Shortest-Job First With Fair Priority and Energy Awareness Scheduling In Green Cloud Computing

1Deepika Saxena and 2Dr. R.K. Chauhan,
1Assistant Professor, 2Professor,
1Department of Computer Science, DMM College, Kurukshetra University Kurukshetra, Haryana, India
2Department of Computer Science And Applications, Kurukshetra University Kurukshetra, Haryana, India

Abstract— Cloud computing offers utility-oriented IT services to millions of users simultaneously. Though, it offers an ease in the form of pay-as-you-go model, that enables hosting of variety of massive applications from consumer, scientific, and business domains. However, data centers which provides services to customers and host the Cloud applications, waste a tremendous amount of energy and emit a considerable amount of carbon dioxide. Thus, it is necessary to significantly reduce pollution and substantially lower energy usage. This paper introduces SJF with Fair Priority Green Cloud Computing Scheduling that is environment friendly and allows sustainable development. This scheduling algorithm includes two main steps: assigning maximum possible tasks to a cloud server in heterogeneous environment with lowest energy, and minimizing the total number of active servers and thus saving electrical power consumption and reducing pollution due to carbon emission. The simulation of proposed algorithm is carried out on Cloud Sim Toolkit. The simulation results indicate that proposed algorithm is showing significant improvement as compared to FCFS and simple SJF algorithm. Surely, the algorithm can bring significant benefits to both resource providers and consumers.

Keywords—Cloud Computing ; Cloudsim ; Green Cloud Computing; Resource Allocation ; SJFAlgorithm.

I. INTRODUCTION

Cloud Computing has emerged as a next step towards building this new digital world of Information Technology. The cloud computing has broaden the world of internet to all most everyone in this world. It has converted the complex internet based services to the simplest laymen requirement of computation. The term “Cloud” means a cluster of interconnected distributed datacenters which are expanded worldwide for example, Google, Amazon etc. At each datacenter, there are several computer systems acting as servers and providing enormous range of services to the customers. Briefly describing, cloud computing is a model for enabling convenient, on-demand network access to shared pool of computing resources for example, networks, server, storage, applications and several other services that can be rapidly provisioned and released with minimal management effort or cloud service provider interaction[4]. The basic idea of cloud computing is based on a very fundamental principal of reusability of IT capabilities. One of the fundamental aspects of virtualization technologies employed in Cloud environments is resource consolidation and management. Using hypervisors within a cluster environment allows for a number of standalone physical machines to be consolidated to a virtualized environment, thereby requiring less physical resources than ever before. While this improves the situation, it often is inadequate. Large Cloud deployments require thousands of physical machines and megawatts of power. Therefore, there is a need to create an efficient Cloud computing system that utilizes the strengths of the Cloud while minimizing its energy footprint. In order to correctly and completely unify a Green aspect to the next generation of Distributed Systems, a set of guidelines needs to persist. These guidelines must represent a path of sustainable development that can be integrated into data center construction and management as a whole [1]. Also, this becomes a very challenging issue to schedule several requests simultaneously at datacenter satisfying all the customers, workload balancing, reduction of energy consumption, faster execution of requests at minimum cost etc. Clearly, default FCFS scheduling cannot provide satisfactory results in all the above mentioned respect. Therefore, there is rigorous demand of an optimized real-time scheduling technique. Though, many researchers have proposed different algorithms for scheduling in cloud computing which are providing good solutions in one or other respect, but none of them is able to completely satisfy all the different criteria of scheduling in cloud system. Broadly, in cloud system there is requirement of a scheduling technique which can satisfy all the different terms and conditions of real-time cloud computation. In this paper we are proposing bi-objective scheduling method which provides faster execution of task at minimum cost and also minimizing power consumption as well as decreasing pollution due to carbon and heat emission.

The rest of the paper is organized as follows: section 2 is literature and background survey, section 3 is proposed methodology, section 4 is experiment and result analysis and finally section 5 is conclusion.

