ENHANCED TASK SCHEDULING ALGORITHMS FOR CLOUD ENVIRONMENT

[1]Dasaradh Ramaiah.K, [2]Aarthi Priya.R, [3]Krishna Prasanna.M
[1]Professor & HoD,[2]Assistant Professor,[3]Assistant Professor
[1]Department of Information Technology
[1]B.V.Raju Institute of Technology, Medak, INDIA

Abstract:

Cloud Computing is a hardback standpoint for large scale distributed computing and parallel processing. Normally it serves computing as a utility service as pay per use. The users task/process which is submitted on to the cloud purely depends cloud serves performance and high efficiency. To improve the performance of the cloud services, scheduling plays vital role on users process/task. Task scheduling is one of the main types of scheduling performed. The brief study of different task scheduling method which is used mostly is task scheduling in the cloud computing. But in cloud computing environment the major problem which decreases the performance is task scheduling. To improve system performance of the cloud, over there need an efficient task-scheduling algorithm. The main focus of the current task-scheduling algorithms is task-resource requirements, CPU memory, execution time and execution cost. Nevertheless, these things will not consider network bandwidth. The virtual machines will act as scheduling machines while we doing scheduling the task. For maintain the correct load on the processor regarding network bandwidth and increase the usages of the processor and more over to maintain the efficiency to decrease the task execution time(Minimum-Execution-Time) by applying scheduling algorithms in cloud environment. By providing the scheduling algorithms onto the cloud, it will control the completion time.

Keywords: Task scheduling; Minimum-Execution-Time, Completion time.

I. Introduction:

Cloud computing is a typical structure for enabling omnipresent, commodious, on-demand access to a shared network containing a pool of adaptable computing resources that can be easily provided and released with minimal management effort or service provider interaction .Presently over the internet cloud computing provides dynamic services like applications, data, memory, bandwidth and IT services etc. scheduling of tasks is one of the various factors provides the reliability and performance of cloud services. Mainly the scheduling can be done at different levels but the important level either task level or resource level or workflow level. According to this paper we are primarily focusing on the task scheduling approaches. Computing jobs, named task are the two major tasks send request by the users to the data center. A task is a small piece of work that should be executed with in a given period of time. To the cloud provider depends on the resources available cloud user's tasks will dispatches by task scheduling. To increase the overall enactment of cloud depends on different parameters by scheduling the tasks. A task may include entering data, processing, accessing software, or storage. Task scheduling can be studied within the computer operating systems Most of them are suitable and can be applied to cloud environment with some modifications.

The main job of scheduling algorithm figure out first requirement basis like First Come First Serve (FCFS) With FCFS [1],if some jobs are waiting for a big job to finish, convoy effect that is the main problem of FCFS has been occurred. The line of processes may effect results like in longer waiting time and lower utilization of resources Scheduling based genetic algorithms are proposed where this algorithms try to enhance the energy consumption, carbon dioxide emissions and the profit of a geographically distributed cloud computing infrastructure. Particle swarm optimization (PSO) technique is proposed [5]. It is built for manipulating job scheduling in cloud environments. By varying the weight of the velocity equation it will modifies the basic PSO algorithm. The weight is varied and decreased over the whole run. In the heterogeneous computing systems PSO-based scheduler to assign jobs It updates particles in a discrete domain by proposing position update mechanism that considers characteristics of discrete variables.

II. Literature Survey

Without managing burden of the underlying infrastructure cloud computing provides to customers, virtually unlimited pay per-use resources. To gain maximum profit cloud computing service providers is using these resources efficiently and to increase the performance. This leads task scheduling to be a core and challenging issue in cloud computing. Combinational optimization problems in the cloud computing is Dynamic task scheduling. In combinatorial optimization problems, we are searching for an object from a finite or possibly infinite set. The searching objects may be an integer number, a subset, a permutation, or a graph etc. Because the usage to the practical significance of combinatorial optimization problems, many algorithms to deal them, have been proposed. The proposed algorithms can be typically differentiated as either complete or approximate algorithms. In the specific time for every finite size of instances of the combinational optimization problem an optimal solution is ensured by complete algorithms. In approximate methods, promise of finding optimal solutions has been forewent for the aim of getting good solutions in a securely reduced amount of time. Approximate methods may be prescribed as constructive methods or local search methods. Constructive algorithms produce results by toting components to a firstly empty partial result until a result is complete. Local search algorithms start from initial solution and iteratively try to replace the current solution by a better solution in a suitably defined neighborhood of current solution.

