A NOVEL APPROACH OF OPTIMIZING PERFORMANCE USING K-MEANS CLUSTERING IN CLOUD COMPUTING

Sheenam Kamboj (1), Mr. Navtej Singh Ghumman (2)

(1) Research Scholar, Department of Computer Science & Engineering, SBSSTC, Ferozepur, Punjab.
sheenam31.sk@gmail.com

(2) Assistant Professor, Department of Computer Science & Engineering, SBSSTC, Ferozepur, Punjab.
navtejghumman@yahoo.com

ABSTRACT

Cloud computing is distributed computing, storing, sharing and accessing data over the Internet. It provides a pool of shared resources to the users available on the basis of pay as you go service that means users pay only for those services which are used by him according to their access times. Load balancing ensures that no single node will be overloaded and used to distribute workload among multiple nodes. It helps to improve system performance and proper utilization of resources. We propose an improved load balancing algorithm for job scheduling in the cloud environment using K-Means clustering of cloudlets and virtual machines in the cloud environment. All the cloudlets given by the user are divided into 3 clusters depending upon client’s priority, cost and instruction length of the cloudlet. The virtual machines inside the datacenter hosts are also grouped into multiple clusters depending upon virtual machine capacity in terms of processor, memory, and bandwidth. Sorting is applied at both the ends to reduce the latency. Multiple number of experiments have been conducted by taking different configurations of cloudlets and virtual machine. Various parameters like waiting time, execution time, turnaround time and the usage cost have been computed inside the cloudsim environment to demonstrate the results. Compared with the other job scheduling algorithms, the improved load balancing algorithm can outperform them according to the experimental results.

Keywords


INTRODUCTION

Cloud computing is a combination of many computing fields and has gained much popularity in the recent years. Cloud computing provides computing, storage, services, and applications over the Internet. Moreover, cloud computing facilitates to reduce capital cost, decouple services from the underlying technology, and provides flexibility in terms of resource provisioning. Cloud computing has become very beneficial for business services, applications and other types of consumer requirements. Very large enterprises are practicing to scale back all their hardware and infrastructure on the cloud and for that reason most of them has already begin consolidating their IT operations and virtualization mechanisms and technologies on the cloud[1].

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing has claimed to jump the enterprise business to a brand-new level and permits them to cut back all the prices through improved production, reduced administration and infrastructure, architecture price and quicker preparation cycles.

Cloud computing is a type of computing that relies on sharing computing resources rather than having local servers or personal devices to handle applications. In cloud computing, the word cloud(also phrased as “the cloud”) is used as a metaphor for “the Internet”, so the phrase cloud computing means "a type of Internet-based computing", where different services such as servers, storage and applications are delivered to an organization's computers and devices through the Internet. Cloud computing is comparable to grid computing, a type of computing where unused processing cycles of all computers in a network are harnessed to solve problems too intensive for any stand-alone machine. The Cloud Computing may be a term that describes the infrastructure, platform, services and different kind of applications. As this is a platform it reconfigures servers or applications where the server can be a physical machines or virtual display machines. Cloud computing is different from ancient computing paradigms because it is customizable, scalable, encapsulated, abstract entity that gives totally different level of services, processes to the clients, driven by economies of scale and also the services area unit dynamically and totally configurable [2].

RELATED WORK

The most of the researches have been working in the area of load balancing in cloud computing process for enhancing the overall performances of the clouds. Some of these tasks should contain the improved traditional mechanisms to achieve the objective of load balancing. So as to appreciate their contribution, determination and better understandability the work ahead.

Al-Rayis et al. [1] explains that basically, load balancers can be deployed based on three different architectures. The centralized load balancing architecture which includes a central load balancer to make the decision for the entire system regarding which cloud resource should take what workload and based on which algorithm(s).

Bhoi et al. [2] discussed that in enhanced Max-Min Task Scheduling Algorithm in cloud computing helps in supplying a high performance computing based on protocols which allowed shared computation and storage over long distances. It
depends upon expected execution time instead of completion time. Max-Min algorithm assign task with maximum execution time to resource produces minimum completion time while Enhanced Max-min assign task with average execution time to resource produces minimum execution time.

