Critical review of scheduling algorithms to optimize datacentre energy consumption and environmental impact towards green cloud computing

MSc Research Project
MSc In Cloud Computing

Pankhuri Jha
Student ID: X21109460

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Pankhuri Jha
Student ID:	X21109460
Programme:	MSc In Cloud Computing **Year:** 2022-2023
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	15/12/2022
Project Title:	Critical Review of Resource Scheduling Algorithm to optimize datacentre energy consumption and environmental impact towards green cloud computing
Word Count:	6486 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Pankhuri

Date: 15/12/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Critical review of scheduling algorithms to optimize datacentre energy consumption and environmental impact towards green cloud computing

Pankhuri Jha
21109460

Abstract

Computing research has increasingly migrated toward green computing. The wide use of technology presents certain environmental issues such as excessive power use and a growing carbon footprint that harms the atmosphere. Therefore, for cloud service providers maintaining the Quality of Service (QoS) and service level agreements (SLAs) while reducing energy usage and resource productivity became very difficult.
The objective of this paper is to analyse scheduling algorithms to optimize resource utilization and to provide high-quality service in a cloud computing data centre which goes through effective resource scheduling algorithms to deliver the benefits of low energy consumption.
Hence, comparison of different resource scheduling like load based, temperature based, and other types of algorithms with respect to different parameters like execution time, response time, load balance and make span of job has been done to find the best resource scheduling algorithm. Therefore, in order to provide effective execution of user jobs a lot of effort has been put in determining the priority of jobs and timing constraints using python. Focus is put on allocating the relevant priorities while consuming less time and energy. With the analysis it could be said round robin scheduling algorithms can effectively increase resource utilization while reducing the energy consumption for job execution because it's better for interactive application as the response time is lesser.

1 Introduction

The "go green" approach has received significantly more focus in the mainstream researchers and media during the past year. As the customers increasing demands for cloud applications are pushing cloud service providers to build out their network of power consuming data centres. These data centres need a tremendous amount of power to operate. Consoles, monitors, network devices, CPU cooling fans, lighting, and cooling equipment all require energy which explains why the energy usage of the IT sector is rising. According to the most recent research, the total power consumed by data centres in 2020 was estimated to be 39 Giga Watts (GW), or around 63% more than in 2015. [1]

New demands and objectives are being driven by issues about global climate change, surging energy use, and power losses. Businesses all over the world are rushing to develop technologies and products that focusses both on improved efficiency and reduced power usage to solve these issues. The methods, strategies, and technology offered by green computing helps to support this innovation. The term green computing refers to resource-efficient computing. The green computing concept has been established to boost the effectiveness of resource utilization in the computing industry in response to the ongoing rise in power, cost, and awareness of global warming. One of the main strategies used in green computing to reduce power consumption while maintaining goal processing performance is power and energy aware scheduling. [2]

In Moore's Law it said that transistor count will double every two years due to decreasing transistor size and other advancements. In the fig. 1 shown below power consumption in computing can be seen. [2]

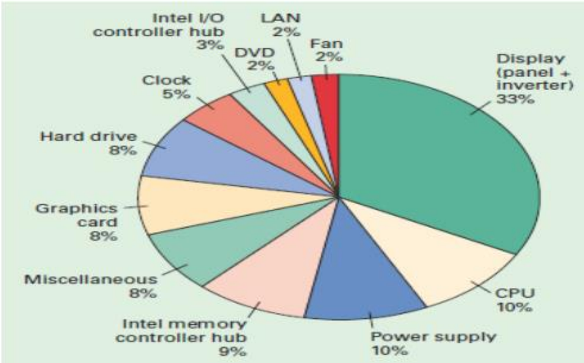


Figure1 : Average power consumption in CPU

It's known that supercomputers use and emit more energy hence, Additional cooling infrastructure is necessary, and it can cost as much as the supercomputer itself as well as using and looking after them involves higher facility costs as shown in fig 2 below [3]

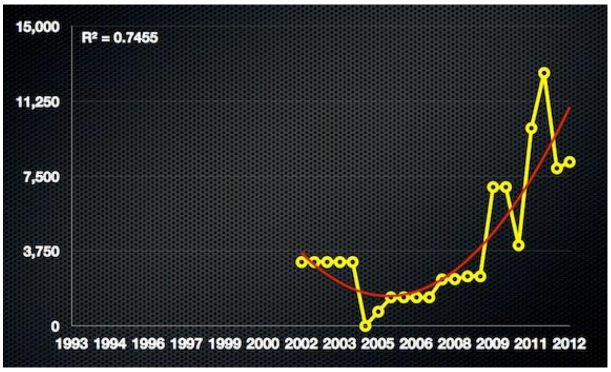


Figure 2: Energy consumption in super computer in KW over the years

The two most significant components that account for 86% of the server room's power usage are the cooling system and essential loads. ICT equipment in the USA uses around 9% of the nation's total energy usage, but that percentage is predicted to climb to 45% within ten years, according to research sources (DeForge and Whitney, 2014). ICT equipment not only uses a lot of energy but also generates a lot of Co2.The major global provider of ICT research and consulting services Gartner, predicts that in 2020 IT equipment will be responsible for 4% of all CO2 emissions worldwide (Gartner,2019). In this research paper, resource scheduling which is a crucial component of every cloud-based network is focused on. The scheduling of tasks is critical since all incoming data and tasks are handled in a specific order (which may be dependent on priority, machine availability, etc.) as they come in. In a cloud, scheduling algorithms are used to ensure that tasks are evenly distributed among the resources that are available and that the overall execution time of all tasks or jobs is as short as possible. After carrying out the necessary verifications, most of the scheduling algorithms used in a Distributed Computing System can likewise be employed in a Cloud Computing Environment.[5]

2 Related Work

The Objective of this section is the comparisons of the literature related to green computing and scheduling. The area and the issues that is shown in the literature review have been based on the overall study, that would be a scheduling algorithms analysis of green computing journals and papers with the ambition to examine the existing information and future perspective. The research is supported by previous explained papers of green cloud literature. This literature review starts with the description of green computing and later Scheduling analysis techniques has been studied with summary at the end stage.[6]

