JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

RESOURCE MANAGEMENT SCHEDULING IN CLOUD COMPUTING

[1]Mayanka Gaur, [2]Manisha Jailia
[1]PhD Scholar, [2]Associate Professor
[1,2]Department of Computer Science & Engineering,
[1,2]Banasthali Vidyapith, Niwai, Tonk, Jaipur, Rajasthan, India
[1]gaur.mac1305@gmail.com, [2]manishajailia@yahoo.co.in

Abstract : Cloud computing is a paradigm of distributed computing that uses the Internet to provide computing in an abstract, virtualized, managerial, and dynamic way driven by demand, that is, processing, storage, services, networks and applications. When we talk about hosting, it means that we are talking about providing services and resources to users in a planned and hosting way. That means we are talking about resource management and resource scheduling. Resource management and resource scheduling both are important now a days because of high demand of cloud computing in Computer Science field. The high demand for cloud computing has led to the need to manage resources through resource scheduling. In this article, we will describe and discuss all the important resource scheduling algorithms aimed at optimizing user quality of service (QoS). Algorithms that tend to improve user QoS have been reviewed or analyzed. The resource scheduling method and detailed comparison between these algorithms have been described.

Index Terms **- Cloud computing, Cloud computing resource management, Cloud computing resource scheduling, resource management scheduling, management and scheduling algorithms.**

I. INTRODUCTION

"Cloud computing is an innovative technology that has brought a revolutionary change in the delivery of computing services. With the continuous development of the Internet and the Web, cloud computing has changed the way information and communication technology users access resources. By dynamically providing resources in a virtualized manner on the Internet, the focus can be shifted from local/personal computing to data center-centric computing. Cloud computing transforms the use of computing into a fifth public utility, just like traditional public services such as water, electricity, gas, and telephone [1], pay-per-use. Cloud Definition Computing Simply put, the term is an emerging technology trend that uses the Internet to provide computing in an abstract, virtualized, managed, and demand-driven manner, that is, processing, storage, services, networks, and applications. The resource that provides the service is located somewhere on the Internet, not on our local system. All of this is provided to end users like any other utility, and they can access these utilities anytime, anywhere with the help of the Internet. It provides users with a set of virtualized resources according to their needs, thereby reducing the burden on users to manage hardware, software, storage, and networks, thus incorporating the concept of flexibility [2]. For these reasons, cloud computing is now considered similar to the Internet. Figure 1 shows cloud computing resource management scheduling.

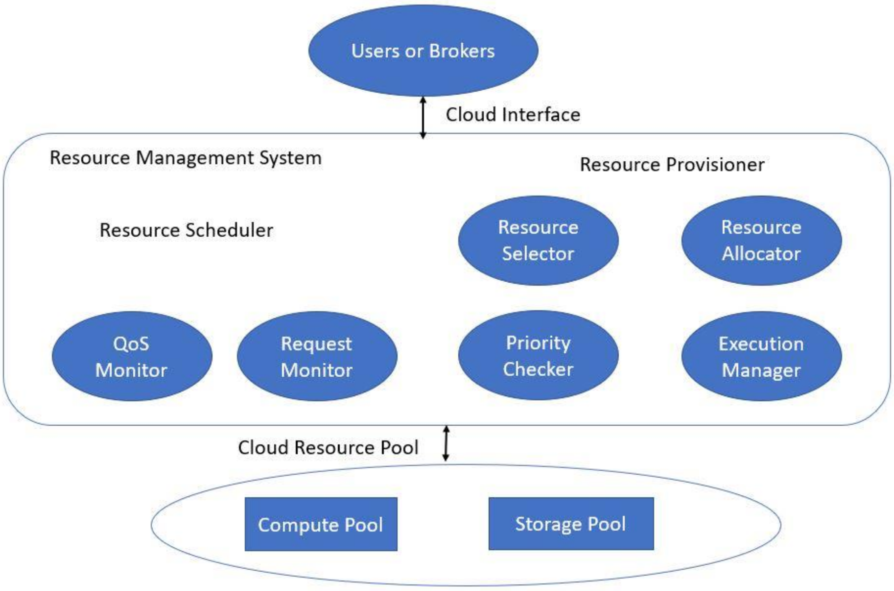

Fig 1: Cloud Computing Resource Scheduling

Although Cloud Computing derives from existing models such as Cluster and Grid, it exhibits distinct characteristics such as virtualization, heterogeneity, measured service and pricing, scalability, and scalability. resource recovery and aggregation. To provide such differentiated functionality, cloud computing faces many challenges such as security and privacy, resource planning, scalability and fault tolerance, energy efficiency, etc. , interoperability, etc. service for users to maintain their trust in the cloud. On the one hand, service providers aim to maximize profits and return on investment, on the other hand, users demand the cheapest, fastest and most reliable service. To meet the requirements of both purposes, an appropriate resource management mechanism is required. In this article, we focus on resource planning, which is one of the hardest problems in the cloud, on both the provider side and the user side." [31]