Bhadani et al. [3] proposed a Central Load Balancing Policy for Virtual Machines (CLBVM) that balances the load evenly in a distributed virtual machine/cloud computing environment.

Bendia et al. [4] introduced the Map Reduced based Entity Resolution load balancing technique in networking which is based on large datasets. In this technique, two main tasks are done: Map task and Reduce task which the author has described.


Buzato et al. [6] proposed Bee Life algorithm which was used for scheduling in Cloud computing. Bee Life algorithm is inspired by the behavior and reproduction of bee to find food source. The algorithm evaluated the performance of the resources and it has the aim to reduce time and complexity of work.

Babu et al. [7] proposed a Honey Bee Behavior inspired Load Balancing [HBB-LB] technique which helps to achieve even load balancing across virtual machine to maximize throughput. It considers the priority of task waiting in queue for execution in virtual machines.

Dorigo et al. [8] has proposed a load balancing technique called colony of cooperating agents in ants based on soft computing for solving the optimization problem. This technique solves the problem with high probability. It is a simple loop moving in direction of increasing value which is uphill. And this make minor change in to original assignment according to some criteria.

Deldari et al. [9] proposed a novel load balancing algorithm called VectorDot in intelligent ants. It handles the hierarchical complexity of the datacenter and multidimensionality of resource loads across servers, network switches, and storage in an agile data center that has integrated server and storage virtualization technologies.

Desai et al. [10] discusses about the emerging technology i.e. a new standard of large scale distributed computing and parallel computing. It provides shared resources, information or other resources as per clients’ requirements at specific times. For better management of available good load balancing techniques are required. And through better load balancing in cloud, performance is increased and user gets better services. So in this author has discussed many different load balancing techniques used to solve the issue in cloud computing environment.

Elzeki et al. [11] discussed in Improved Max-Min Algorithm in Cloud Computing that focuses on the cloud computing which further deals with the allocation of the tasks to the resources while observing different parameters like waiting time, Average waiting time, Turn Around time, processing cost. So, an algorithm named as Max-Min in improved manner from load balancing has been shown to overcome such kinds of problems.

Fahringer et al. [12] introduced a static load balancing technique called Ant Colony Optimization. In this technique, an ant starts the movement as the request is initiated. This technique uses the Ants behavior to collect information of cloud node to assign task to the particular node. In this technique, once the request is initiated, the ant and the pheromone starts the forward movement in the pathway from the “head” node.

Fang et al. [13] discussed a two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

OBJECTIVES
The primary objectives of this research work are summarized as follows:

- To study the performance of existing load balancing algorithm.
- To propose a new efficient load balancing with clustering at both sides i.e client side and server side.
- To apply the sorting mechanism on the clusters formed at the client as well as cloud side.
- To reduce the overhead time of scanning the entire VM’s in a cluster by arranging them in descending order.
- To implement the concept of priority based execution depending upon client’s cost.
- To implement the proposed algorithm in cloudsim simulator.
- To evaluate the performance of proposed algorithm with current algorithm.

The Cloud network consist multiple users input with their different requirements which needed to fulfill by efficiently utilizing the available resources. There are different ways to fulfill user's requirement (Like priority). The main objective of this research work is to answer the question: in identical cloud environments, which load balancing architecture: centralized, decentralized or hierarchical architecture will give the best results in terms of response time and server load
To answer this question a robust evaluation framework is implemented which includes the following steps:

- To balance the load equally among different VMs.
- Fetch all the available virtual machines in the datacenter/host.
- Retrieve the processing capacity of the available virtual machines.
- Clustering at the cloudlet side is done on the basis of the user requirements in terms of cloudlet length and cost.
High, medium, low priority is assigned to tasks and priority is directly proportional to cost i.e. high is the priority, more cost will be charged.

Cluster to cluster assignment of cloudlets is done which reduces the time as compared when cloudlets were assigned one by one.

Descending order is applied at both sides within clusters for maximizing the benefit.

The task is allocated with the help of load balancing algorithm.

