Energy Efficient Dynamic Tasks Oriented Scheduling in Green Control Cloud

Abstract
Cloud computing is the energetic distribution of information technology resources and proficiencies as a service above the Internet. Virtualization is the establishment of a virtual arrangement of system such as system computing device or machine server, a computer operating system, or network system device or hard disk storage device. To address the main problem of high load consumed in cloud system the efficient algorithm should be present and used. To make available scalability and eradicate the points of let-down, it is compulsory to prepare self-motivated Virtual Machine (VM) consolidation algorithms in a distributed manner. Therefore, a unique procedure is compulsory to create the dynamic VM consolidation distributed system. The scheduling objectives are to improve the system’s time conflicts and time utilization for real-time tasks and save energy. An effective green control algorithm is developed which lead to significant task scheduling and reduced energy consumption. The experimental performed using CloudSim simulator indicates the scalability and performance of system in cloud computing. Our proposed technique can also improve the system performance of cloud computing in actual time cloud based environment.

Keywords: Cloud Computing, Virtualization, Energy Efficiency, Round Robin, Load Balancing

Introduction
Cloud computing [1] is a protocol for allowing widespread, suitable, on-demand network system access. Cloud based computing is a process of computing wherever immensely scalable IT services enabled competences are distributed ‘as a service’ using Internet based technologies to multiple outdoor clients. At its simplest form, cloud based computing is the run time transport of information technology devices or resources and competencies as a facility above the The cloud computing resources are storage, networks, applications, servers, and services. Virtualization [2] has cultivated a real-world prerequisite for this generation. Its predisposition is unending, for the reason that when functional, it transports many proceeds such diminution in functioning capital and costs, Hard-to-find person’s resource savings, Physical space decrease, Access to storage re-sources, computer network, server and on demand, Energy savings for a greener environment. Virtualization is the establishment of a virtual arrangement of such as computing system device or system server, a computer operating system, or network system devices and storage system device. Traditionally, the development of cloud computing architecture focuses on performance of the system. The unbalanced load will increase the energy consumption of the cloud system. An efficient implementation of cloud architecture, improves the utilization of cloud resources with dynamic and static data. To fulfil undefined workloads and to be exceedingly accessible for consumers anyplace at some intervals, system or resource over provisioning is a shared condition in a cloud based system. The aim of computing is shifted to energy consumption and task balancing. The foremost element is servers are only full of activity 12-28 percent of the time on average. If at all possible, shutting downcast servers as soon as they are absent idle throughout small load epochs is one of the furthestmost shortest ways to diminish power consumption. To address the main problem of high load consumed in cloud system the efficient algorithm [4] should be present and used. The ultimate goal of the load balancing algorithm is to improve the resource utilization and compact energy consumption so that the overall performance [5] of the cloud architecture increases. An effective green control procedure is established which lead to significant task scheduling and reduced energy consumption. An efficient implementation of cloud architecture improves the utilization of cloud resources with dynamic [6] and static data or info. The reaction interval of the cloud resources increases which improves the overall performance [7] and cost of the cloud architecture. The proposed algorithm is developed to perform the following objectives

1. To decrease the energy utilization of the cloud resources
2. To develop efficient task management algorithm with static and dynamic load
3. Reduction the cost and upsurge the performance of the cloud services

Paper is organized as follows. Section II represents work related to green computing, and energy saving algorithm. Section III provides proposed work and algorithm in detail. Section IV represents the implementation work and result analysis of proposed
algorithm. Section V concludes the paper with future scope.