II. RELATED WORK

"As in recent years, the popularity of cloud computing is increasing, several researchers resolve RSP in the cloud computing environment and provide the same method. To worry is to design a planning algorithm that optimizes the quality of the service such as MaisPan, execution, deadline, reliability, response time, migration costs and availability. Many surveys were conducted to integrate other approaches for the same solution. Bala et al. [3] The workflow describes the need to import a cloud for the execution of the existing algorithm that solves an example of an online banking system and solved the problem of workflow planning. Salot [4] reviewed the basic linear approach to resolve RSP. A wide range of Singh et al investigations. [5] Provisioning resources, as well as a reservation of resources, including resource reservations. They ranked them according to the parameters of QoS such as the cost, time, profit, priority, ALS, energy, etc., but they have ranked various approaches of RSP, but the cloud computing architecture is Performed by Wadhonkar [6] Another survey was another survey. Led. Plan for RSP." [31]

III. NEED FOR RESOURCE MANAGEMENT SCHEDULING

"Cloud computing not only allows users to move their data and calculations to a remote location with minimal impact on system performance, but also ensures easy access to the cloud environment to access their data and get calculations. anytime and anywhere.

Cloud computing tries to provide easy and cheap access to compute resources that can be measured and billed against other models like distributed computing, grid computing, etc. In a cloud computing environment, tasks are distributed to separate compute nodes. For cloud resource allocation, redundant nodes capable of compute are detected and analyzed for network bandwidth, line quality, response time, cost of tasks, and allocation reliability. Therefore, the quality of cloud service can be described by resources like network bandwidth, full time, task cost and reliability, etc.

The cloud is a type of parallel, distributed system made up of a collection of interconnected and virtualized computers. These computers are dynamically provisioned and presented as one or more unique computing resources based on service level agreements, established by negotiation between service providers and consumers. Computing resources can be dynamically allocated based on user needs and preferences.

Resource management and resource scheduling both are important now a days because of high demand of cloud computing in Computer Science field. High demand of cloud computing has led to the need of managing resources which can be done through resource scheduling." [30]

IV. SCHEDULING ALGORITHMS

There are many types of scheduling algorithms in distributed computer systems. Most of them can be applied in a cloud environment with the right checks. The main benefit of the job scheduling algorithm is that it achieves high compute performance and excellent system throughput. Traditional task scheduling algorithms cannot provide scheduling in a cloud environment. According to the simple classification, task scheduling algorithms in cloud computing can be classified into two main groups: batch mode heuristic scheduling algorithms (BMHA) and online mode heuristic algorithms. In BMHA, tasks are queued and aggregated into sets of as they arrive in the system. The scheduling algorithm will start after a fixed amount of time. The main examples of algorithms based on BMHA are Scheduling Algorithm Come First Served (FCFS), Algorithm Round Robin (RR), Algorithm Min-Min and Max-Min. Thanks to the heuristic scheduling algorithm in online mode, tasks are schedule when they enter the system. Because the cloud environment is a heterogeneous system and the speed of each processor changes rapidly, the inline-mode heuristic planning algorithms are better suited to the cloud environment.

4.1 First come First-Served Scheduling

The first come, first served (FCFS) scheduling method simply assigns work based on their arrival time. The CPU will be assigned to the job that is first in the ready queue. The faster the work receives the CPU, the shorter the job's arrival time. If the burst time of the first process is the longest among all the tasks, FCFS scheduling may produce an issue of hunger. Figure 2 shows example of FCFS scheduling.

Process	Arrival Time	Burst Time
P1	0	10
P2	3	5
P3	5	2
P4	6	6
P5	8	4

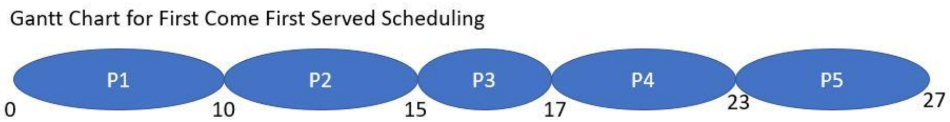

Fig 2: Example of FCFS Scheduling

FCFS has several advantages

- Simple or Easy
- It's a first-come, first-served basis.