2.1 The term green cloud computing

Since Information Systems (IS) research has focused on green computing, different definitions of green IT have been developed. The process of developing policies and procedures that increase computing resource efficiency in order to minimise energy consumption and the environmental impact of their use is known as "green IT" or "green computing," according to MataToledo and Gupta (2011). (Baliga et al. 2011; Kabiraj et al. 2010). Trimi and Park (2013) define "green" as a composite term that refers to the environment and IT and addresses concerns such environmental pollution, energy use, waste disposal, and resource recycling. [11] One of the widely quoted and referred-to papers that helped to establish the idea of green computing is (Murugesan 2008). As seen in Fig. 3, the author has proposed four categories of green IT including green usage, green design, green manufacture and green disposal. [12]

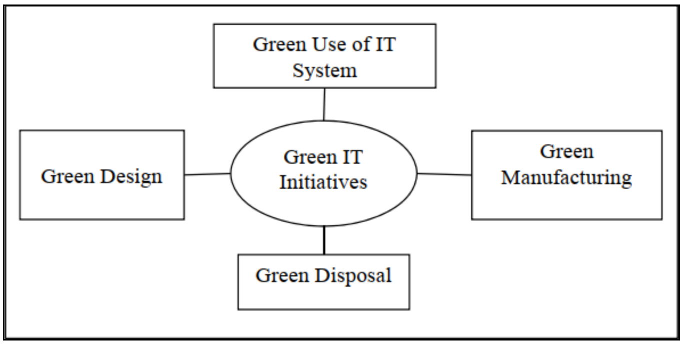


Figure 3: Domains of Green computing system

2.2 Various initiatives in green cloud computing

The initiatives vary from scientific solutions for more environmentally friendly IT that supports Green IT. The technical solutions concentrate on using, designing, manufacturing, and discarding energy-efficient gear (Murugesan 2008). By utilising energy-efficient hardware, enhancing airflow management systems to lower cooling needs, and taking into account virtualization technology, according to Kokkali et al. (2009), IT can be "Green" [9]. Chetty et al. (2009) investigate the huge potential for energy savings in the combination of new computer systems with improved power management and end user power management practises. Ansari et al. (2010) advocate achieving Green IT by managing e waste and a soft solution like printing on both sides inside the organisational level while explaining how e-waste is contributing to an environmental concern.[10]

2.3 Various benefits of green computing

Operating cost and environmental considerations are two of the main advantages of using green IT. Brooks et al. (2012) stated that there are "two broad types of benefits: environmental benefits and cost reduction benefits" when considering the advantages of green IT/IS. [13] According to the literature, according to Bose and Luo (2011), "the three main motivations of Green IT projects are: (1) decreasing costs owing to budget constraints, (2) reducing consumption due to resource restrictions, and (3) complying with local law." So, it is clear that green IT tries to save costs while also protecting the environment. [14]

2.4 Scheduling algorithm and it's analysis

Scheduling is an approach with which a certain workload can be operated and placed in a specific order and hence improved for the best outcomes. Across many fields, scheduling is done to organize the work-order. Scheduling can be static or dynamic. There are three main categories in cloud systems and i.e., task scheduling, resource scheduling, and workflow scheduling. Workflow scheduling is useful for controlling the completion of interrelated parts inside processes and for mapping out their workflow. Task scheduling is the method by which predetermined Jobs are assigned to the available machines for execution in a specific order which is chosen by the algorithm to ensure maximum efficiency. The steps in a task scheduling process are shown in the fig. 4, which is where the scheduling algorithm enters the frame.[6]

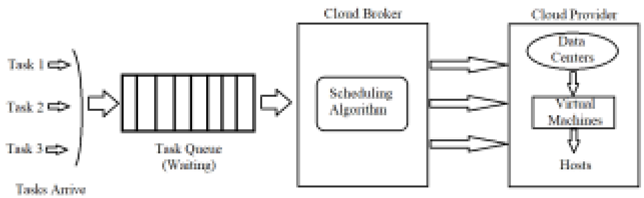


Figure 4: Scheduling of the task process

The term "resource scheduling" refers to the distribution of available resources among individual tasks. The system manages the prioritized queues of tasks awaiting CPU time and must choose which task to accept from the queue and how much time to allocate to it in order to make sure that all tasks are finished fairly and quickly. Scheduling techniques classify the scheduling algorithms into the three categories below:

• Temperature based scheduling algorithms

• Load based scheduling algorithm
• Other types of scheduling algorithms [7]
This paper mainly focusses on resource scheduling algorithms and their comparison. Few of the resource scheduling algorithm and their division is shown diagrammatically in the fig. 5.

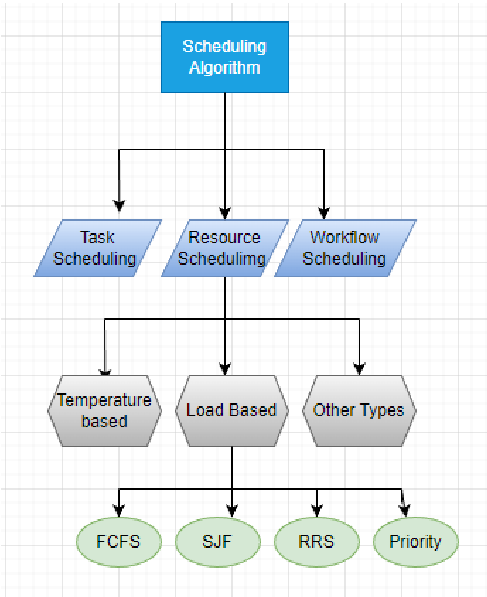


Figure 5 : Classification of Scheduling Algorithms

P. Guo, et al. in [8] covered the significance of Green Computing in cloud-based 5G networks where reducing scheduling fault tolerance is the whole purpose. Traditional fault-tolerant scheduling methods added redundant components to the system, which affects energy usage. As per the understanding with this technology, they offered a way to enhance scheduling in the cloud-based 5G network while lowering the energy consumption.
An energy-conscious green scheduler was presented by Kliazovich et al. (2012), which arranges the workloads in the machines with the least amount of processing. The scheduler regularly monitors the network switches buffer occupancy to ensure high-performance computing workloads.[9] When there is traffic, the scheduler avoids the crowded channels, even if they lead to servers that can handle the workloads processing demands. The power supply is then reduced for the less loaded servers and idle servers are put into sleep state or mode. Random scheduling algorithm distributes the jobs to the available servers at random. So, even if the server is carrying a load, no matter how heavy or small the status wcould have discarded. [9] Therefore, due to the possibility of a significant server load, this action would have resulted in the job being recorded as unsuccessful (Liu et al., 2013). All DC servers can receive jobs fairly distributed through the round robin (RR) scheduling technique. As a result, load balancing could be accomplished while avoiding congestion and delays.[10] Hence, unsuccessful tasks can be minimized (Mathew, 2014).