II. LITRETURE SURVEY

At Data centers, users submit their requests and the datacenter Broker which is itself an intelligent software to decide scheduling of tasks at available vms (Virtual Machines hosted on physical machines). So, basically all the scheduling framework is designing of datacenter broker. High energy costs and huge carbon emission occurs due to massive amounts of electricity needed to power and cool numerous servers hosted in these data centers. Regarding this we have surveyed different research work as follows. There is an algorithm that covers environment conscious issue for scheduling of HPC applications on distributed cloud centers. The study shows that at present the carbon emission of ICT industry is becoming equal to the carbon emission of aviation industry. Now some of the government imposing a carbon emission limit over the ICT industry. So if cloud providers not consider this issue they are not able to extend their infrastructure in future. For environment conscious the author consider the carbon emission rate of data center. Carbon emission rate of different data center is different so scheduling algorithm schedule the job to a data center which have minimum carbon emission rate. This paper also covers the issue of cost which will maximize the profit of cloud provider. For cost he considers execution price, elasticity price and data transfer price. On base of these information authors suggest algorithm which will find out data center which maximize the profit from the data center pair of
minimum carbon rate. Author’s algorithm is of HPC application where applications have the need of more than one VM. After selection of a data center scheduling algorithm select the VM according to the requirement. Author’s solution is good for environment conscious but there is the limitation of load balancing to choose the VM form a data center which is also a major issue at present time [1]. Another study suggests the factor that influence the job rejection in the cloud environment. He shows the comparison of SJF and R-R scheduling algorithm in peak hour means when no of arrival of job is very high. This paper suggests R-R scheduling for scheduling and SJF for load balancing and process migration to avoid deadlocks. It shows the result that in peak hour SJF has better performance in job rejection issue. SJF has the problem of starvation so long job has very high turnaround time. His study shows that in the case of cloud computing the number of job rejection should be less because cloud is pay-go model so if customer’s job is rejected then the feature of cloud computing would not attract the customer. This paper shows a comparisons between SJF and RR scheduling but he not suggest any way of changing the algorithm from SJF to RR and form RR to SJF author only says about the migration of processes for load balancing but how migration will occur and what is the method of knowing overloaded VM. This paper implements a factor that influence the job rejection in cloud environment and it suggest the RR algorithm for scheduling and SJF scheduling for load balancing but R-R scheduling has a lot of overhead of preemption and SJS has starvation problem [2]. The paper [6] propose a solution for managing large image collections. It presents a cloud computing service and its application for the storage and analysis of very –large image. Its solution allows that an input image can be divided into different sub-images that can be stored and processed separately by different agents in the system, facilitating processing very-large images in a parallel manner. Its purpose is to create a cloud computing service capable of storing and analyzing very- large image datasets. Adequate parallelism and workload balancing of our distributed system is crucial feature to ensure an improved performance. This paper actually presents real life application of very large image datasets and parallelism will really improves the performance such application. Its solution divide large image into sub-images and improve the performance by parallelism. This paper also lack the issue of environment and cost and also of load balancing [6].The paper [8] suggest an algebraic scheduling of the processes because some different processes have different need for the execution so here the algorithm shows the process resource demand in the form of utility function . He suggest that desired resource demand should be in the form soft constraint means it is not necessary that process can execute with the desired resource. As cloud resources and applications grow more heterogeneous, allocating the right resources to different tenants’ activities increasingly depends upon understanding tradeoffs regarding their individual behaviors. One may require a specific amount of RAM, another may benefit from a CPU, and a third may benefit from executing on the same track as a fourth. Here the author modify the existing approach where resource consumers has to specify zero or more hard constraints with each request, based on some predetermined attribute schema understood by the cluster scheduler . Such constraints could serve as a filter on the set of machines, enabling identification of the subset that is suitable for the corresponding request. But, this approach ignores an important issue: in many cases, the desired machine characteristics provide benefit but are not mandatory. Double Level Priority based algorithm [15] which groups the task on the basis of data and requested resources and prioritize the tasks. Resource selection is done on the basis of its cost and turnaround time both using greedy approach. Task selection on the basis of a priority formula, in this way, it attempts to perform better over sequential approach in terms of cost and execution time. All of these existing strategies for scheduling are better than the traditional scheduling strategies in one way or the in other. Some are delivering better time bound results; others are concerned with reducing the cost incurred or with the efficiency and several such factors. Although, work is already done towards minimizing the execution time and cost tasks but there are some flaws as mentioned above. Our proposed algorithm is worth doing in this respect, and is showing significant improvement over existing algorithms.