In this strategy, current system state plays major role while making decisions. Despite the fact that dynamic load balancing has higher run time complexity then static one, dynamic has better performance report as it considers current load of system for choosing next datacenter to serve the request. This will surely provide an optimal choice from available ones for that state of system. Workload in the cloud is regularly a multi-objective problem. In this thesis we will highlight and pay attention to some of these problem and possible solution, so as to obtain an optimal solution. We expect that every application comprises of a number of slightly parallel tasks. Every application has a strict fulfillment due time. Prior to this due time, all computational assignments in the application must be completely executed with the results conveyed to the client. Our current application model concentrates on random sort of workloads. With two different clients group one with higher resources accessing rights while other group has relatively lower resources accessing rights.

RESEARCH METHODOLOGY

Cloud services provide computing on demand in real time. Number of users accessing cloud environment are always more than that were using it on previous day. Cloud has application areas for developing applications, providing and managing infrastructure, patching applications. Users and their requests for accessing cloud infrastructure are highly dynamic and loading servers running in data center. We need efficient strategy to balance load on these servers so that the servers don’t get crash and they can persist long. Precisely Objective is to achieve accuracy, performance of servers and the cloud environment can be maintained.

Steps:
1. Initialize the Cloud Sim in Java
2. Create the Datacenter with different number of hosts.
3. Each Host will have the different numbers of Virtual machines of different capacities.
4. Then we will create the Cloudlets of varying length and size.
5. The list containing the Virtual machines [18] and Cloudlets will be given to the Data Center Broker (DCB)
6. DCB will compute the processing capacity of all the virtual machines and will divide them into multiple clusters using K-Means clustering by using various parameters like bandwidth, memory and processing capability.
7. Data center broker will maintain the list of the cloudlets and will also divide them into multiple clusters using K-Means Clustering. Various parameters used over here are cloudlet length, priority of the cloudlet and the cost associated with it.
8. All the cloudlets and the virtual machines inside the clusters are sorted in descending order.
9. Dispatch our cloudlet to appropriate virtual machine in the cluster. Since all the cloudlets and virtual machines are sorted in descending order, so the cloudlet with higher instruction size and higher priority will be assigned to the virtual machine with higher resources.
10. Repeat the same procedure for all the remaining cloudlets in the list.

Figure 1. Proposed Load Balancing Model
ALGORITHM

Algorithm of the proposed work is written as follows:
Input:- Unallocated Tasks/Cloudlets/Virtual Machines.
Output:- Response Time, Waiting Time, Processing Cost.
Algorithm:
1. Input the Cloudlets(CL) to the CloudSim.
2. foreach Cloudlet k in CL
   find the Instruction length, priority/deadline and cost of k.
end
3. Start the K-Means Clustering and divide them into 3 clusters.
4. Assume Centroid A, B, C
5. foreach Cloudlet k in CL
   Calculate Euclidean Distance of k with A, B and C.
   if previous distance = new distance
   { Stop Iterations
   }
   else
   { Add Ck into minimum clustered distance
     Again Compute the Centroids
   }
end for
6. Create Virtual machines (VMs) in the CloudSim.
7. foreach VM m in VMs
   find the capacity, memory and bandwidth of each m.
end
8. Start the K-Means Clustering and divide virtual machines into 3 clusters.
9. Assume Centroid I, J, K
10. foreach VM k in VMs
    Calculate Euclidean Distance of k with I, J and K
    if previous distance = new distance
    { Stop Iterations
    }
    else
    { Add Vk into minimum clustered distance
      Again Compute the Centroids
    }
end for
11. Apply descending sort on VMs and Cloudlets in Clusters.
12. foreach Cloudlet j
    assign Cloudlet j to VMj of appropriate cluster
    increment the VM.
    if VMmax >= ListSize
    { VMindex = 0;
    }
end for
Cloud Sim [4] could be an illustrious machine for cloud parameters developed within the CLOUDS Laboratory, at the pc Science and computer code Engineering Department of the University of Melbourne. The Cloud Sim library is employed for the subsequent operations: massive scale cloud computing at information centers. Cloud Computing includes the virtualized server machines/hosts with all types of modifiable policies. It has the support for simulation and modeling of very large scale knowledge/information centers.