Related Work
Green based computing and energy preservation in present disseminated computing situation are getting countless deal of consideration in the research communal and efficient scheduling methods [8]. According to the survey the energy consumption increases year wise due to the computing resources. The ultimate goal of the load balancing algorithm is to improve the resource utilization and compact energy consumption so that the overall performance of the cloud architecture increases. On the other hand, it decreases the energy utilization of the cloud resources. The encounters of regulatory facility rates and smearing the N-policy to elevate operative cost within a concert assurance are first considered. A cost purpose has been established in which the prices of power ingestion, method cramming and server start-up are all engaged into contemplation. The consequence of energy effectiveness is control over reaction intervals, functioning methods and suffered costs all these are validated. [9] Proposes is to discover the optimum facility degree and approach substituting constraint, so as to minimize price within a reaction interval assurance under variable entrance rates. An effective green control (EGC) procedure is first suggested for resolving constrained optimization difficulties and construction costs and performances compromises in schemes with dissimilar power redeemable strategies. Simulation outcomes display that the reimbursements of dropping functioning prices and enlightening reaction times can be certified by smearing the power saving strategies joint with the suggested procedure as likened to an archetypal method beneath a same enactment assurance [16] Deliberated the green computing effective administration matter of similar resources in network or Internet accommodating centers. The energy consumption is represented in kilowatt. The suggested methodology diminishes the energy ingestion and provides green computing by exchanging idle system servers to power redeemable manners and is appropriate for power effective resource distribution at the info or data center level. It develops efficient task management algorithm with static and dynamic load. [11] Suggested a good performance and energy cognizant setting up procedure in collected environment for calculating concentrated works with unidentified facility period. To diminish the energy ingestion of data center [12] the task management system should be proper. The unbalanced system may generate carbon dioxide and increased the electricity bill of the organization. [14] Considered disseminated performance focused on DV-FS arranging approaches that can make noteworthy energy savings without cumulative implementation of time by changing scheduling granularity. In order to reduce energy consumption of resources, the system should reduce the energy wastage of computing resources. An efficient green control algorithm should be developed which lead to significant task scheduling and reduced energy consumption. It reduces the cost and upsurge the performance of the cloud services.[15] suggested two data power cognizant development algorithms (space-shared and time-shared) for request of tasks real time applications on DVS permitted groups to diminish energy indulgence while assembly applications’ limits. The energy consumption of resources increases year by year. [16] Considered the energy redeemable problem for intermittent inhibited limit real-time jobs on an immovable quantity of processors. The suggested scheduling procedure is preventive; each procedure can jump to implement on some processors and might wander at runtime, if it becomes anticipated by former limit processes. To address the main problem of high load consumed in cloud system the efficient algorithm should be present and used.

Proposed Method
An efficient green control algorithm should be developed which lead to significant task scheduling and reduced energy consumption. In proposed system we provide efficient task management algorithm with static and dynamic load. Due to job management the energy utilization of the cloud resources will decrease. It will also decrease the cost of the organization. The response time of the cloud resources should be increased which progresses the complete cost and performance of the cloud architecture. To address the main problem of high energy consumed in cloud system the efficient algorithm should be applied. The key purpose of proposed work is to develop efficient task management algorithm with static and dynamic load, to shrink the energy consumption of the cloud resources, to shrink the cost and escalation of the performance of the cloud services. An effective green control procedure is developed which lead to significant task scheduling and reduces energy consumption. An efficient implementation of cloud architecture progresses the consumption of cloud based resources with dynamic and static data. The response time of the cloud resources increases which increases the complete cost and performance of the cloud architecture.

Proposed Algorithm
Set all the virtual machines and data centers as available
Initialize list without entry
Repeat until new task request from client or user Do
Middle were server make queue for the requests Middle
were server eliminates a task demand from the starting
of the queue
If (data list structure comprise in the least entry of a data
center server corresponding to the present demanding
system/user AND server distribution status = is
available) then The data base server machine is altered to
the demanding system/ user Else
Distribute a data center system server to the demanding system/user using round_robin procedure or method. Assign data center system or CPU to each and every process in round robin method, according to prearranged period significant only for unique time. Afterwards, the accomplishment of step-1 tasks/processes are organized in cumulative order or their left over burst interval in the ready process queue. Fresh priorities are allotted rendering to the left over burst time of tasks/processes; the tasks/process with least remaining burst period is allocated with maximum priority. The tasks or processes are accomplished according to the new significance based on the left over bursts interval. Altered the record of the system and the data base system server in the record data list structure and the state collections.

End if

While (new task request are expected by the data center server) End loop

**Step 1:** The collection list of obtainable servers is kept in available data center server list. Set all the virtual machines and data centers as available Initialize list without entry. Available is linked list having collection of all make ready data center system server. Configure the virtual machine and data center for proper tasks distribution.

**Step 2:** Prepare hash map data structure without entry. The task requests Data center server eliminates a task demand from the starting of the main queue.

**Step 3:** A user/client demand to the datacenters systems for essential information. It collects the process/task demand from the user/client, checks the respective load from dissimilar datacenters.

**Step 4:** Linked list data structure comprise any new entry of a data center system server corresponding to the current demanding task/user and system/server distribution position is accessible then the data center system/server is altered to the demanding user. It also makes sure accessibility position of the datacenter system/server.

**Step 5:** Distribute a data center system server to the demanding system/user using round robin method.

**Step 6:** Assign data center system to each and every process in round robin method, according to prearranged period significant only for unique time. Fresh priorities are allotted rendering to the left over burst time of tasks; the tasks with least remaining burst period is allocated with maximum priority. The tasks are accomplished according to the new significances based on the left over bursts interval.

**Step 7:** Altered the record of the system and the data base system server in the record data list structure and the state collections. If system is accessible, then the demand is transported to the system and process will be accomplished.

