all jobs. These minimum values used for correct order of task which in return minimize the overall waiting time [31]. After getting bested optimal order of task, Queue generated algorithm is applied to find the threshold then dispatches a task to this queue. The scheduler then schedules a task to a suitable resource (server) [35].

The main objective of PSO is to allocate a user request to a suitable resource [33], [35]. To schedule a task on cloud environment efficiently, the task scheduling process requires such optimal algorithm that takes a task and resources into consideration. The PSO algorithm considers both resource and task and helps in keeping the resource as busy as possible and minimizing the processing time of the task [31], [33].

G. First Come First Serve Algorithm

The most fundamental and simplest techniques which uses the task arrival time to schedule the task on cloud environment. The task will be schedule and executed depending on which task has arrived first in queue.it totally depends on arrival time and doesn't consider any other parameter. The tasks will be scheduled by selecting correct order of jobs. The task or user request which comes first to data centre will be assigned to VM first for execution. The data centre controller checks for free Virtual machine and then assign task to that VM then remove that task from queue.

If four task arrives on cloud environment having three virtual machines then FCFS scheduler will schedule three task on VM parallel leaving one task until one VM becomes free for first schedule. For second schedule if task 4 has Childs then childs can't be executed until their parent executed. When task 4 is executing on VM then two other VM remains idle which cause the less utilization of resources.

VI. COMPARISON OF EXISTING TASK SCHEDULING ALGORITHMS

Task scheduling algorithms that we discussed earlier are compared in by their methodology.

TABLE. I. COMPARISON OF VARIOUS TASK SCHEDULING ALGORITHMS

Algorithms	Methodology	Parameters	Merits	Demerits
Genetic Algorithm	Genetic algorithm wants a depiction of the solution domain and suitable function to estimate the solution domain.	1.population size 2. Crossover probability 3.mutation probability	1. It can solve the mathematical problems and financial problems more accurately. 2. Easy to understand the concepts. 3. Some applications required less time for processing.	1. This algorithm work very slowly. 2. This algorithm cannot find the exact solutions. 3. Method of selection should be appropriate.
Greedy Algorithm	This algorithm tries to find the global optimum by following the problem-solving heuristic approach for choosing every step.	1. Parameter μ 2. Domain D 3. Population n	1. Easy to implement. 2. This algorithm needs fewer resources. 3. Execution is very fast. 4. Scheduling is very fast.	1. Global optimization solution is not fulfilled by this algorithm. 2. Very difficult to make changes in parameters.
Priority-Based Job Scheduling Algorithm	Dependency mode	1. Priority to each queue	1. Priority of the process increases with the increases in the time. 2. Easy to use and user-friendly. 3. Best for the applications which require time and	1. Jobs having lowest priority will be lost when the system crashes. 2. Starvation for resources they need.

			resources. 4. Less finish time	
Round Robin	The algorithm works on cyclic approach in which each task has equal chance to be chosen and has an equally small unit of time for execution	1. Arrival time 2. Time slice	1. Response time is good. 2. Load is balanced. 3. less complex	1. Pre-emption causes the process out once time slice expires
Particle Swarm Optimization	The algorithms use population to find the optimal minimum values that help in creating a correct order of tasks and schedule task to a suitable resource	1. Inertia, 2. C1,C2 constants	1. high utilization of resources, finding the optimal solution, minimizing processing time	1. Slow convergence speed if search space is large
Min-Min Algorithm	This algorithm works on strategy in which task having minimum execution time is selected for all task	1. Makespan	1. Better makespan	1. load imbalance 2. poor QoS
First Come First Serve	This algorithm manages the task scheduling with FIFO queue. Task which comes first will be executed first on VM	1. Arrival time	1. Simple and fast execution	1. Task scheduling is based on arrival time, doesn't consider any other criteria Less utilization of VM

VII. DISCUSSION

Task Scheduling is one of the biggest challenges in Cloud Computing. The principle motive of task scheduling is to distribute the incoming tasks from users to the available virtual machines keeping in mind the different parameters Load Balancing, execution time, load balance, Quality of service, performance, response time and fairness resource allocation in which task can be executed. Some algorithms consider only load balance while some consider response time. As most algorithms works with one or two parameters, due to which good result can't be achieved effectively. Better results can be produced by coupling more scheduling metrics to generate one efficient algorithm as an enhancement but this can be little bit complex.