We need efficient strategy to balance load on these servers so that the servers don’t get crash and they can persist long. Precisely Objective is to achieve accuracy, performance of servers and the cloud environment can be maintained. Various experiments have been conducted and the results of existing work and the proposed work have been mentioned in the Table 1 and Table 2

EXPERIMENTAL RESULTS

<table>
<thead>
<tr>
<th>Sno.</th>
<th>No of Cloudlets</th>
<th>Total Turn Around Time</th>
<th>Total Execution Time</th>
<th>Total Waiting Time</th>
<th>Average Waiting Time</th>
<th>Total Processing Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>10</td>
<td>9</td>
<td>1</td>
<td>0</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>19</td>
<td>18</td>
<td>1</td>
<td>0</td>
<td>56</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>187</td>
<td>73</td>
<td>114</td>
<td>3</td>
<td>223</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>390</td>
<td>110</td>
<td>280</td>
<td>5</td>
<td>335</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>1015</td>
<td>183</td>
<td>832</td>
<td>8</td>
<td>558</td>
</tr>
<tr>
<td>6</td>
<td>150</td>
<td>2207</td>
<td>274</td>
<td>1933</td>
<td>13</td>
<td>837</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>3857</td>
<td>366</td>
<td>3491</td>
<td>17</td>
<td>1115</td>
</tr>
<tr>
<td>8</td>
<td>300</td>
<td>8525</td>
<td>548</td>
<td>7977</td>
<td>27</td>
<td>1673</td>
</tr>
<tr>
<td>9</td>
<td>400</td>
<td>15020</td>
<td>731</td>
<td>14289</td>
<td>36</td>
<td>2230</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>23347</td>
<td>914</td>
<td>22433</td>
<td>45</td>
<td>2788</td>
</tr>
<tr>
<td>11</td>
<td>700</td>
<td>45473</td>
<td>1279</td>
<td>44194</td>
<td>63</td>
<td>3903</td>
</tr>
</tbody>
</table>
In the table 2, we have mentioned the results of the various experiments conducted on the proposed algorithm by taking the same configuration of virtual machines and the same properties of the cloudlets.

