A New Era to Balance the Load on Cloud using ACCLB Hybrid Load Balancing Technique

Gagandeep Kaur*, Er. Sushil Kamboj
SUSGOI, Tangori
Gagandeepkaur1087@gmail.com*, er.kamboj@gmail.com

Abstract: Cloud computing is rapidly improving the latest technology. It is a term which is generally used in internet. This technological trend has enabled the realization of a new computing model called cloud computing, in which shared resources, information, software & other devices are provided according to client requirement at specific time, are provided as general utilities that can be leased and released by users through the Internet in an on-demand fashion. In cloud computing, load balancing is required to distribute the dynamic local workload evenly across all the nodes. It helps to achieve a high user satisfaction and resource utilization ratio by ensuring an efficient and fair allocation of every computing resource. Load balancing in large distributed server systems is a complex optimization problem of critical importance in cloud systems and data centers. Load balancing algorithms are classified as static and dynamic algorithms. Static algorithms are mostly suitable for homogeneous and stable environments and can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes to the attributes during the execution time. In this paper, we have studied and implemented three algorithm using Java Programming and simulate the algorithms on Cloudsim. Cloudsim provides novel support for modeling and simulation of virtualized Cloud based data center environments such as dedicated management interfaces for VMs, memory, storage, and bandwidth. CloudSim layer manages the instantiation and execution of core entities (VMs, hosts, data centers, application) during the simulation period. And, we have proved that our proposed algorithm ACCLB gives the better results as compared to Vector Dot and Join Idle queue. We analyzed the results on the basis of different performance parameters such as Response time, Total Execution time and Energy consumption.

Indexed Terms: ACCLB, Join idle queue, Vector Dot, VMs, Data Center, Load balancing, Cloudsim

I. INTRODUCTION

Cloud computing is a new technology and it is becoming popular because of its great features. In this technology almost everything like hardware, software and platform are provided as a service. A cloud provider provides services on the basis of client’s requests. An important issue in cloud is, scheduling of users requests, means how to allocate resources to these requests, so that the requested tasks can be completed in a minimum time and the cost incurred in the task should also be minimum. In case of Cloud computing services can be used from diverse and wide spread resources, rather than remote servers or local machines. There is no standard definition of Cloud computing. Generally it consists of a bunch of distributed servers known as masters, providing demanded services and resources to different clients known as clients in a network with scalability and reliability of datacenter. Cloud computing is not a single type of system, but it encompasses a range of underlying technologies and configuration options. The strengths and weaknesses of the different cloud technologies, configurations, service models, and deployment methods should be considered by organizations evaluating services to meet their requirements.

A. Cloud Service Models

![Cloud Computing Service Models](image)

Figure 1. Cloud Computing Service Model [5]

- **Infrastructure-as-a-service (IaaS)**
  Infrastructure-as-a-service is a provisioning model in which an organization outsources the equipment used to support operations, including hardware, servers, and storage and networking components. The service provider has the equipment and is creditworthy for housing, running, and maintaining it. The [4] client typically pays on a per-use basis. Services provided in Infrastructure-as-a-service are: Storage as a service, Compute as a service, Network as a service, Disaster recovery as a service, High availability services, Backup as a service, Infrastructure for application development and testing, Virtual private cloud.

- **Platform-as-a-service (PaaS)**
  Platform-as-a-service [6] is a way to rent operating systems, storage, hardware and network capacity over the Internet. The service delivery model grants the customer to rent virtualized servers and associated services for running existing applications or developing and testing new ones. Examples of PaaS providers include Microsoft Windows Azure Platform, force.com and Google App Engine.

- **Software-as-a-service (SaaS)**
  Software-as-a-service [8] is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network, typically the Internet. The benefit of SaaS clouds is that clients only focus on the use of the software and do not have to worry about the cost and effort to keep software licenses current, nor the dealing of software updates. Examples of SaaS providers include Salesforce.com, Rack space.