VIII. CONCLUSION

Efficient scheduling algorithm can yield more desirable services to users and increase the performance provided by cloud environment. The main objective of task scheduling in cloud environment is to reduce the execution time of tasks and to maximize the resource utilization. In this paper, a study related to different existing task scheduling algorithms in a cloud environment has been presented. A short description of each algorithm methodology has been presented and most algorithms consider on one or two parameters. More satisfactory results can be achieved by adding more metrics to existing algorithms. Table I is based on different scheduling parameters such as execution time, load balance, Quality of service, performance, response time and makespan. The major problem in task scheduling is load balancing, response time, resource utilization and memory storage. Efficient scheduling algorithm can be achieved by combining different parameters to existing algorithms which will improve their overall performance of cloud environment.

REFERENCES

- [1] S. Sonia, "Task Scheduling in Cloud Computing," International Journal of Advanced Research in Computer Engineering & Technology (IJARECT), vol. 4, no. 6, June 2015.
- [2] D. a. S. N. R. Nallakumar, "A Survey of Task Scheduling Methods in Cloud Computing," International Journal of Computer Science and Engineering (JCSE), vol. 2, no. 10, 31 Oct 2014.
- [3] P. H. Srimathi, "Survey and Analysis of Task Scheduling in Cloud Environment," Indian Journal of Science and Technology, vol. 9(37), October 2016.
- [4] M. B. Yougita Chawla, "A study on Scheduling Methods in Cloud Computing," International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol. 1, no. 3, October 2012.
- [5] F. A. O. Safwar A. Hamad, "Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 7, no. 4, 2016.
- [6] T. Y. K. J. K. K. a. J. S. L. S. H. Jang, "The study of genetic algorithm-based task scheduling for cloud computing," International Journal of Control and Automation, vol. 5, pp. 157-162, 2012.
- [7] T. G. a. A. Agrawal, "Host Scheduling Algorithm Using Genetic Algorithm In Cloud Computing Environment," International Journal of Research in Engineering & Technology (IJRET), vol. 1, 2013.
- [8] R. R. a. R. N. C. R. Buyya, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in High Performance Computing & Simulation, 2009. HPCS'09. International Conference on, 2009.
- [9] B. M. P. D. . J. K. M. a. S. D. K. Dasguptaa, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing," International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA), 2013.
- [10] S. A. H. a. A. Omara, "Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 7, no. 4, 2016.
- [11] A. S. a. M. V. S. A. Malik, "Greedy Algorithms," International Journal of Scientific and Research Publications, vol. Volume 3, no. Issue 8, August 2013.
- [12] Z. HE, "Research on Improved Greedy Algorithm for Train Rescheduling," Seventh International Conference on Computational Intelligence and Security, 2011.
- [13] V. K. Z. B. S. B. N. Krivsha, "Greedy algorithms for granular computing problems in spatial granulation technique," XIIth International Symposium «Intelligent Systems», INTELS'16, 5-7 October 2016, Moscow, Russia.
- [14] M. Ann Arbor, "J.H. Adaptation in Natural and Artificial Systems," in University of Michigan Press, Holland.
- [15] J. P. a. X. F. Gaochao Xu, "A load Balancing Model Based on Cloud Partitioning for the Public Cloud," TSINGHUA SCIENCE AND TECHNOLOGY, vol. 18, no. 1, pp. 34-39, February 2013.
- [16] J. L. a. J. X. Gang Lui, "An Improved Min-Min Algorithm in Cloud Computing".
- [17] E. a. GibetTaniHicham, "Smarter Round Robin Scheduling Algorithm for Cloud Computing and Big Data," Journal of Data Mining and Humanities, 23 January 2017.
- [18] M. Ann Arbor, "J.H. Adaptation in Natural and Artificial Systems;," University of Michigan Press.
- [19] S. a. M. Othamn, "A Priority Based Job Scheduling Algorithm in Cloud Computing," International Conference on Advances Science and Contemporary Engineering, 2012.
- [20] W. H. Z. J. Yin H., "An Improved Genetic Algorithm with Limited Iteration for Grid Scheduling," Yin H., Wu H., Zhou J., "An Improved