2.1 Max–.Min algorithm:

In the Max-Min algorithm chooses bulky tasks to be executed initially , which in turn small task delays for long time[2]. Shortest job First (SJF)[6] also known as Shortest Job Next (SJN) or Shortest Process Next (SPN) is a scheduling technique that selects the job with the smallest execution time. The jobs are queued with the smallest execution time placed first and the job with the longest execution time placed last and given the lowest priority. In the round robin scheduling, processes are dispatched in a FIFO manner but are given a limited amount of CPU time called a time-slice or a quantum. The CPU is pre-empted and given to the next process waiting in a queue, if a process does not complete before its CPU-time expires.

2.2 Round Robin:

Round-robin (RR) is one of the algorithms employed by process and network schedulers in computing. Round-robin scheduling is simple, easy to implement, and starvation-free. Round-robin scheduling can also be applied to other scheduling problems, such as data packet scheduling in computer networks.

2.3 FCFS

First Come First Serve, is just like FIFO(First in First out) Queue data structure, where the data element which is added to the queue first, is the one who leaves the queue first. This is used in Batch Systems. It's easy to understand and implement programmatically, using a Queue data structure, where a new process enters through the tail of the queue, and the scheduler selects process from the head of the queue.

2.4 SJF

Shortest job next (SJN), also called as shortest job first (SJF) or shortest process next (SPN), is a scheduling policy that chooses for execution the waiting process with the smallest execution time. SJN is a non-preemptive algorithm. Shortest remaining time is a preemptive variant of SJN.

III Related Work

3.1 Simulation

Our Application should have a Datacenter Broker that implements the Task Scheduling Policy. The different Entities in our application are:

1.The Respective Algorithm

2.Broker

3.Datacenter Creator

4.Vms Creator

5.Tasks or Cloudlet Creator.

The Algorithm entity creates the Datacenter ,Vms and Cloudlets or Tasks by calling respective other entities. Once all of them are created, Vms and Tasks are submitted to the Broker.

Broker : This is a Datacenter Broker and it schedules the tasks to the Vms on the basis of respective policy. The tasks are got from the Entity.

Datacenter Creator : This entity is used to create Datacenter(s).

VmsCreator : This entity creates the specified number of Vms given by the user.

Cloudlet Creator : This entity creates the specified number of tasks. Thus, we can perform Tasks.

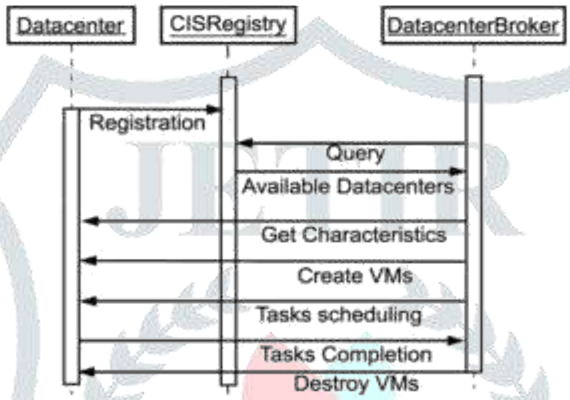

Fig 3.1.1.Sequence Diagram of Any CloudSim Application.

3.2 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling. The solution is represented as an array of VM's IDs representing the order of VMs. The first task will be mapped to the first ID in the array of VM's IDs; the second task will be mapped to second ID and so on. The PSO algorithm tracks solutions that have the best quality, founded by any of the used particles[3].

Allocation to the VMs on the basis of Algorithm[9]

PSEUDOCODE:

Input: List of Cloudlet (Tasks) and List of VMs
Output: the best solution for tasks allocation on
VMs Steps:

1 Initialize:
Set value of parameters Number_ of_particles, V_Max,tmax. **Set** t=1.
Set gBest=null.
Generate random solution for each particle

2. **For** each particle
{
Calculate solution fitness value
If the fitness value is better than pBest
Set pBest = current fitness value
End If
If pBest is better than gBest
Set gBest = pBest
End If
}

3. **Sort** particles by their pBest, from best to worst

4. **For** each particle

{
Calculate particle Velocity **Use**
Velocity to update particle
}

5. **Increment** t by one.

6 .**If** (t <tmax,)

Gotostep 2

Else

Print gBest.

