Load Balancing in Cloud Computing Using K-Medoids Clustering Approach

Pearly Charles¹, Dr. S.Vimala²

¹Ph.D. Research Scholar, ²Associate Professor and Head
Department of Computer Science, Mother Teresa Women’s University, Kodaikanal, TamilNadu, India

Abstract— Cloud Computing is an emerging technology to share the resources in an efficient way. In this technology, every resource is offered as service over the internet. The services are concurrently accessed through the servers, it leads to an overloading problem. This issue can be overcome by an efficient load balancing approach. Load balancing is the process, of assigning the workload to the Pool of computers. In this process, the system should maintain the stability and satisfy the user needs. To perform this many load-balancing algorithms were proposed. Recently, research is done to improve the existing load balancing approach. Clustering approach is one such technique. In this research work the main objective is to design an efficient load balancing technique that use clustering approach to distribute workload evenly among the available virtual machines in a data center. At the same time the overall response time is also reduced.

Keywords— Cloud Computing, Clustering, Load Balancing, K-Medoids, Virtual Machine

I. INTRODUCTION

Load balancing technique distributes the entire load into smaller loads. In cloud computing environment it plays a vital role to distribute the workload to all nodes. The benefits of workload distribution increase the resource utilization which in turn enhances the system performance. This technique is divided into two categories static and dynamic load balancing approach. In static approach, the load is evenly distributed to all the servers and the system should have a prior knowledge about the available resources. In dynamic approach, the workload is distributed during the runtime. Therefore, the system should require the current state of information. Handling this type of load balancing creates the biggest issue in cloud computing [1].

II. LITERATURE REVIEW

Nakai et al. introduced a load-balancing algorithm based on the life of Bee. This algorithm is developed based on the behaviour of Bee. The algorithm produces the less response time compared with the existing load balancing algorithms [2].

Kumar et al. [14] proposed a CLBVM approach to handle the load balancing process in the cloud environment. The developed approach balanced the load uniformly to the distributed virtual machine [3].

Bhoi et al introduced an algorithm by enhancing the Max-Min Algorithm. The algorithm produces minimum execution time compared with the existing load balancing algorithms [4].

Bhawna et al. developed a new dynamic load-balancing algorithm. The algorithm assigns the user requested jobs to VMs based on the current status of the VMs. The result of this algorithm shows less response time than the other load balancing algorithms [5].

Salehi et al. proposed a load-balancing algorithm based on the behavior of ants intelligence. The algorithm efficiently handle the load from different users. The algorithm outperforms the existing algorithms [6].

Ramesh et al. proposed load balancing algorithm based on the heuristic approach called VectorDot. The algorithm designed on the behaviour of the ants. The algorithm handles the complexity of the data centers during the resource allocation [7].

Elzeiki et al introduced a new algorithm by enhancing the Max-Min load-balancing algorithm. This algorithm efficiently distributed the load to the servers. The algorithm is compared with the existing algorithm. The result shows that the algorithm performs better than the existing algorithm [8].

Vijay et al proposed an enhanced load-balancing algorithm. This algorithm schedules the task based on the available VMs. The result shows an improvement in the response time [9].

Nae et al developed a static load-balancing algorithm. This algorithm uses the Ants behaviour to collect information of cloud node to assign task to the particular node. The result of the algorithm shows the better performance and improved response time [10].

Utkarsh et al. proposed a load balancing algorithm using the k-means clustering approach. In this approach, the client’s requests are clustered based on the file size. An index table is maintained for the available VMs. Thus the job is assigned to the available VMs [11].

III. METHODOLOGY

In cloud computing environment, the load balancing approach is mainly used to improve the cloud performance and to attain the optimal resource utilization. The main aim of this approach is to distribute the total load into smaller even units of individual load to the various virtualized computing nodes. In this research work a new algorithm is proposed to handle the efficient load balancing approach.

In the proposed algorithm, the client requested jobs are clustered using the K-Medoids to find the priority of the jobs. The same algorithm is used to cluster the VMs based on the respective parameters. The Fig.1 shows the workflow of the proposed approach.
A. Algorithm

S = \{x_i| x_i\text{ is a job}\}
T = \{y_j| y_j\text{ is a vm}\}
Input S
for each x
    Compute the length of x
end
apply k-medoids and divide S into Two Clusters
J, K are assumed centroids
for each x
    Distance is computed for x
    if distance! = new distance
        Add Sx into minimum-clustered distance
    end for
for each y in T.
    calculate the capacity of y
end
apply k-medoids and divide T into Two Clusters
for each y
    Distance is computed for y with the centroids.
    if distance! = new distance
        Add Ty into min distance Cluster
    end for
end
 assign x \rightarrow y from appropriate cluster
end

B. Simulation Analysis

The proposed algorithm has to handle the entire load in the cloud environment. In this simulation different number of cloudlets and VMs are executed based on its priority. The Response Time parameter is considered to prove the efficiency of the algorithm. Therefore, the Response Time is computed by the following formula,

\[ RT = FT - ST \]

Where

RT - Response Time
FT - Finish Time
ST - Start Time

The proposed algorithm is analyzed using the CloudSim simulator. The obtained result is tabulated in table 1. In this table the values of the result is measured in terms of milliseconds.

<table>
<thead>
<tr>
<th>No. of Cloudlets</th>
<th>Start Time (ms)</th>
<th>Finish Time (ms)</th>
<th>Response Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>100</td>
<td>225</td>
<td>125</td>
</tr>
<tr>
<td>50</td>
<td>225</td>
<td>412</td>
<td>187</td>
</tr>
<tr>
<td>100</td>
<td>150</td>
<td>412</td>
<td>262</td>
</tr>
<tr>
<td>150</td>
<td>412.5</td>
<td>732.5</td>
<td>320</td>
</tr>
<tr>
<td>200</td>
<td>522.5</td>
<td>952.5</td>
<td>430</td>
</tr>
</tbody>
</table>

Table 2: Comparative Analysis Proposed and Existing Algorithms

<table>
<thead>
<tr>
<th>No. of Cloudlets</th>
<th>Existing Approach</th>
<th>Proposed Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>158</td>
<td>125</td>
</tr>
<tr>
<td>50</td>
<td>230</td>
<td>187</td>
</tr>
<tr>
<td>100</td>
<td>282</td>
<td>262</td>
</tr>
<tr>
<td>150</td>
<td>348</td>
<td>320</td>
</tr>
<tr>
<td>200</td>
<td>590</td>
<td>430</td>
</tr>
</tbody>
</table>

Fig 1: Workflow of Proposed Load Balancing Approach

Fig 2: Comparative Analysis of Existing and Proposed Algorithm
IV. CONCLUSION

Cloud computing is an advent technology to utilize the resources in an efficient manner over the internet. In this technology, load balancing has become one of the greatest issue. Many techniques were used in the past literature to handle the issue but still, the issue persist. In this research work, a new clustering load balancing approach is proposed to overcome the issue. The proposed approach is implemented using the cloud sim simulator and the result is obtained. The obtained result is compared with the existing load balancing approach. The comparison result shows that the proposed approach takes lesser response time that the existing approach. Hence the proposed approach will perform the load balancing process efficiently over the cloud environment.

REFERENCES