2.5 Scheduling in green computing summary and implications

This part discusses the history of green cloud computing and the scheduling algorithm that serves as the foundation for this research. In conclusion, green cloud is highly complex and before the deployment process gets started, the organization needs to understand several scheduling systems. When implementing the Scheduling algorithm, a variety of factors needs to be taken into account, from the cost of implementation and maintenance to the best way to maximize the results. As information trends and patterns have changed, it is clear that scheduling algorithms are needed in computing. Additionally, it is crucial that the gaps in the literature that have been found inspire and contribute to the completion of new work without the need for additional study.

3 Research Methodology

There are various research techniques available that can be used to lower data centres energy consumption:

1) Scheduling Analysis Case Studies, 2) History, 3) experiment and 4) survey

Scheduling analysis method has been used in this research methodology dependent on numerous aspects and under specific conditions. Our research strategy is primarily focused on creating an effective system in order to satisfy the requirements for green computing for the cloud computing system.[15] Reducing the amount of electricity used in data centres and the ratio of carbon emissions is key to creating a green computing environment.
Researchers have suggested a variety of technologies, structures, and strategies, including resource orchestration, migration, and consolidation techniques, as well as scheduling activities and resources. In the context of cloud computing, ineffective resource use has been identified as the primary driver of excessive electricity demand and high carbon emissions. Scheduling algorithms help alleviate the issue of ineffective resource usage. [16]

Effective resource scheduling is required to increase system performance when a lot of operations are run continually in a cloud environment. Several techniques are available for cloud task scheduling that link user tasks to resources for execution. Because cloud computing is still relatively new, researchers claim that traditional scheduling algorithms cannot handle the demands of the cloud. As a result, experts are working to modify conventional algorithms to handle cloud requirements like quick elasticity, resource pooling, and on-demand self-service. Various scheduling algorithms are considered in this study depending on execution time, resource utilisation, energy consumption, and service quality. (Syed,2016). The idea of virtualization, which uses the encapsulation process to run several logical (virtual) workstations on a single physical workstation (hardware device), has also been considered. The main advantages of virtualization include lower costs for cloud computing, lower energy use, and lower carbon emissions. To balance the load on processing nodes and improve resource utilisation, the notion of VM movement is applied in this article. (Archana,2019) [15]

3.1 Scheduling algorithms

3.1.1 Method: In the field of technology and computing, archival analyses are frequently used to examine research trends and uncover knowledge gaps (Eden et al. 2012; Tushi et al. 2014). Fig. 6 shows that identifying the data sources and creating a meaningful search string represent the data collecting step. The method for analysing is the classification framework, random function, which is used to make task length of 1000s of numbers, the specifics of how the future trends will be detected, and the methodology for selecting topics for additional research are all included in the data analysis segment.[17]

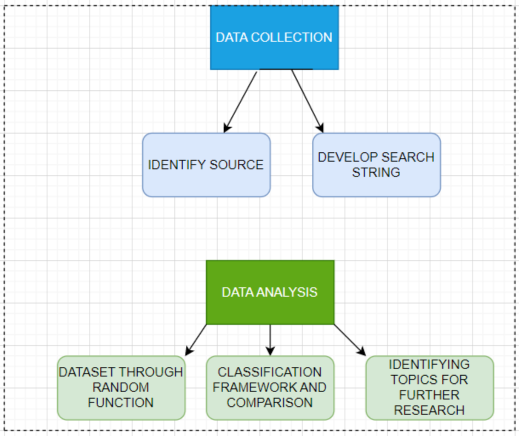


Figure 6: Method

3.1.2 Resource requirement: The client end often has access to this component. When a client submits many HTTP requests to a web service or application, the scheduler works to handle each request. The dynamic load on the system is produced by this component. The demand that must be met by the cloud computing system is mostly represented by resource requirements.[17]

3.1.3 Job Queue: In the dataset, it serves as a queue using FIFO technique. Before submitting to the VM servers, the tasks are provisionally located in the job queue. The quantity of requirements in the queue can be used to analyse the impending traffic in the cloud computing system.[18]

3.1.4 Load Balancer: In the suggested methodology, machine learning techniques have been used for balancing the load. In order to identify patterns and important information. The main goal of these algorithms would be to compare the algorithms. The resource prediction machine learning algorithm is depicted in the fig.7 below [18]. The temperature-based algorithm and the load-based scheduling algorithm are the two scheduling algorithms that have been used for prediction.

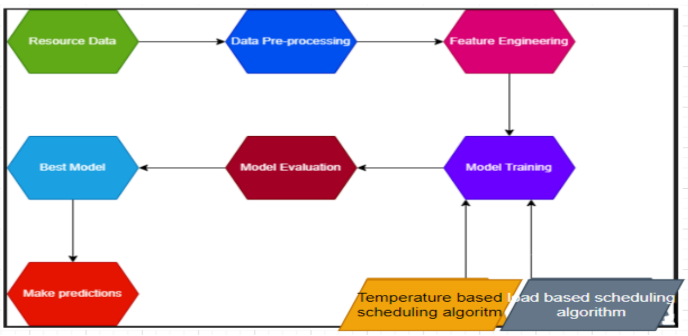


Figure 7: Prediction using Machine learning

3.1.5 Green strategies for compilers: Resource compilers examine software programmes while they run and rewrite software source code using several green design principles. The following green practises can be used locally, globally, or amongst procedures to make programmes more energy aware.[19]

A) Cache Skipping: Loops are very important in the programming environment. Due to the repetition of the same activity, replication in loops results in great performance but high energy consumption. Cache operations can be skipped during unneeded replication, which is a desirable strategy. In this method, the compiler must separate the blocks with a lower likelihood of execution, such as an exception block. This study shows that, in an ideal scenario, cache is not used, which reduces power consumption as a result of the technique.

B) Use of register operands: Each machine uses energy at a different cost while accessing memory resources. Most of the research demonstrate that memory reads, and writes are more expensive than using register operands. Register operands use less energy since they are less abstract than memory accesses (read/write).