FCFS has several drawbacks

- The procedure will run to completion since the scheduling mechanism is non-preemptive.
- The problem of starvation may arise due to the algorithm's non-preemptive nature.
- Although it is simple to build, it performs poorly since the average waiting time is longer than with other scheduling methods.

4.2 Shortest Job First Scheduling

Processes are scheduled using the SJF scheduling algorithm based on their burst time. The process with the shortest burst time in the ready queue is scheduled first in SJF scheduling. However, predicting the burst time required for a process is extremely difficult, making this method challenging to apply in a system. Figure 3 shows example of SJF scheduling.

Process	Burst Time
P1	15
P2	4
P3	5
P4	1

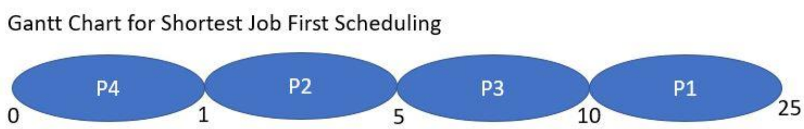

Fig 3: Example of SJF Scheduling

SJF Benefits

- Maximum throughput
- Average wait time and turnaround time are kept to a minimum.

SJF's disadvantages

- It's possible that you'll have to deal with a hunger problem.
- It can't be implemented since a process's precise Burst time can't be predicted in advance.

4.3 Priority Scheduling

Priority Scheduling is a way of prioritizing processes for scheduling. The scheduler uses this method to choose which tasks to work on based on their priority. Jobs with higher priorities should be completed first, whereas jobs with equal priorities should be completed in a round-robin or FCFS fashion. Prioritization is determined by memory needs, time constraints, and other factors. Figure 4 shows example of priority scheduling.

Process	Burst Time	Priority
P1	21	2
P2	3	1
P3	6	4
P4	2	3

Gantt Chart for Priority Scheduling

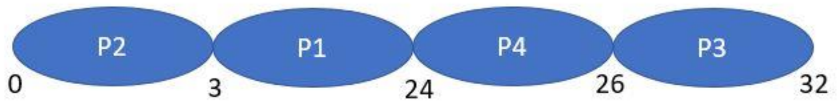

Fig 4: Example of Priority Scheduling

Priority scheduling has several advantages

- Scheduler that is simple to utilize Because processes are prioritised, those with the highest priority do not have to wait as long, saving time.
- This technique gives a suitable framework for determining the relative importance of each operation.
- Suitable for applications that have a changing time and date.

Priority scheduling disadvantages

- If the system crashes, all low priority processes are lost.
- If high-priority activities consume a large amount of CPU time, lower-priority operations may suffer and be delayed indefinitely.
- Some low-priority activities may be left waiting forever as a result of this scheduling method.
- When a process is ready to begin but must wait for the CPU because another process is presently executing, it will be blocked.
- If a new higher priority process is continually added to the ready queue, the process in the waiting state may be forced to wait for an extended period of time.

4.4 Round Robin Scheduling

The name of the algorithm comes from the polling principle, that is, everyone takes turns to get an equal share of something. It is the oldest and simplest programming algorithm, mainly used for multitasking. In Round robin scheduling, each ready task only runs in a circular queue for a limited time in turn. The algorithm also provides starvation-free process execution. Figure 5 shows example of round robin scheduling.

Process	Burst Time
P1	21
P2	3
P3	6
P4	2

Gantt Chart for Round Robin Scheduling

Fig 5: Example of Round Robin Scheduling

The following are the advantages / benefits of the Round robin scheduling method:

- You do not face the problem of starvation or the effects of escort.
- All jobs get a fair allocation of CPU.
- Handles all processes without any priority
- If you know the total number of processes in the execution queue, you can also assume the worst response time of the same process.
- This programming method does not depend on burst time. That is why it is so easy to implement in the system.
- Once a process runs within a set of specified time periods, the process will be superseded and another process will run within the given time period.
- Allow the operating system to use context switch methods to save state from interrupted processes.
- Provides the best performance in terms of average response time.