End If

7 **Return**

Working Procedure :

Initially, this algorithm assigns random positions to all particles in search space. It advances the position of each particle successively based on its velocity using the global best known position and the best position known to a particle. Over time, the particles get together around optima or several optima. The algorithm keeps track of global best known position (gBest) and stopping value indicating when the algorithm should stop. Each particle consists of data that represent a solution, personal best (pBest) value and velocity value denoting how much the data can be modified. The velocity value is calculated according to how particle's data is out of the way from the target.

3.3 Max-Min Algorithm

Max-Min is a resource allocation and scheduling algorithm used in cloud and in grid computing environments to minimize makespan, cost and maximize profit and resource utilization. Max-min algorithm allocates task T on the resource R where large tasks have highest priority rather than smaller tasks. For example, if we have one long task, the Max-min could execute many short tasks concurrently while executing large one.

PSEUDOCODE:

Step 1: For all submitted tasks in meta-task.

Ti **Step 2**: For all resource Rj

Step 3: Compute Cij = Eij + rj

Step 4: While meta-task is not empty

Step 5: Find the task Tm consumes maximum completion time.

Step 6: Assign task Tm to the resource Rj with minimum execution time.

Step 7: Remove the task Tm from meta-tasks set

Step 8: Update rj for selected Rj

Step 9:UpdateCij for all Ti

Working Procedure:

This is done by selecting a task in the job list with the highest completion time on a resource that can execute it within a shorter period of time. The concern of this algorithm is to give priority to tasks with maximum completion time by executing them first before assigning other tasks that has minimum completion time. The disadvantage of Max-Min algorithm is that, the execution of task with maximum completion time first might increase the total response time of the system thus making it inefficient. Researchers on scheduling techniques in cloud computing have endorsed Max-Min as the best method of scheduling resources in cloud and grid computing. They have contributed tremendously by making Max-Min an efficient task scheduling algorithm. The efficiency of this algorithm has made cloud computing acceptable in both educational institutions and industries as a preferred platform for data storing and information dissemination.

3.4 Round Robin Scheduling

Round Robin is one of the most frequently used algorithms in resource allocation and task scheduling. Round Robin, a dedicated time slot is allocated to each job waiting in a queue to be scheduled by the scheduler . This means that, no task will be allocated to a resource for more than the allocated time and if a task is not able to complete scheduling within the allocated time, it will be preempted by another task and sent back in the queue to give way for other tasks[4].

PSEUDOCODE:

1. The scheduler maintains a queue of ready Processes and a list of blocked and swapped out processes.
2. The process control block of recently formed process is new to end of ready queue. The process manage block of terminating process is disconnected from the scheduling data structures.
3. The scheduler always selects the process control block at head of the ready queue.
4. When a running process finishes its portion, it is motivated to end of ready queue.
5. The event handler perform the following action,

a) When a process makes an input -output demand or swapped out, its process control block is isolated from complete queue to blocked/swapped out list.
b) When input-output process scheduled by a process finishes or process is swapped in its process control block is removed from blocked/swapped list to end of ready queue.

Working Procedure:

In task scheduling this algorithm is known as processor sharing algorithm and in this algorithm we consider the number of context switches is very high. It selects the load on random basis and leads to the position where some nodes are heavily loaded and some are evenly loaded. Though the algorithm is very simple but there is an added load on the scheduler todecide the size of quantum and it has longer average waiting time, higher context switches, higher rotate time and low throughput.

Process contains three stages:

- First , In which process is moving on for Execution
- Second , In which process is in the waiting in queue.
- Third, Migration with other process.

3.5 First Come First Serve (FCFS)

Analysis made from various researchers on resource allocation and task scheduling algorithms has identified First-Come-First-Served (FCFS) as the simplest scheduling algorithm as far as task scheduling and resource allocation are concerned. The default strategy implemented by the VM provisioned is a straightforward policy that allocates a VM to the Host in First-Come-First-Serve (FCFS)[7] basis. The disadvantages of FCFS is that it is non pre-emptive. The shortest tasks which are at the back of the queue have to wait for the long task at the front to finish .Its turnaround and response is quite low.