III. PROPOSED WORK

The aim of this paper is to addresses the problem of enabling energy-efficient resource allocation, hence leading to Green Cloud computing data centers, to satisfy competing application’s demand for computing services and save energy and schedule the request with efficiency in minimum time.

Our algorithm includes following concepts:

1. All the jobs submitted at datacenter are sorted in decreasing order of their length and arranged in the form of sorted list.
2. Prepare three priority queues high, mid and low of sorted tasks and assign weight =3 to high priority queue, weight =2 to mid priority queue and weight = 1 to low priority queue.
3. Accordingly, In round 1, first 3 task will be selected from high queue, then first 2 task will be selected from mid queue and then first one task will be selected from low queue. Until all queues are empty these rounds will be repeated. These weights are assigned to prevent starvation of low priority queue. Thus, we have applied fairness at priority level.
4. Arrange the VMs in ascending order of minimum turnaround time.
5. Assign selected task to the VM with minimum turnaround time and MIPS of VM >=total length of task.
6. Calculate total processing time as follows:

 execution time = Task length / Processing power of selected VM

7. Calculate total cost of execution as follows:

 execution cost = Cost of selected VM * Task length

The idea behind the allocation of selected shortest length task to minimum turnaround time VM is that shortest length task get finish in shortest possible time. Then the VM that has now become idle after executing the current task can take up next selected task from fair priority queue. This will minimize the number of active VMs as well as active servers so that electrical power consumption is less and thus reducing heat and carbon emission as well. Maximum possible heterogeneous VMs can be deployed on minimum number of active physical machines. So that, enormous amount of heat produced due active physical machines get decreased and requirement of air conditioning at datacenter also get reduced. In this way, load balancing and resource allocation can greatly reduce electrical power consumption and emission of carbon and heat radiations.
IV. EXPERIMENTAL RESULT AND ANALYSIS

The CloudSim toolkit is used to demonstrate the simulation. The simulation results are verified using CloudSim (2.1.1) to check the correctness of the proposed algorithm. The simulation results of the proposed algorithm are compared with the Sequential assignment, which is built-in in CloudSim and Shortest Job First Algorithm for Task scheduling.

The configuration of Datacenter created is as shown below, Table I shows the configuration of hosts in this simulation framework.

Here, Number of processing element - 1

Number of hosts – 6

### Table I: Configuration Of Host

<table>
<thead>
<tr>
<th>RAM (MB)</th>
<th>10240</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing Power (MIPS)</td>
<td>110000</td>
</tr>
<tr>
<td>VM Scheduling</td>
<td>Time shared</td>
</tr>
</tbody>
</table>

A. Table II shows the configuration of the system on which these results are obtained as shown below:

### Table II: System Configuration

<table>
<thead>
<tr>
<th>Processor</th>
<th>Pentium® Dual-core CPU <a href="mailto:T4400@2.20GHz">T4400@2.20GHz</a>, 2.20GHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAM</td>
<td>3.00GB</td>
</tr>
<tr>
<td>System types</td>
<td>32bit operating system</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows 7 Ultimate</td>
</tr>
</tbody>
</table>

Table III is showing configuration of VMs for this experiment.

### Table III: Configuration of VMs

<table>
<thead>
<tr>
<th>VM</th>
<th>RAM</th>
<th>Processing Power (MIPS)</th>
<th>Processing Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1</td>
<td>5024</td>
<td>22000</td>
<td>1</td>
</tr>
<tr>
<td>VM2</td>
<td>1048</td>
<td>11000</td>
<td>1</td>
</tr>
<tr>
<td>VM3</td>
<td>3308</td>
<td>22000</td>
<td>1</td>
</tr>
<tr>
<td>VM4</td>
<td>4604</td>
<td>32000</td>
<td>1</td>
</tr>
<tr>
<td>VM5</td>
<td>8028</td>
<td>55000</td>
<td>1</td>
</tr>
<tr>
<td>VM6</td>
<td>4000</td>
<td>41000</td>
<td>1</td>
</tr>
</tbody>
</table>

**Performance with Time:** - The experimental results show the remarkable improvement in time over the sequential approach as well as shortest Job First without fair priority.