**Step 8:** End

**Implementation and Result Analysis**

CloudSim and CloudAnalyst tools are used for simulation of proposed work. Java platform is used for implementation of our proposed algorithm. For simulation environment we used i5 processor with 3.0GHz processor, 8 GB RAM, 7200 RPM, 1 TB hard disk. CloudAnalyst and CloudSim are Java based widely used simulators for cloud computing implementation.

We have tested simulations on 2, 5, 10 and 20 virtual machines. The data center used for implementation is 5, 8 and 10 with different region selected. We have tested our simulation with different static and dynamic load with different data packet size and time duration.

**Data Center Detail:** The table below represents the data center detail.

<table>
<thead>
<tr>
<th>Name</th>
<th>DC1</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMM</td>
<td>Xen</td>
</tr>
<tr>
<td>Region</td>
<td>0</td>
</tr>
<tr>
<td>Cost per VM $/Hr</td>
<td>0.1</td>
</tr>
<tr>
<td>OS</td>
<td>Linux</td>
</tr>
<tr>
<td>Memory cost $/Hr</td>
<td>0.05</td>
</tr>
<tr>
<td>Physical Hardware unit</td>
<td>2</td>
</tr>
<tr>
<td>Storage Cost $/S</td>
<td>0.1</td>
</tr>
<tr>
<td>Architecture</td>
<td>X86</td>
</tr>
<tr>
<td>Data Transfer Cost $/Gb</td>
<td>0.1</td>
</tr>
</tbody>
</table>

**Table 1: Data Center Detail**

The data center installed on Linux machine with x86 64bit processors. The Xen virtual machine is used for implementation of virtual machine. Number of processors used in virtual machines are 4. The speed of the processor is 10000 ms each. The policy used in implementation is time shared. And the algorithm used in implementation is improved round robin algorithm. The storage of the data center is 10,00,00,000 Mb and the size of main memory is 20,480 Mb. Physical hardware unit used in data center is 2.

**Figure 1: Simulation Result**

The figure above shows the simulation result using CloudAnalyst. The test scenario is represented in table below.
Data center | DC1 | DC2
---|---|---
VM | 5 | 10
Data size | 10000 | 15000
Memory bytes | 512 | 1024
Bandwidth | 1000 | 1000

Table 2: Test Scenario 1 for Data Center

<table>
<thead>
<tr>
<th>Task</th>
<th>Task 1</th>
<th>Task 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Region</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Req/user/hr</td>
<td>60</td>
<td>120</td>
</tr>
<tr>
<td>Data size/req bytes</td>
<td>100</td>
<td>150</td>
</tr>
<tr>
<td>Avg peak users</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

Table 3: Task Detail

Table 4: Response Time

<table>
<thead>
<tr>
<th>Task</th>
<th>Avg (ms)</th>
<th>Avg (ms)</th>
<th>Avg (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>300.141</td>
<td>245.381</td>
<td>359.212</td>
</tr>
<tr>
<td>Task 2</td>
<td>300.963</td>
<td>239.128</td>
<td>374.11</td>
</tr>
<tr>
<td>DC1</td>
<td>0.332</td>
<td>0.01</td>
<td>0.516</td>
</tr>
<tr>
<td>DC2</td>
<td>0.442</td>
<td>0.02</td>
<td>0.616</td>
</tr>
</tbody>
</table>

Overall response time = 300.8 ms, Data transfer cost = 0.507 + 0.129 = 0.636

Table 5: Overall Response Time

<table>
<thead>
<tr>
<th>Avg (ms)</th>
<th>Avg (ms)</th>
<th>Avg (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall response time</td>
<td>300.8</td>
<td>240.1</td>
</tr>
<tr>
<td>Data center processing time</td>
<td>0.34</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Table 6: Data Center Servicing Times

Table above represents the data center service response time of the proposed work.

Conclusion

The difficulty comprises in defining the best subset of VMs to migrate, that will provide the most beneficial system reconfiguration. The ultimate goal of the load balancing algorithm is to improve the resource utilization and reduced energy consumption so that the overall performance of the cloud architecture increases. To address the issue of scheduling problem in real time system we have introduced an improved scheduling algorithm which can also be scalable in cloud computing. Our proposed technique can also improve the system performance of cloud computing in real time cloud environment. An effective green control algorithm is developed which lead to significant task scheduling and reduced energy consumption. The experimental performance using CloudSim, indicated the scalability and performance of system in cloud computing. In future work, we will also apply security features in our algorithm in real big data difficulties. On the same grounds, we are preparing to implement our algorithm in real time task oriented problems.

References