FLOWCHART:

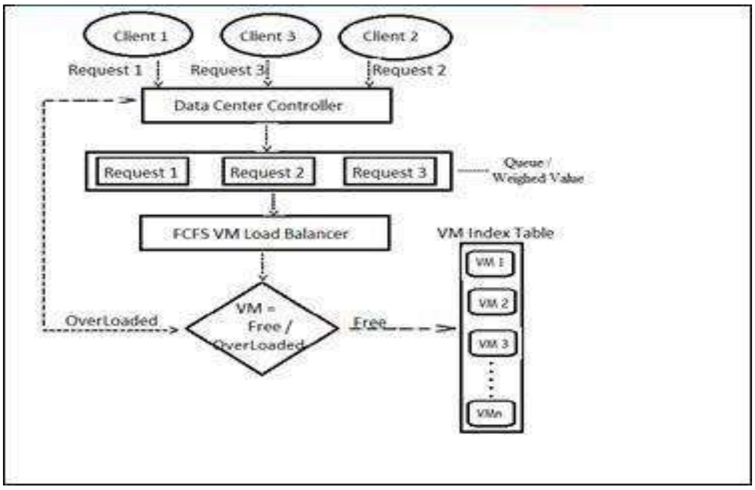

Fig 3.4.1 Flowchart of FCFS

Working Procedure:

The Cloud Sim toolkit supports First Come First Serve (FCFS) scheduling strategy for internal scheduling of jobs. Allocation of application-specific VMs to Hosts in a Cloud-based data center is the responsibility of the virtual machine provisioned component. The default policy implemented by the VM provisioned is a straightforward policy that allocates a VM to the Host in First-Come-First-Serve (FCFS) basis. For an Application in Java using CloudSim 3.0.3 Library, which simulates the Cloud and performs Tasks Allocation to the VMs on the basis of FCFS, First-Come-First-Served Scheduling Policy, in the Cloud.

3.6 Shortest Job First

A. SCHEDULING

Scheduling is also a fundamental function of operating system. Almost all Computer devices and resources are scheduled before use. As CPU is one of the primary computer resources. Thus its scheduling is central to O.S designs. Scheduling is the strategy by which the system decides which task should be executed at any given time. Whenever the CPU became ideal the O.S must select one of the processes in the ready queue to be executed.

B. SCHEDULING CRITERIA

1.CPU Utilization–This is a measure of how much busy the CPU is. The proper use of CPU then the scientific developed technique is used. CPU utilization can range from 0 to 100 percent.

2.Throughput-Throughput is the number of processes completed in how much per time period. For long processes this rate may be two processes per hour. For short transaction throughput might be 15 processes per second.

3.Waiting time-Waiting time is the sum of periods spends waiting in the ready queue. CPU executes only one process at a time. The rest of processes wait for the CPU.

4.Turnaround time-It is the total time taken to execute a process. The interval from the time of submission of a process to the time of completion is the turnaround time.

5.Response time-Response time is the time from the submission of a request until the first response is produces.

C. Better CPU Scheduling criteria are following

Maximize: Utilization and Throughput of the CPU.

Minimize: Turnaround Time, Response time, Waiting time and context Switch.

Scheduling Methods	Parameters Considered	Advantages	Disadvantages
First Come First Serve	Arrival Time	Simple Implementation	Doesn't consider any other criteria for scheduling
Shortest Job First	Service Time, Arrival Time	The Throughput is increased because more processes can be executed in less amount of time.	Can cause process starvation for longer jobs.
Round Robin	Arrival Time Time Quantum	Less Complexity	Pre-emption is required
PSO	Bird Flocking	Time sharing	When the algorithm converges, the fixed values of the parameters might cause the unnecessary fluctuation of particle.
Max-Min	Makespan Expected compilation time	Better Makespan compared to others	Poor load balancing

.

Table 3.6.1:Comparison Table of Different Task Scheduling Algorithms

IV Results

The experiments are carried out with 5 Datacenters with 5 VMs and[0-15] tasks under the simulation platform.

Entity Type	**Parameters Value**
Task	Length of Task 1000-20000 Total number of Tasks[0-15]
Virtual Machine	Total No of VMs[0-15] MIP-250 Virtual Memory(RAM) 128-2048 Virtual Bandwidth 250-500 Cloudlet Scheduler Space shared and Time Shared Number of PEs requirement[1-4] Number of Datacenter 5
Datacenter	Number of Host[2-6] VmScheduler Space shared and Time shared