4 Design Specification

4.1 Scheduling process: Cloud resource scheduling process is largely divided into three phases as shown in fig. 8: [20]

1) Resource discovery and filtering: The cloud service provider finds a list of the resources that are accessible in a particular network, gathers them, and evaluates their functionality.

2) The most crucial phase of task scheduling is resource selection, commonly referred to as the decision phase. Resources that are needed are chosen based on a specific parameter and in accordance with the specifications for task completion.

3) Task Submission: The task is submitted to the resource for execution after the necessary resource has been chosen.

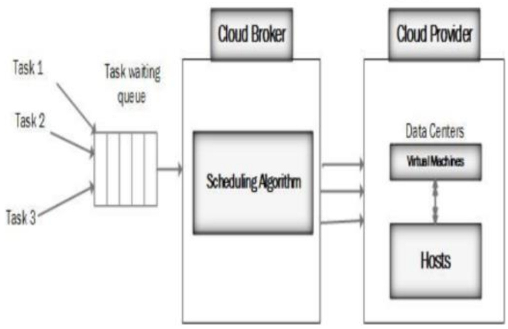


Figure 8: Task scheduling process

Scheduling Techniques separate the scheduling algorithms into the 3 types below:
i) Load based scheduling
i) Temperature based scheduling
iii) Other scheduling algorithms
Starting with load-based algorithms, all the algorithms would be taken into consideration.

4.2 Load Based Scheduling Algorithms

4.2.1 First Come First Serve: The method that makes the most sense and is the simplest is to let the first process that is submitted to get execute first. This method of scheduling is known as first-come, first-served (FCFS). When processes are submitted, they are in fact added to the end of a queue. Each process completes its execution before the next one is selected from the front of the queue. The CPU with the shortest burst time is given the task. A scheduler places the processes with the shortest burst times at the front of the queue and the ones with the longest burst times at the back. This calls for in-depth information or calculations of the length of time needed to accomplish a process[21]. The concept is shown in Fig. 9.

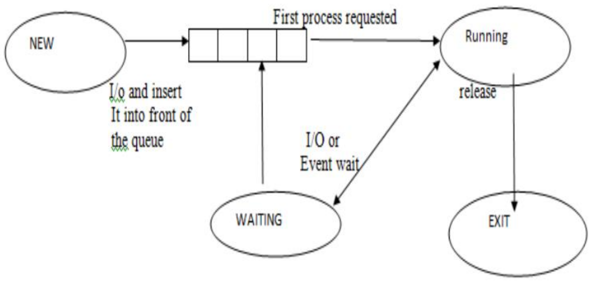


Figure 9: FCFS Scheduling

Characteristics:
1. There is no hunger since there is no priority; all processes eventually finish.
2. There is a high rate of turnaround, waiting, and response time.
3. Even if other processes burst times are too brief, the process with the longest burst time can control the CPU. As a result, throughput is low.

FCFS Algorithm:
• Start the tasks.
• Add tasks up to n numbers after the first task is added to the queue.
Add task "l," the following task, to the end of the main queue.

4.2.2 Round-Robin Scheduling Algorithm: For each job, a time slot is selected using this process. Only for that short of a period of time would one job be carried out before the next was started. A queue is maintained for the prepared processes. The scheduler moves through this queue, giving each process access to the CPU for the allocated quantum of time. The tail of the queue is expanded with new processes [21]. More information is provided in Figure 10 and 11.

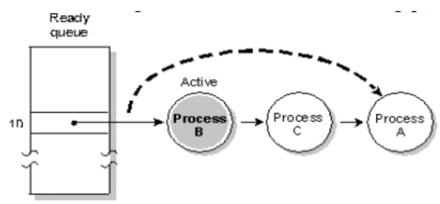


Figure 10: Round Robin Process

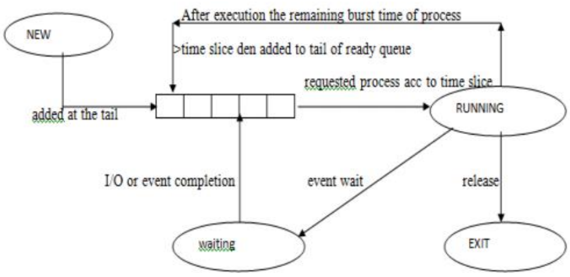


Figure 11: Round Robin Algorithm

Characteristics
1. Too short of a quantum setting results in too frequent context transitions, which reduces CPU effectiveness.
2. A too-long quantum setting may result in slow response times and approximations of FCFS.
3. Deadlines are rarely reached in a pure round robin system due to lengthy wait periods.

Round Robin Scheduling Algorithm

• The ready queue will continue to be a FIFO queue of processes.
  New processes will be added to the ready queue's tail.

• The CPU scheduler selects the first process in the ready queue, sets a timer for 1 time slot later, and dispatches the process.

• The CPU burst for the process might be less than one time quantum.
  - In this scenario, the process will freely release the CPU.
  - After that, the scheduler will move on to the subsequent process in the ready queue.

• In the absence of this, the timer will sound and the OS will be interrupted if the CPU burst of the presently active process lasts longer than 1 time quantum.
  - The process will be moved to the end of the ready queue when a context switch is carried out.
  - The following process in the ready queue will be chosen by the CPU scheduler.

4.2.3 Shortest job Scheduling Algorithm: The CPU with the shortest burst time is given the task. The processes with the shortest burst times are placed at the front of the line and those with the longest burst times are placed towards the back. This calls for in-depth information or calculations of the length of time needed for a process to finish. In the majority of situations, this algorithm is built for maximum throughput. The CPU with the shortest burst time is given the task. A scheduler places the processes with the shortest burst times at the front of the queue and the ones with the longest burst times at the back. In most cases, this method is built for maximum throughput. Fig 12 illustrates this concept.