B. Load Balancing On Cloud Computing

With the increasing popularity[10] of cloud computing, the amount of processing that is being done in the clouds is
surging drastically. A cloud is constituted by various nodes which perform computation according to the requests of the clients. As the requests of the clients can be random to the nodes they can vary in quantity and thus the load on each node can also vary. Therefore, every node in a cloud can be unevenly loaded with tasks according to the amount of work requested by the clients. This phenomenon can drastically reduce the working efficiency of the cloud as some nodes which are overloaded will have a higher task completion time compared to the corresponding time taken on an under loaded node in the same cloud. This problem is not only confined only to cloud but is related with every large network like a grid, etc.

Load balancing in large distributed server systems is a complex optimization problem of critical importance in cloud systems and data centers. Load balancing algorithms are classified as static and dynamic algorithms. Static algorithms are mostly suitable for homogeneous and stable environments and can produce very good results in these environments. However, they are usually not flexible and cannot match the dynamic changes to the attributes during the execution time. Dynamic algorithms are more flexible and take into consideration different types of attributes in the system both prior to and during run-time [2]. These algorithms can adapt to changes and provide better results in heterogeneous and dynamic environments. However, as the distribution attributes become more complex and dynamic. As a result some of these algorithms could become inefficient and cause more overhead than necessary resulting in an overall degradation of the services performance.

Parameters for Load Balancing

There are mainly three parameters and their sub-parameters which may participate in the decision process of load balancing and hence may affect the performance of cloud system. Network parameter has some sub-parameters like Inter-communication delay, available bandwidth, and communication link power and network latency. No. of available Parallel Elements, processing power and memory capacity are the characteristics of computing node. Application can affect the performance due to its execution time, pre-emptive and non-pre-emptive characteristics.

II. RELATED WORK

In complex and large systems, there is a tremendous need for energy efficiency as well as load balancing on cloud environment. For simplifying such needs globally one thing which can be done is, employing techniques would act at the components of the clouds in such a way that the load of the whole cloud is balanced and results in energy efficiency.

Ebin Deni Raj et al. (2015) [2] described Big Data and parallel computing are used extensively for processing large quantities of data, structured, semi structured or totally unstructured. MapReduce and Hadoop are used for the parallel data processing of these kinds of data. Various scheduling policies are used for MapReduce scheduling which is discussed in detail and a new scheduling technique Two Phase Scheduling Policy (TPSP) based resource allocation for MapReduce is implemented and the efficiency is verified.

Wei Deng et al. (2014) [3] proposed the Harnessing Renewable Energy in Cloud Datacenters. They provided taxonomy of the state-of-the-art research in applying renewable energy in cloud computing datacenters from 5 key aspects, including propagation models and prediction methods of renewable energy, content planning of green datacenters, intra-datacenter workload scheduling and load balancing across geographically distributed datacenters.

Patrick Raycroft et al. (2014) [11] analyzed the effects of global virtual machine allocation on energy consumption, using a kind of real-world policies and a realistic testing scenario. They found that by using an allocation policy designed to minimize energy, total energy consumption could be minimized by up to 14%, and total monetary energy costs could be reduced by up to 26%. Further, they have begun performance qualification of their energy cost driven allocation policies through network capability tests. The results indicate that performance and IaaS provider implementation costs have a significant influence on selection of optimal virtual machine allocation policies. By choosing a more energy efficient allocation policy, energy consumption on cloud platforms can potentially be reduced by approximately 7% to 14%, lowering overall energy costs by anywhere from 11% to 26%. Such an improvement comes at the cost, however, of increased CPU load.

S.K. Tesfatsion et al. (2014) [12] presented a system that integrates hardware and software management techniques to improve energy efficiency. They combined horizontal and vertical scaling to provide a richer choice for applications with different requirements and combine with a hardware approach to change the server CPU frequency. They experimentally determined a system model that captures the relationship between the inputs (CPU frequency, number of VMs, and number of cores) and the outputs (power consumption and performance). They adopted an online approach to update the system model based on real-time measurements of performance and power usage. By this, the system behavior adapts to needs of individual applications under different workload conditions. They also presented a feedback controller that determines an optimal configuration to minimize energy consumption while meeting the performance target. The results show that their combined approach achieves up to 34% energy savings compared to the constituent approaches—core change, virtual machine change, and CPU frequency change policies, while meeting the performance target.