Table 4.1: parameters setting of cloudsim

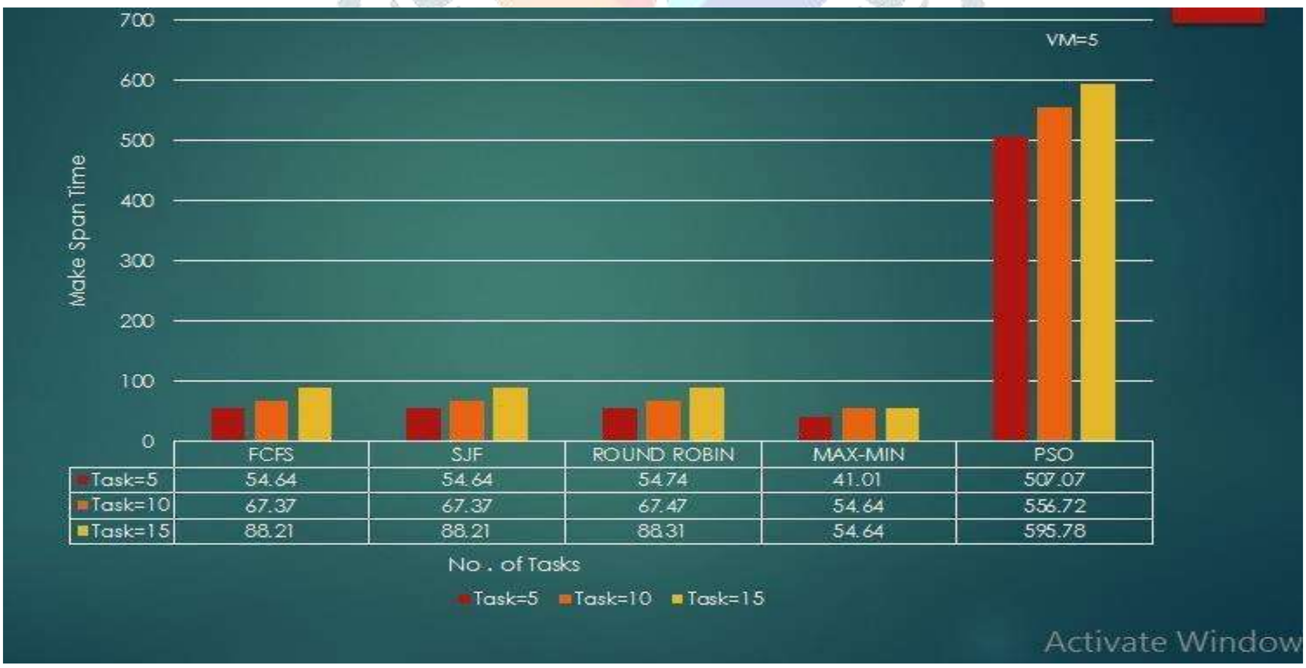

Fig 4.1: Pictorial Representation of Task Scheduling Algorithm

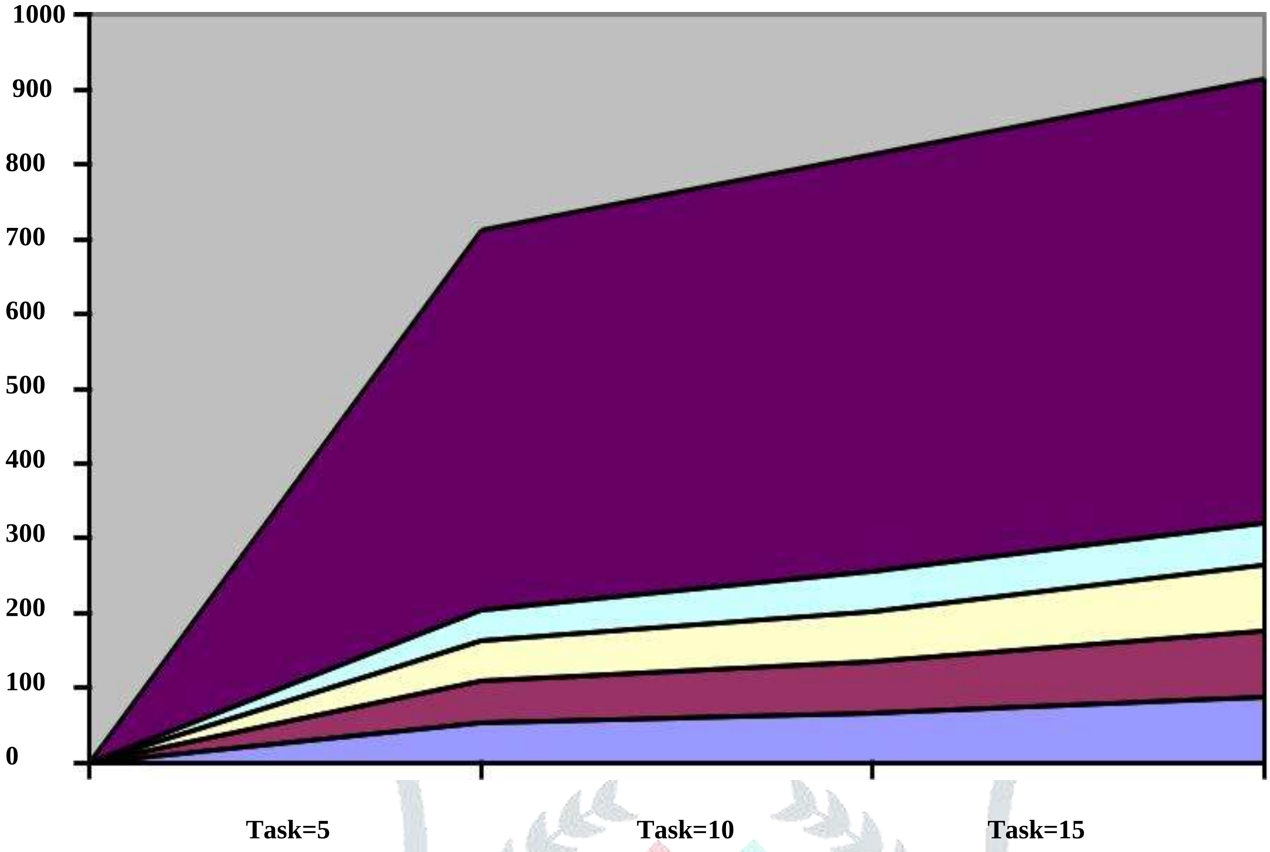

Fig 4.2: Pictorial Representation of Task Scheduling Algorithm

V. Conclusion and Future Enhancements:

Efficient scheduling algorithms always play a significant role in the performance provided by a cloud computing system. A study of existing task scheduling algorithms is done A brief analysis of each method is done and most algorithms perform scheduling based on one or two parameters. A better scheduling algorithm can be developed from the existing methods by adding more number of metrics which can result in good performance and outputs that can be deployed in a cloud environment in future. The table created, consolidates all the different scheduling parameters used in the existing scheduling algorithms. A good scheduling algorithm must consider the requirements of users satisfying their needs provided in SLA and at the same time beneficial to the cloud providers. Combining different parameters such that to obtain an efficient scheduling algorithm and improve the overall performance of the cloud services can be done as an enhancement.

VI. References:

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

[2] Braun, T.D., Siegel, H.J., Beck, N., Bölöni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D. and Freund, R.F., 2001. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. Journal of Parallel and Distributed computing, 61(6), pp.810-837.