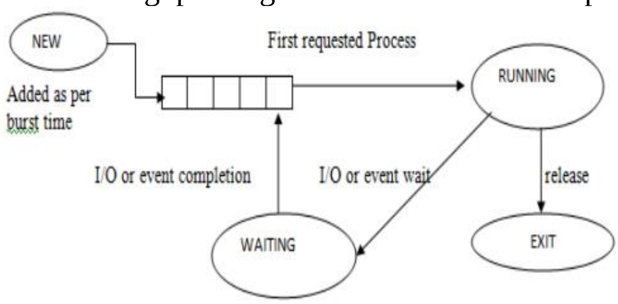


Figure 12: SJF Scheduling

Characteristics
1. Determining the length of the upcoming CPU request is where the SJF algorithm really struggles.
2. SJF reduces waiting times by servicing smaller processes before servicing larger ones. Although the average wait time is reduced, it may still handle requests with long wait times. When the ready list is full, processes with lengthy service times are more likely to be left on the list while smaller processes are handled. In the worst case scenario, processes with long service times won't ever be served while the system has little downtime[22].

Algorithm for Shortest Job First:
- for a = 0 to a < main queue-size

- if task a+1 length < task a length then
- add task a+1 in front of task a in

the queue
- end if
- if main queue-size = 0 then
- task a last in the main queue
- end if
- end for

4.2.4 Priority based Scheduling Algorithm: Every task in this algorithm is given a priority. The high priority tasks would be completed first, followed by the low priority tasks. Each process is given a defined priority rating by the O/S. Incoming higher priority processes can disrupt lower priority processes[23]. In Fig. 13, this concept is shown.

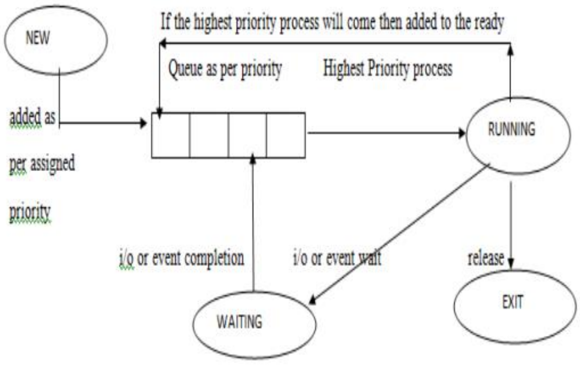


Figure 13: Priority based scheduling

Characteristics
1. A process with a low priority may starved.
2. The wait time for procedures with equivalent priority steadily lengthens.
3. Waiting and response times for higher priority processes are shorter.

Algorithm for Priority based scheduling

• for a = 0 to a <main queue-size
- if priority (task a+1) > priority (task a)

then
- add task a+1 in front of task a in

the queue
- end if
- end for

4.3: Temperature based Scheduling

Though it has quickly become a generally accepted framework for computing, cloud computing research is still in its development phase. Although cloud computing offers a number of cutting-edge capabilities, it nevertheless has certain drawbacks, including a comparatively high operational cost and environmental risks like rising carbon footprints. Effective scheduling of Cloud resources can help to some extent avoid these risks. The operating temperature of a machine can be used as a scheduling requirement for

virtual machines (VMs). A new proactive technique uses a temperature predictor to take into account the present and maximum threshold temperatures of Server Machines (SMs) before scheduling decisions are made, preventing the maximum temperature from ever being achieved. A variety of workload scenarios have been considered. The findings collected demonstrate that the proposed system performs better than existing VM scheduling systems that do not take current node temperature into account while making scheduling decisions. As a result, it has been confirmed that a Cloud environment requires less cooling systems[24].
A. **Core level scheduling**: First the scheduler determines the projected temperature of each core at the core level and moves the job from hot to cold cores accordingly.

B. **Schedules at the socket level**: A scheduler controls the job between sockets at the socket level. Information about fan speed, performance, and temperature is inputted.
The methods employed are dispersal and consolidation.[24]

```
IF (Event(Application_Arrival) ==TRUE)
  Put new request in waiting queue
  FOREACH physical machine in the Unified list
    If (current temperature < threshold temperature)
       Estimate  for each node and put  in the predict queue
    END IF
  END FOREACH
  Sort predict Queue in the decreasing order of
  FOREACH VM request in waiting queue
    Check availability of VM in the first node in predict queue
    IF Available then allocate VM to this VMRequest
      Update Unified and Local lists.
    ELSE
      Check next node in the Predict Queue till last node is
reached
    END ELSEIF
    IF no node is available
      Event(Activate_SM) and allocate VMRequest to a new
VM on this SM.
    ENDIF
  END FOREACH
ENDIF
IF (Event(Application_Complete) ==TRUE)
    Event(Free_VM)
    Update Unified and local lists
  IF(Active_VM_Count == 0)
    Event(Hibernate(SM)).
    Update Unified and Local lists.
  ENDIF
ENDIF
```

Algorithm 1 : Proactive Thermal scheduling

4.3 Other Types of Scheduling Algorithms

4.3.1 Ant Colony Algorithm: It follows the same bee algorithm. It was created and inspired by how ants navigate between their colony and a food source to find the best path through a network. Its applications include the following: Ant Colony Optimization using Fuzzy Logic. This technique gives ants fuzzy intelligence to speed up their capacity to search.

4.3.2 Bee Algorithm: It is an algorithm that draws inspiration from nature and attempts to follow bees as they search for food. To begin with, they choose scout bees to explore a large region. If a scout bee discovers a prospective food source, it returns to its hive and performs a waggle dance to signal other bees the location and distance of the resource. It is an algorithm that draws inspiration from nature and attempts to follow bees as they search for food. To begin with, they choose scout bees to explore a large region. If a scout bee discovers a prospective food source, it returns to its hive and performs a waggle dance to signal other bees the location and distance of the resource.[25]

It can be used for a variety of purposes, including the following:

- Forming manufacturing cells
- Training neural networks for pattern recognition.
- creating a production machine's work schedule.
- Constant problem solving and engineering optimization.
- Identifying several workable solutions to an early design issue.
- Multiple-objective optimization.