<table>
<thead>
<tr>
<th>Sno.</th>
<th>No of Cloudlets</th>
<th>Total Turn Around Time</th>
<th>Total Execution Time</th>
<th>Total Waiting Time</th>
<th>Average Waiting Time</th>
<th>Total Processing Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>15</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
<td>15</td>
<td>10</td>
<td>5</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>3</td>
<td>40</td>
<td>161</td>
<td>41</td>
<td>120</td>
<td>3</td>
<td>126</td>
</tr>
<tr>
<td>4</td>
<td>60</td>
<td>351</td>
<td>63</td>
<td>288</td>
<td>5</td>
<td>191</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>923</td>
<td>104</td>
<td>820</td>
<td>8</td>
<td>317</td>
</tr>
<tr>
<td>6</td>
<td>150</td>
<td>2052</td>
<td>157</td>
<td>1895</td>
<td>13</td>
<td>478</td>
</tr>
<tr>
<td>7</td>
<td>200</td>
<td>3590</td>
<td>208</td>
<td>3381</td>
<td>17</td>
<td>636</td>
</tr>
<tr>
<td>8</td>
<td>300</td>
<td>8020</td>
<td>313</td>
<td>7706</td>
<td>26</td>
<td>956</td>
</tr>
<tr>
<td>9</td>
<td>400</td>
<td>14136</td>
<td>417</td>
<td>13719</td>
<td>34</td>
<td>1273</td>
</tr>
<tr>
<td>10</td>
<td>500</td>
<td>22029</td>
<td>522</td>
<td>21507</td>
<td>43</td>
<td>1592</td>
</tr>
<tr>
<td>11</td>
<td>700</td>
<td>43015</td>
<td>731</td>
<td>42285</td>
<td>60</td>
<td>2229</td>
</tr>
<tr>
<td>12</td>
<td>1000</td>
<td>87560</td>
<td>1044</td>
<td>86516</td>
<td>87</td>
<td>3185</td>
</tr>
<tr>
<td>13</td>
<td>2000</td>
<td>349217</td>
<td>2088</td>
<td>347129</td>
<td>174</td>
<td>6371</td>
</tr>
<tr>
<td>14</td>
<td>3000</td>
<td>785170</td>
<td>3133</td>
<td>782037</td>
<td>261</td>
<td>9559</td>
</tr>
<tr>
<td>15</td>
<td>5000</td>
<td>2178548</td>
<td>5222</td>
<td>2173326</td>
<td>435</td>
<td>15931</td>
</tr>
<tr>
<td>16</td>
<td>7000</td>
<td>4268335</td>
<td>7310</td>
<td>4261025</td>
<td>609</td>
<td>22304</td>
</tr>
<tr>
<td>17</td>
<td>8000</td>
<td>5574486</td>
<td>8355</td>
<td>5566131</td>
<td>696</td>
<td>25490</td>
</tr>
<tr>
<td>18</td>
<td>9000</td>
<td>7055333</td>
<td>9400</td>
<td>7045933</td>
<td>783</td>
<td>28678</td>
</tr>
<tr>
<td>19</td>
<td>10000</td>
<td>8708634</td>
<td>10443</td>
<td>8698190</td>
<td>870</td>
<td>31863</td>
</tr>
<tr>
<td>20</td>
<td>20000</td>
<td>34824312</td>
<td>20888</td>
<td>34803424</td>
<td>1740</td>
<td>63728</td>
</tr>
<tr>
<td>21</td>
<td>30000</td>
<td>78349034</td>
<td>31332</td>
<td>78317702</td>
<td>2611</td>
<td>95594</td>
</tr>
<tr>
<td>22</td>
<td>40000</td>
<td>139275024</td>
<td>41776</td>
<td>139233249</td>
<td>3481</td>
<td>127458</td>
</tr>
<tr>
<td>23</td>
<td>50000</td>
<td>217611392</td>
<td>52220</td>
<td>217559173</td>
<td>4351</td>
<td>159323</td>
</tr>
<tr>
<td>24</td>
<td>60000</td>
<td>313358803</td>
<td>62664</td>
<td>313296139</td>
<td>5222</td>
<td>191189</td>
</tr>
<tr>
<td>25</td>
<td>70000</td>
<td>426502150</td>
<td>73108</td>
<td>426429043</td>
<td>6092</td>
<td>223052</td>
</tr>
</tbody>
</table>
Figure 5.1 illustrates the turn around time for the existing work and proposed work. It is clear from the line chart, that the turn around times of the cloudlets have been reduced, thereby increasing the overall efficiency of the system.

![Turn Around Time](image)

Response Time

- It is defined as total time taken by a load balancing algorithm to finish the execution of a cloudlet.
- Response Time (RT) = FT – ST
  
  where,
  
  FT = finish time of execution
  ST = start time of execution

![Execution Time of Proposed Algorithm](image)

Processing Cost

- It is obtained by addition of cost per storage, cost per memory and cost per memory.
  
  Processing Cost = RT * unit_cost.
  
  where,
  
  RT = response time
  unit_cost = cost per unit time

![Processing Cost](image)
Figure 5.4 illustrates the waiting time for the base work and proposed work. It is clear from the line chart, that the waiting times of the cloudlets have been reduced, thereby decreasing the cost and increasing the overall efficiency of the system.

CONCLUSION AND FUTURE SCOPE

The performance of improved load equalization algorithmic program has been studied in this research work. The request time for the policies applied are same which suggests there's no impact on data centers request time after modifying the algorithm. The processing cost, turnaround time, waiting time, execution time are calculated using various number of experiments. The experiments conducted are compared with previous algorithms. The results indicate that the approaches surpass to previous algorithmic program in terms of execution time, waiting time and turnaround time. The experimental results are obtained by applying the new planned algorithmic program within the Cloud Sim simulator developed in java programming language, shows that the new work has outperformed the present programming algorithms in giant scale distributed systems. To get a much better answer and more precise results, the model ought to be created a lot of realistic by considering problems regarding load equalization like information section, communication price and flow time and results may be tested in real cloud setting. Moreover, fault tolerance, virtual machine migration and the power consumed by the virtual machine are also the considerable factors that can be explored in the future work.

REFERENCES