Here are the disadvantages / disadvantages of using Round robin programming:

- If the operating system cutoff time is short, the processor output will be reduced.
- This method spends more time on context switching
- Its performance is highly dependent on the amount of time.
- Cannot set priority for process.
- Round Robin scheduling does not give special priority to the most important tasks.
- Reduced understanding
- A shorter amount of time will result in more context switching overhead on the system.
- In this system, finding a correct time slice is a very difficult task.

V. COMPARISON BETWEEN EXISTING RESOURCE MANAGEMENT SCHEDULING ALGORITHMS

5.1. Determinism / Linear Method Used for Resources Programming

"In this subcategory of work, we discuss various algorithms that use linear methods, such as priority, integer programming, etc. Table I provides a brief comparison and description of these algorithms Cao et al. [8] provided an activity-based costing method, ie the number of any program / job does not determine its complexity, so number-based cost drivers will not produce accurate results. A better solution is to measure the usage cost of each resource (CPU, memory, I/O) generated by the job while it is running.

This will evenly distribute the cost of the disturbance and help generate accurate costs and more profit. Fang et al. [9] Provides a load balancing method for resource scheduling by considering cloud characteristics such as elasticity and flexibility. Your method applies to two levels: one is used to schedule tasks for the appropriate virtual machines, and the second is used to schedule virtual machines on the appropriate hosts so that the load is evenly distributed across the two levels. Consider the dynamic characteristics of the task and complete the virtual machine migration (if necessary) according to the needs of the task. They focus on faster response time, reducing time to completion and improving resource utilization. The time interval here refers to when the last submitted task ran successfully.

Xindu et al. [10] Two simple algorithms were proposed to reduce task completion time. The two algorithms are Shortest Cloudlet to Fastest Processor (SCFP) and Longest Cloudlet to Fastest Processor (LCFP). Cloudlets, in the cloud environment, refer to submitted tasks. SCFP, as the name suggests, organizes tasks according to task duration and organizes processors according to throughput. Then, it maps the tasks from the ordered task list to the ordered processor list. LCFP works in a similar way, except that the tasks are arranged in order of decreasing length. In the experiment, it was found that LCFP performed better than SCFP and FCFS. Gambari et al. [11] The priority-based job scheduling algorithm (PJSC) is used. They used Analytical Hierarchy Process (AHP) that is a multi-attribute and multi-criteria decision model to calculate the priority vector and comparison matrix, and its value determines the correct

Assign tasks to correct resources. Considering this consistent comparison matrix, they conducted numerical simulations with the goal of reducing production capacity while maintaining priority.

Li [12] designed a random integer programming model that uses the minimized geometry Buchberger algorithm (MGBA, an extension of Grobner base is designed to meet's various SLA constraints, such as cost, cost, performance, and delay. Numerical simulation was carried out, and the optimal solution was obtained in a reasonable time. Wu et al. [13] proposed a QoS-based task scheduling, the objective of which is to execute higher priority tasks on resources that occupy the least time possible. The priority is determined based on special QoS attributes. The algorithm is compared with the MinMin algorithm and Berger's model, and it is found that the completion time of the proposed method is better than that of the previous two.

Abdullah [14] proposes another multiQoS job scheduling method, which uses divisible load theory (DLT). DLT is designed to evenly distribute the load among existing machines, thus minimizing the overall job completion time. Using this method is aimed at optimizing the completion time (from the user side) and the total cost (from the provider side), and it is found that the result increases as the number of processors increases.

Thomas et al. [15]. They developed an algorithm to calculate task credits based on the duration and priority of the task. By looking for the difference between the actual duration of the task and the average duration, a credit calculated by length is found. Priority credit refers to the quotient of the priority of each task and its respective split factor. Total credit is the product of duration credit and priority credit. Compare the results with a credit system that uses duration and priority individually and then collectively. The collective approach was found to perform best in achieving the minimum time to completion.

Devi Priya et al. [16] Provides an improved maxmin method to find a task scheduling solution that considers the completion time of all tasks, not just the execution time of the current task, that is, it aims to change the MaxMin algorithm by selecting resources that minimize the total completion time. This scheme performs better than the basic MaxMin algorithm.