4.3.3 Greedy Capacity Algorithm: The list of virtual machines on the machine is kept in queue with tasks in queue1 and local agent. In addition to maintaining queue1 of submitted tasks, local agent also maintains the queue of available VMs on machines. This algorithm's main goal is to use every computing core available, which lowers power consumption. [25]

```
Begin
    while (true)
    {
      for( i = 1 to i <= queue1.length( ))
      {
        If  ci >= 1 && queue1.length( )>0)
        {
           If check capacity vm on ci
           {
              Schedule vm on ci AND task on VM
              ci – 1
            }
        }
      }
    }
End
```

Algorithm 2: Greedy capacity Algorithm

5 Implementation

The performance of 10 processes have been evaluated that arrive at the same time in the order listed below using several scheduling algorithms (specifically FCFS, SJF, RR, and Priority) with the specified priorities and a 9 ms time quantum. MATLAB has been used for the analysis to attain accurate results:

Table 1.2: List of process with the Priority and burst time

Process	CPU burst time	Priority
P1	0	5
P2	5	6
P3	3	1
P4	9	2

P5	6	4
P6	6	3

Gantt chart for FCFS as per process is:

P1 0	P2 5	P3 3	P4 9	P5 6	P6 6

Calculations of average waiting time and average turn around time:

Awaiting time: 10/6= 3.33 ms; Turnaround time = burst time + waiting time
Average Turnaround time: 4+6=10

```
PROBLEMS 20   OUTPUT   DEBUG CONSOLE   TERMINAL
PS D:\Eclipse Workspace\VSCode\Python> & C:/Users/DELL/AppData/Local/Microsoft/WindowsApps/python3.9.exe "d:/Eclipse Workspace/VSCode/Python/F
CFS.py"
FIRST COME FIRST SERVE SCHEDULING
[('P1', [0, 5]), ('P2', [3, 9]), ('P3', [6, 6])]
('P1', [0, 5])
('P2', [3, 9])
('P3', [6, 6])
Average Waiting Time:  3.3333333333333335
Average Turn Around Time:  10.0
PS D:\Eclipse Workspace\VSCode\Python>
```

Figure 14: Output of first come first serve scheduling algorithm

Gantt chart for SJF is:

P1	P2	P3	P4	P5	P6
1	8	6	4	5	0

Average waiting time= 5.2

Average turn-around time = 9.8

```
PROBLEMS 31   OUTPUT   DEBUG CONSOLE   TERMINAL
('P3', [6, 6])
Average Waiting Time:  3.3333333333333335
Average Turn Around Time:  10.0
PS D:\Eclipse Workspace\VSCode\Python> & C:/Users/DELL/AppData/Local/Microsoft/WindowsApps/python3.9.exe "d:/Eclipse Workspace/VSCode/Python/S
JF.py"
SHORTEST JOB FIRST SCHEDULING
{'P1': [0, 3], 'P2': [1, 8], 'P3': [2, 6], 'P4': [4, 4], 'P5': [5, 2]}
[[0, 3], [1, 8], [2, 6], [4, 4], [5, 2]]
[[4, 4], [5, 2], [2, 6], [1, 8]]
[[5, 2], [2, 6], [1, 8]]
[[2, 6], [1, 8]]
[[1, 8]]
Average Waiting Time:  5.2
Average Turn Around Time:  9.8
PS D:\Eclipse Workspace\VSCode\Python> []
```

Figure 15: Output of shortest job first scheduling algorithm

Gantt chart for Round robin scheduling algorithm:

P1	P2	P3	P4	P5	P6
0	9	5	8	9	6
P1	P2	P3	P4	P5	P6
12	16	19	22	29	32
P2	P4	P5	P6	P10	P1
62	70	76	88	91	93

```
PS D:\Eclipse Workspace\VSCode\Python> & C:/Users/DELL/AppData/Local/Microsoft/WindowsApps/python3.9.exe "d:/Eclipse Workspace/VSCode/Python/
R.py"
Round Robin SCHEDULING
Execution of process: ['P1', 'P2', 'P3', 'P4', 'P5', 'P1', 'P2', 'P4', 'P5', 'P2', 'P4', 'P4']
Average Waiting Time is  48.0
Average Turnaround Time is  68.0
PS D:\Eclipse Workspace\VSCode\Python> []
```

Figure 16: Output of round robin scheduling algorithm

Average waiting time= 48
Average turn-around time = 68

Gantt chart for Priority scheduling algorithm:

P8	P10	P9	P7	P6	P4	P5	P3	P1	P2
0	0	0	0	4		8	2	1	4

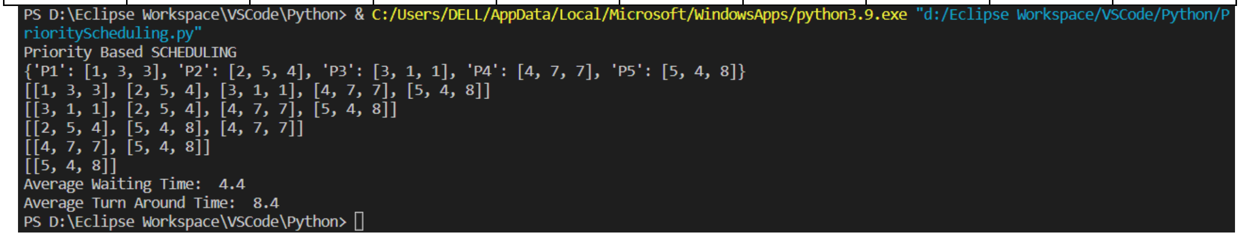


Figure 17: Output of Priority scheduling algorithm

Average waiting time= 4.4
Average turn-around time = 8.4

6 Evaluation

To compare the effectiveness of the algorithms, two fundamental metrics were gathered:

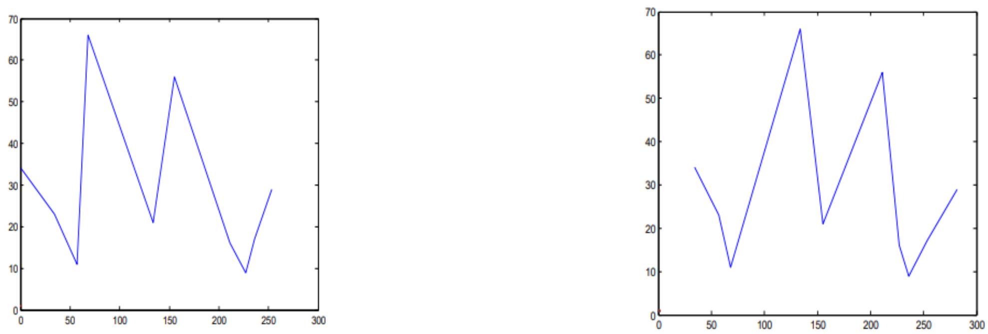
- The typical response time is: The total wait time + the sum of the execution times for each job. This indicator measures the scheduling algorithm's capacity to handle smaller, quicker jobs. The quantity of smaller, quicker jobs in the job stream makes the statistic a strong indicator of how well the scheduling algorithm is working.
- Weighted Average Response Time: A job's weight is determined by multiplying its resource requirements by the time it takes to complete. The result of adding the job weight and the job response time together and averaging it across all jobs is the weighted average response time. This essentially serves as a gauge for how efficiently the scheduling system moves major jobs forward. It also serves as a gauge of total resource usage over time.[26]

Different parameters can be used to construct a scheduling framework. The following requirements should be present in a good scheduling framework. It should centre on:
1) Load balancing and virtual machines and data centres energy efficiency
2) It should satisfy the security characteristics.
3) Resource allocation fairness plays a crucial part in scheduling.
Due to the impossibility of putting into consideration all metrics in one scheduling framework, the design becomes more complex, yet is used.