- Genetic Algorithm with Limited Iteration for Grid Scheduling", IEEE Sixth International Conference on Grid and Cooperative Computing, 2007. GCC 2007, Los Alamitos, CA.
- [21] S. P. a. U.Bhoi, "Priority Based Job Scheduling Techniques in Cloud Computing: A Systematic Review," INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH , vol. 2, no. 11, NOVEMBER 2013 .
- [22] S. B. A. A. E. a. A. M. Hicham Bin Alla, "A Novel Architecture with Dynamic Queues Based on Fuzzy Logic and Partical Swarm Optimization Algorithm for Task Scheduling in Cloud Computing," Advances in Ubiquitous Networking 2, Lecture Notes in Electrical Engineering, 2017.
- [23] A. I. E.-D. M. F. A.-r. El-Sayed T. El-kenawy, "Extended Max-Min Scheduling Using Petri Net and Load Balancing," International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, vol. 2, no. 4, September 2012.
- [24] D. B. J. K. J. B. Shalmali Ambike, "An Optimistic Differentiated Job Scheduling System for Cloud Computing," International Journal of Engineering Research and Applications (IJERA) , vol. 2, no. 2, pp. 2248-9622.
- [25] M. J. ., M. B. S. a. M. K. T. Arash Ghorbannia Delavar, "'RSDC (RELIABLE SCHEDULING DISTRIBUTED IN CLOUD COMPUTING)";" International Journal of Computer Science, Engineering and Applications (IJCEA) , vol. 2, no. 3, June 2012.
- [26] ". Saeed Parsa and Reza Entezari-Maleki, " RASA: A New Task Scheduling Algorithm in Grid Environment," World Applied Sciences Journal 7 (Special Issue of Computer & IT), pp. 152-160, 2009.
- [27] Mrs.S.Selvarani1 and D. Sadhasivam, "improved cost-based algorithm for task scheduling in Cloud computing," IEEE, 2010.
- [28] A. I. E.-D. M. F. A.-r. El-Sayed T. El-kenawy, "'A Priority based Job Scheduling Algorithm in Cloud Computing," International Conference on Advances Science and Contemporary Engineering 2012 (ICASCE 2012).
- [29] M. S. S. Kalra, "A review of metaheuristic scheduling techniques in cloud computing Egypt. Inf. J. 16(3), 275–295 (2015). Elsevier".
- [30] J. J. e. a. Durillo, "Multi-objective particle swarm optimizers: An experimental comparison," International Conference on Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2009.
- [31] A. a. S. K. Verma, "Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud," Engineering and Computational Sciences (RAECS), 2014 Recent Advances in. IEEE, , 2014.
- [32] Y. L. Y. L. X. Dai, "A task scheduling algorithm based on genetic algorithm and ant colony optimization algorithm with multi-QoS constraints in cloud computing," In: 7th International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 428-431, IEEE (2015).
- [33] A. R. M. Beegom, "Beegom, A., Rajasree, M.: A particle swarm optimization based pareto optimal task scheduling in cloud computing," Lecture Notes in Computer Science, p. 79–86, Springer (2014).
- [34] P. Salot, "A survey of various scheduling algorithm in cloud computing environment," International Journal of Research in Engineering and Technology, vol. 2, no. 2, pp. 131-135., 2013.
- [35] "Parallel Workloads Archive: NASA Ames iPSC/860. http://www.cs.huji.ac.il/labs/parallel/workload/l_nasa_ipsc/."