[3] Elhady, G.F. and Tawfeek, M.A., 2015, December. A comparative study into swarm intelligence algorithms for dynamic tasks scheduling in cloud computing. In Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on (pp. 362-369). IEEE.

[4] Maniyar, B. and Kanani, B., 2015. Review on round robin algorithm for task scheduling in cloud computing. Journal of Emerging Technologies and Innovative Research, 2(2), pp.788-793.

[5] Pacini, E., Mateos, C. and Garino, C.G., 2014. Distributed job scheduling based on Swarm Intelligence: A survey. Computers & Electrical Engineering, 40(1), pp.252-269.

[6] Azmi, Z.R.M., Bakar, K.A., Abdullah, A.H., Shamsir, M.S. and Manan, W.N.W.,2011. Performance comparison of priority rule scheduling algorithms using different inter arrival time jobs in grid environment.International Journal of Grid and Distributed Computing, 4(3), pp.61-70.

[7] ManiyarBhumi, J., 2015, March. REVIEW ON ROUND ROBIN ALGORITHM FOR TASK SCHEDULING IN CLOUD COMPUTING. In International Journal of Emerging Technologies and Innovative Research JETIR (Vol. 2, No. 3 (March-2015)). JETIR..

[8]Belalem, G., Tayeb, F.Z. and Zaoui, W., 2010, October.Approaches to improve the resources management in the simulator CloudSim. In International Conference on Information Computing and Applications (pp. 189-196). Springer, Berlin, Heidelberg.