Lakra et al. [17] designed a multi-objective task scheduling algorithm, which forms a non-dominated task list based on the priority assigned to the task according to QoS. The execution time of the proposed method is better than FCFS and priority scheduling strategy uses these deterministic methods to provide cloud computing users with a better programming environment and pay more attention to parameters such as execution time, priority, and cost.” [31]

Table I. Comparison of deterministic methods for resource scheduling

Algorithm	**Objective Criteria**	**Description**	**Experimental Environment**	**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)” [8]	Algorithm explanation	Traditional way of task scheduling
Task scheduling based on Load Balancing [9]	Makespan, Resource	“Considers users’ dynamic requirements by scheduling tasks to VMs with lightest load.” [9]	Cloudsim	Grid Environment Load Balancing Algorithm
LCFP, SCFP[10]	Makespan	“Considers computational complexity as the basis for making scheduling decision” [10]	Cloudsim	FCFS
Priority Based Job Scheduling [11]	Makespan	“Considers three level of priorities- scheduling, resource and job level.” [11]	Numerical Simulation	-
Stochastic Integer Programming [12]	Cost, Throughput, Latency	“Applies Grobner Bases Theory to optimize SLA based resource schedule.” [12]	Numerical simulation	-
QoS-Driven Task Scheduling [13]	Makespan, Average Latency	“Uses several QoS parameters to assign priority to tasks.” [13]	Cloudsim	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.” [14]	Numerical Simulation	-
Improved Max-Min [15]	Makespan	“Improves performance of Max-Min algorithm by considering completion time rather than individual execution time.” [15]	Numerical simulation	Max-Min Algorithm

Credit Based Algorithm [16]	Makespan	"Assigns credit to each task based on its length and priority." [16]	Cloudsim	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." [17]	Cloudsim	FCFS, Priority Scheduling

5.2. Evolutionary method of Resource programming

Because the programming issue is an NPhard problem, it is not possible to find the optimal solution using a linear method. As a result, natural-inspired algorithms have piqued the curiosity of researchers. The magnitude of the challenge has a quadratic connection with the complexity of the evolutionary method's algorithm. The most common evolutionary algorithms, such as the Genetic Algorithm, will be described in this section (GA), The work done by the Ant Colony Algorithm (ACO), Particle Swarm Optimization (PSO), and the BAT algorithm in a cloud computing environment. The strategies employed by such algorithms to solve RSP are briefly reviewed in Table II. Population initiation, fitness evaluation, iterative updating, and identifying the best solution are the three basic steps of any evolutionary technique. A chromosome is a possible solution in a genetic algorithm, and this initial collection of chromosomes is referred to as a population. The fitness function is assessed once the first population is generated, and the best chromosome is chosen using the "survival of the fittest" approach. Through crossover and mutation processes, these chromosomes create new children in the following step. This procedure should be repeated until the termination condition is fulfilled and the best solution is found.

Zhao et al. [18] created an objective function to reduce the maximum execution time of any job using this approach. The numerical simulation was run with two jobs and two resources, and a solution was discovered in a short amount of time.

Kumar et al. [19] produced a superior starting population than by replacing a random population assembly discovered while using the Minmin and Maxmin algorithms. This enhances the regular genetic algorithm's performance as well as a series of job sets. Gan et al. are another enhancement on the traditional genetic method. "A person who uses a simulated annealing (SA) technique in the development of genetic algorithms. Simulated annealing not only aids in the discovery of improved optimum solutions, but it also aids in the reduction of the risk of accepting a severe chromosome for the following group by regulating acceptance factors, which is beneficial. In this paper, the author weighted a total of 5 QoS parameters, including completion time, bandwidth, cost, and distance. The results were superior to general genetic algorithms. It is an optimization of the particle group, its work is similar to the genetic algorithm. The population sets are called as swarm, which is a collection of particles. The dimensions of each particle correspond to the task. Each particle is a solution that can determine movement in the best position in a particular direction, depending on the best local position and the best particle position depending on the best local position and the best particle position. The particles have their own speed and position. Each particle assesses its own performance, which is influenced by the outcome of two elements as well as social interactions. To get an optimum solution, the particles' speed and location are modified according to the physical conditioning function." [31]

"Used PSO designed an objective function to minimize the total cost of running workflows in a cloud computing environment. The total cost includes the cost of execution and the cost of transmission. Compare the results with the Best Resource Selection (BRS) algorithm. And found that the utilization rate is much better than the last method." [21] . "Zhan et al. uses a simulated annealing and PSO algorithm, and uses the fast search capability of simulated annealing to improve PSO convergence speed. The performance of this method is better than GA, SA and PSO. Guo et al. Brought another improvement." [22]. Who updated the rate equation and incorporated PSO into local search and crossover and mutation operations? " [23]. "Rodríguez et al. proposed time-bound resource scheduling and resource delivery strategy using PSO. Their goal is to minimize execution costs and meet deadline constraints." [24]. "The PSO algorithm is used as is. The only goal to achieve is the cloud computing environment. Since PSO generates continuous values, Li et al. Provided a method for discrete PSO values of" [25]. "They suggested that the resources should be numbered according to their computational speed, rather than just giving them random values. The result is better than the basic PSO algorithm.