Table 1.1 Evaluating different resource scheduling algorithm based on different parameters

Algorithms	Methodology	Parameters	Merits	Demerits
First come first serve	This algorithm controls the FIFO queue used for task scheduling. The task that arrives first will be carried out on VM 1^{st}.	1. Arrival time	1. Simple and fast execution	Task scheduling relies decisions on arrival time and disregards all other factors, which results in reduced VM usage

Priority job scheduling algorithm	Dependence mode	Importance to individual queue	1. As the amount of time increases, the process significance rises. 2. Simple to use and approachable 3. ideal for time- and resource-intensive programmes.	1. If the crash happens, the jobs with the lowest priority will be lost. 2. Famine from lack of necessary supplies.
Greedy Algorithm	By picking each step according it seeks to determine the overall optimal solution.	1. Parameter u 2. Domain D 1. Population n	1. Simple to use. 2. Requires less resources 3. Scheduling happens quickly.	Universal optimization issue is not satisfied with this algorithm
Round robin scheduling algorithm	task in this algo has an equal of choose & executed in a unit that is smaller.	2. Arrival time 3. Time slice	1. Quick response times. 2. Time is in equilibrium. 3. less difficult	1. When a time slice expires, pre-emption causes the process to stop.



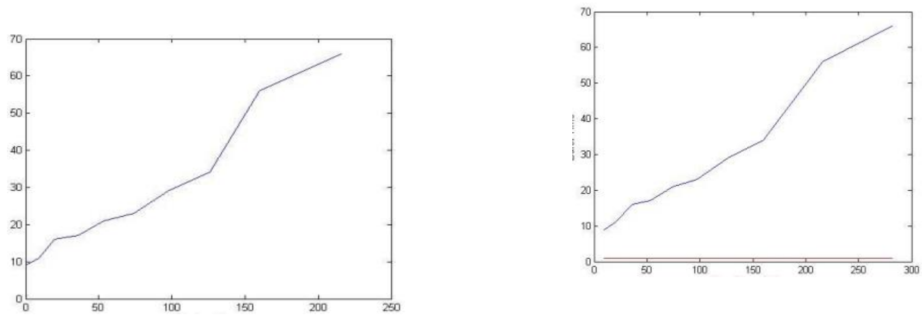
x-axis: Waiting time

y-axis : Burst time

x axis: Turn around time

y axis: Burst time

Figure 18: Graph to represent performance of the processes in FCFS



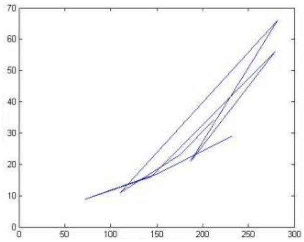
x-axis: Waiting time

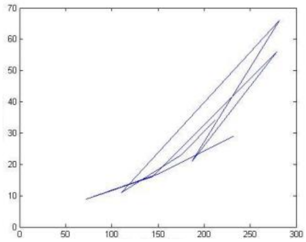
y-axis : Burst time

x axis: Turn around time

y axis: Burst time

Figure 19: Graph to represent performance of the processes in SJF





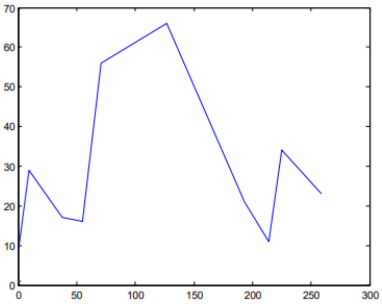
x-axis: Waiting time

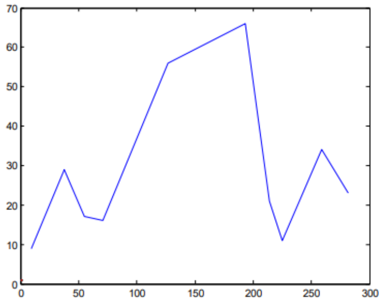
y-axis : Burst time

x axis: Turn around time

y axis: Burst time

Figure 20: Graph to represent performance of the processes in Round-robin





x-axis: Waiting time

y-axis : Burst time

x axis: Turn around time

y axis: Burst time

Figure 21: Graph to represent performance of the processes in Priority Scheduling

6.1 Discussion

Studying the various algorithms, we discover that Round Robin has the highest waiting time. While FCFS and SJF algorithms have shorter waiting times than Priority algorithm. The round robin Algorithm is the simplest Scheduling Algorithm when discussing complexity, which determines which sort of algorithm is simple or straightforward to utilise in processing. Though Round Robin Algorithm performs well regardless of time quantum size, SJF and Priority Algorithm become challenging to grasp and programme.

When using the FCFS algorithm, CPU is distributed to the processes in the order they arrive, whereas when using the SJF algorithm, CPU is distributed to the process with the shortest CPU burst time. The higher priority work can run first based on priority in the Priority Algorithm. [27] Delay occurs in the case of the Round Robin Algorithm after a predetermined amount of time.

7 Conclusion and Future Work

Conclusion: An effective scheduling algorithm can improve the performance offered by the cloud environment and give users with better services. In a cloud context, the major goal of resource scheduling is to maximise resource utilisation while minimising the number of resources needed to complete activities. This research article presents a review of various task and resource scheduling methods that are currently in use in a cloud setting. The approach for each algorithm has been briefly described, and most algorithms rely on one or two parameters. When evaluated and compared it has been found out that round robin algorithm is preferred over other as it's less complex and response time is lesser than others.