### Table IV: Results of proposed algorithm and existing algorithm

<table>
<thead>
<tr>
<th>No. of cloudlets</th>
<th>Sequential algorithm</th>
<th>SJF algorithm without fair priority</th>
<th>Proposed SJF algorithm with Fair Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>98.6154</td>
<td>63.6984</td>
<td>42.3135</td>
</tr>
<tr>
<td>40</td>
<td>407.5499</td>
<td>243.441</td>
<td>178.2001</td>
</tr>
<tr>
<td>60</td>
<td>1241.3430</td>
<td>549.131</td>
<td>436.3890</td>
</tr>
<tr>
<td>80</td>
<td>1953.1897</td>
<td>785.0639</td>
<td>676.0989</td>
</tr>
<tr>
<td>100</td>
<td>2103.9777</td>
<td>1476.6538</td>
<td>1066.1269</td>
</tr>
</tbody>
</table>

Comparing proposed algorithm with sequential i.e. FCFS scheduling algorithm and SJF algorithm shows the tremendous improvement in results. As the number of cloudlets are increasing definitely, the total execution time has decreased together for both deadline and cost based tasks. The result shown above are the standard mean of total execution time obtained after several number of execution for each number of cloudlet (e.g. we have run the implementation 15 times for 100 cloudlets and calculated its mean).

### Performance with Cost:

The experimental results show the enhanced performance of proposed algorithm over existing algorithms.

### Table V: Results of proposed algorithm and existing algorithm

<table>
<thead>
<tr>
<th>No. of cloudlets</th>
<th>Sequential algorithm</th>
<th>SJF algorithm without fair priority</th>
<th>Proposed SJF algorithm with Fair Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>86.0</td>
<td>79.377</td>
<td>69.400</td>
</tr>
<tr>
<td>40</td>
<td>179.0</td>
<td>135.111</td>
<td>121.001</td>
</tr>
<tr>
<td>60</td>
<td>383.0</td>
<td>270.500</td>
<td>184.690</td>
</tr>
<tr>
<td>80</td>
<td>486.0</td>
<td>267.036</td>
<td>221.904</td>
</tr>
<tr>
<td>100</td>
<td>545</td>
<td>359.837</td>
<td>300.845</td>
</tr>
</tbody>
</table>

As the number of cloudlets are increasing the total execution cost has decreased together for both deadline and cost based tasks. The results shown above are the standard mean of total execution time obtained after several number of execution for each number of cloudlet (e.g. we have run the implementation 15 times for 100 cloudlets and calculated its mean).

**Figure. 1** Traditional V/S Proposed algorithm with respect to time

The above bar graph showing task completion time and comparison of traditional algorithm and proposed algorithm.

**Figure. 2** Cost comparison Traditional V/S Proposed algorithm with respect to cost
CONCLUSION

This work advances Cloud computing Shortest Job First Scheduling by adding Fair Priority and green computing features. Also, it plays a significant role in the reduction of computation time and cost as well as reducing data center energy consumption costs and thus helps to develop a strong, competitive Green and eco rich cloud computing industry. To meet thousand of service requests while making best possible use of available resources and simultaneously satisfying both user as well as service provider, is the challenge for task scheduler. Traditional methods of scheduling lead to overpricing and slow processing for bulk of tasks. Some task scheduling algorithm is cost based, some are deadline based and many algorithms make use of priority based scheduling.

But they suffer from long waiting priority queues. Our proposed algorithm definitely meet all the challenges, along with constraint based optimization scheduling. We have also, introduced fair-priority scheduling concept i.e. combination of priority with Shortest job First scheduling scheme. Also, consumers are increasingly becoming conscious about the environment.

In future, more pragmatic algorithm can be devised using several other combination of scheduling schemes. Grouping of tasks can be done based on complexity of task, or group by location etc., before resource allocation. Researchers from worldwide can take it as a challenge to enhance energy-efficient scheduling and management of Cloud computing environments.

References