Ant Colony Optimization uses ant and pheromone values to find the best solution. Initialize the population based on the pheromone value. Then in each generation, the performance pheromone and heuristic information are brought to help to the best resources are obtained for each task. Here, task is considered as every ant step. This algorithm is generally used for load balancing strategies. For resource scheduling, Banerjee et al." [26] "The ACO algorithm was improved by modifying the pheromone update mechanism. And the completion time of the tasks performed is greatly reduced. Compared to the basic ACO algorithm, it works better. Wen et al. Made another modification." [27] "Who will combine the benefits of PSO with ACO so that the optimal local solution is not misinterpreted as the optimal solution, that is, to avoid premature convergence. This solution performs better than basic ACO in terms of convergence speed and cost.

In addition to the above algorithms, more algorithms have recently been developed. One of them is the Cuckoo Search Algorithm (CSA) used by Navimipour and others to solve resource scheduling problems in a cloud computing environment." [28] "CSA uses flight behavior characteristics of animals, similar to Levy Flights, which can be successfully applied to optimization problems. It gives promising results in terms of speed and convergence. Raghavan et al. Used the BAT algorithm influenced by the characteristics of bats." [29] Used to solve workflow scheduling problems. Compared with the BRS algorithm, the execution cost is better.

Table II. Comparison of evolutionary methods for resource scheduling

Algorithm	Objective Criteria	Description	Experimental Environment	Results Compared
Genetic Algorithm [18]	Makespan	"Uses Genetic Algorithm for task scheduling. Considers time utilization and resource utilization." [18]	Numerical Simulation	Traditional algorithm
Improved Genetic Algorithm [19]	Makespan	"Uses Min-Min and Max-Min algorithm to initialize population of Genetic Algorithm" [19]	Cloudsim	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/" [20]	Numerical Simulation	Standard Genetic Algorithm
Particle Swarm Optimization [21]	Cost	"Considers collective minimization of both execution cost and transmission cost." [21]	Cloudsim	Best Resource Selection Algorithm
Improved PSO based Task Scheduling [22]	Execution Time	"Incorporates the process of simulated annealing in the PSO algorithm to improve convergence." [22]	Cloudsim	GA, SA, ACO, PSO
PSO based Heuristic [23]	Cost	"Improves PSO by adding crossover and mutation and SPV." [23]	Cloudsim	Basic PSO
Deadline based Resource Scheduling [24]	Cost	"Considers deadline satisfaction as the constraint. Uses basic PSO to optimize cost and time." [24]	Cloudsim	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" [25]	Cloudsim	Basic PSO
Modified Ant Colony Optimization [26]	Throughput	"Improves the pheromone updation method of basic ACO algorithm." [26]	Google App Engine, MS Mesh	Basic ACO
ACO PSO Resource Scheduling [27]	Execution Time	"Incorporates ACO algorithm with PSO to avoid premature convergence." [27]	Matlab	ACO Algorithm
BAT Algorithm [28]	Cost	"Uses a new metaheuristic algorithm called as BAT Algorithm to minimise cost of execution." [28]	Cloudsim	BRS algorithm
Cuckoo Search Algorithm [29]	Speed of convergence	"Uses a new metaheuristic algorithm called as Cuckoo Search Algorithm to increase speed of convergence." [29]	Matlab	-

VI. CONCLUSION

The rapid growth of cloud computing has necessitated careful management of the resources that deliver services to consumers. This idea is handled via resource scheduling. This article describes several strategies to solve resource scheduling problems. Extensive research has been done on methods that tend to improve users' QoS indicators. First, the linear strategy for solving the resource scheduling problem is introduced in detail in the literature, and then the evolutionary method for solving the problem is described. In each category, we analyzed the goals and methods of the current works, and then compared them in detail. In Cloud Computing, Resource Scheduling is still a new issue that has to be researched further.

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), 217264.

[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. 271277). 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 QoSdriven 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.

[30] Tian W, Zhao Y (2013) Resource Management and Scheduling in Cloud. Optim Cloud Resour Manag Sched 3:1–6

[31] Bulchandani N, Chourasia U, Agrawal S, et al (2020) A survey on task scheduling algorithms in cloud computing. Int J Sci Technol Res 9:460–464