However, more metrics can be added to the present algorithms to produce more favorable results. Table 1 is based on many scheduling factors, including execution time, load balancing, performance, quality of service, response time, and ease of implementation. Load balance, response time, resource utilisation, and memory storage are the main issues with task and resource scheduling. By integrating several characteristics with already-existing algorithms, an efficient scheduling algorithm may be created, improving the overall performance of the cloud environment.

Future Work: Future research could consider creating an effective scheduler and classifying jobs into high- and low-priority categories based on their resource and energy usage. Along with this, a dependency mapping idea should be created so that dependent jobs are run immediately. The results should then be reviewed and executed appropriately and compared to earlier algorithms.

References

[1] Verma, Amandeep & Wadhwa, Bharti. (2014). Energy saving approaches for Green Cloud Computing: A review. 2014 Recent Advances in Engineering and Computational Sciences, RAECS 2014. 10.1109/RAECS.2014.6799608.

[2] Sharma, Shubham & Saxena, Shefali & Khamesra, Ridhima. (2010). SUSTAINABLE IT SERVICES: ENERGY AND POWER AWARE SCHEDULING AND RESOURCE ALLOCATION TO MEET GREEN COMPUTING CHALLENGES.

[3] Koomey, J. Growth in Data Center Electricity Use 2005 to 2010. 2011. Available online: http://www.analyticspress.com/datacenters.html (accessed on 12 June 2016).

[4] Zhang, L. Cheng, And R. Boutaba, "Cloud Computing: State-Of-The-Art and Research Challenges", Journal of Internet Services And Applications, April 2010.

[5] Armbrust, M., Stoica, I., Zaharia, M., Fox, A., Griffith, R., Joseph, A.D. et al. (2010) 'A view of cloud computing', Communications of the ACM, Vol. 53, No. 4, p.50.

[6] R. M. Singh, S. Paul, and A. Kumar, "Task scheduling in cloud computing: Review," International Journal of Computer Science and Information Technologies, vol. 5, no. 6, pp. 7940–7944, 2014.

[7] "Accenture Microsoft report on cloud computing and sustainability: the environmental benefits of moving to the cloud http://www.wspenvironmental.com/media/docs/newsroom,2010

[8]. Guo, M. Liu, J. Wu, Z. Xue and X. He," Energy-Efficient Fault Tolerant Scheduling Algorithm for Real-Time Tasks in Cloud-Based 5G Networks," in IEEE Access, vol. 6, pp. 53671-53683, 2018.

[9] Kliazovich, D., Arzo, S.T., Granelli, F., Bouvry, P. and Khan, S.U. (2013) 'e-STAB: energy-efficient scheduling for cloud computing applications with traffic load balancing', IEEE International Conference on Green Computing and Communications (GreenCom), pp.7–13.

[10] Kolpe, T., Zhai, A. and Sapatnekar, S.S. (2011) ‘Enabling improved power management in multicore processors through clustered DVFS’, 2011 Design, Automation & Test in Europe, Vol. 1, pp. 1–6.

[11] Babin, R., and Nicholson, B. 2011. "How Green Is My Outsourcer? Measuring Sustainability in Global It Outsourcing," *Strategic Outsourcing: An International Journal* (4:1), pp. 47-66

[12] Beloglazov, A., and Buyya, R. 2010a. "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers," *International Conference on Cluster, Cloud and Grid Computing*: IEEE, pp. 577-578

[13] Bener, A.B., Morisio, M., and Miranskyy, A. 2015. "Green Software," *IEEE Software* (31:3), pp. 36-39.

[14] Balnaves, M., and Caputi, P. 2002. *Introduction to Quantitative Research Methods: An Investigative Approach*. Sage

[15] Butler, T. 2012. "Regulating Green, It: Standards, Laws and Protocols," *Harnessing Green IT: Principles and Practices* (298)

[16] Castro, D. 2010. "Learning from the Korean Green It Strategy," Washington, DC: Information Technology and Innovation Foundation), pp. 1-4.

[18] Masanet, E.; Shehabi, A.; Ramakrishnan, L.; Liang, J.; Ma, X.; Walker, B.; Hendrix, V.; Mantha,P. The Energy Efficiency Potential of Cloud-Based Software: A U.S. Case Study. 2014.(accessed on 12 January 2017).

[19] Buyya, Rajkumar, Anton Beloglazov, and Jemal Abawajy. "Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges (2010).

[20] Patili, Archana, and Dr Patili. "An Analysis Report on Green Cloud Computing Current Trends and Future Research Challenges." An Analysis Report on Green Cloud Computing Current Trends and Future Research Challenges (March 19, 2019) (2019).

[21] Masnida Hussin, Young Choon Lee, and Albert Y. Zomaya, "Priority-based scheduling for Large-Scale Distribute Systems with Energy Awareness",Ninth IEEE International Conference on Dependable, Autonomic and Secure Computing, 2011, pp.503- 509

[22] Lu Huang, Hai-shan Chen and Ting-ting Hu, "Survey on Resource Allocation Policy and Job Scheduling Algorithms of Cloud Computing" Journal of Software, Vol. 8, No. 2, February 2013, pp. 480-487.

[23] Tejinder Sharma, and Vijay Kumar Banga, "Efficient and Enhanced Algorithm in Cloud Computing", International Journal of Soft Computing and Engineering (IJSCE), Volume-3, Issue-1, March 2013, pp. 385-390.

[24] Supriya Kinger,Rajesh Kumar,Anju Sharma,”Prediction based proactive thermal virtual machine scheduling in green clouds” The Scientific World Journal,Vol 2014,Article ID 208983,12 pages,2014,doi:10.1155/2014/208983

[25] W.Lin, S.Xu, L.He, and J.Li, Multi-resource scheduling and power simulation for cloud computing,Elsevier journal of Information Sciences, Vol 397398,pp: 168186,August 2017.

[26] Mohammad Reza EffatParvar, Karim Faez, Mehdi EffatParvar, Mehdi Zarei, Saeed Safari. “An Intelligent MLFQ Scheduling Algorithm (IMLFQ) with Fault Tolerant Mechanism”

[27] J. M. Arco D. Meziat B. Alarcos. “Deficit Round Robin Alternated: A New Scheduling Algorithm”

[28] Kishor, Lalit & Goyal, Drdinesh. (2013). Comparative Analysis of Various Scheduling Algorithms. International Journal of Advanced Research in Computer Engineering & Technology (IJARCET). 2. 4.