Rajesh Gorge Rajan et al. (2013) [15] have the investigated the different algorithms proposed to resolve the issue of load balancing and task scheduling in Cloud Computing. They discussed and compared these algorithms to provide an overview of the latest approaches in the field. Load Balancing is essential for efficient operations in distributed environments. As Cloud Computing is growing rapidly and clients are demanding more services and better results, load balancing for the Cloud has become a very interesting and important research area. Many algorithms were suggested to provide efficient mechanisms and algorithms for assigning the client’s requests to available Cloud nodes. These approaches aim to enhance the overall performance of the Cloud and provide the user more satisfying and efficient services.
III. PROBLEM FORMULATION

Load balancing can be define as a method for distributing workload on the multiple computers or a computer cluster through network links to achieve optimal resource utilization which maximizes throughput and minimizes overall response time. It minimizes the total waiting time of the resources as well as avoids too much overload on the resources. In this technique traffic is divided among servers, so that data can be sent and received without maximum delay. Load balancing is a methodology to distribute work load across multiple computers, or other resources over the network links to achieve the optimal resource utilization, minimum data processing, minimum average response time and avoid overhead. In the past number of load balancing algorithms have been developed specifically to suit the dynamic cloud computing environments such as WLC (Weighted Least Connection) algorithm, LBMM (Load Balancing Min-Min) algorithm, Bee-MMT (Artificial Bee Colony algorithm Minimal Migration Time), active Clustering algorithm, Honeybee Foraging Algorithm. In this paper, we are proposing the ACCLB-based on ant colony and complex network theory (ACCLB) to balance the load on the cloud and compared it with the existing load balancing methods such as Vector Dot and Join idle queue. The main objective of the research is balance the load on cloud and consumes less energy as compared to previous, on the cloud by using proposed methods. Also we have to prove that our proposed techniques are more efficient for load balancing and energy consumption on cloud as compared to previous.

A. METRICS FOR LOAD BALANCING IN CLOUDS

Various metrics will be considered in load balancing techniques in cloud computing are discussed below.

1. Throughput is used to calculate the no. of tasks whose execution has been completed. It should be high to improve the performance of the system.

2. Overhead Associated determines the amount of overhead involved while implementing a load-balancing algorithm. It is composed of overhead due to movement of tasks, inter-processor and inter-process communication. This should be minimized so that a load balancing technique can work efficiently.

3. Fault Tolerance is the ability of an algorithm to perform uniform load balancing in spite of arbitrary node or link failure. The load balancing should be a good fault-tolerant technique.

4. Response Time is the amount of time taken to respond by a particular load balancing algorithm in a distributed system. This parameter should be minimized.

5. Resource Utilization is used to check the utilization of resources. It should be optimized for an efficient load balancing.

6. Scalability is the ability of an algorithm to perform load balancing for a system with any finite number of nodes. This metric should be improved.

7. Performance is used to check the efficiency of the system. This has to be improved at a reasonable cost, e.g., reduce task response time while keeping acceptable delays.

IV. RESEARCH OBJECTIVES

The Objective of the proposed work is to balance the load and energy efficient using Dynamic load balancing model for cloud computing architecture.

The Objectives of the research work are:

1. To optimize the performance of cloud architecture.

2. To implement proposed method ACCLB using java programming and simulate on cloud computing environment using CloudSim toolkit.

3. To compare the ACCLB with exiting load balancing techniques such as Vector Dot and Join- Idle queue for energy efficiency and load balancing.

4. To analyze the behavior of ACCLB using following parameters-
   - Energy Consumption
   - Response time
   - Total Execution Time

5. To Evaluate the performance and behavior of proposed ACCLB load balancing technique by comparing it with existing load balancing methods such as vector Dot and Join-Idle queue.

V. PROPOSED WORK

ACCLB a load balancing mechanism based on ant colony and complex network theory (ACCLB) in an open cloud computing federation. It uses small-world and scale-free characteristics of a complex network to achieve better load balancing. This technique overcomes heterogeneity, is adaptive to dynamic environments, is excellent in fault tolerance and has good scalability hence helps in improving the performance of the system.

A. Simulation Results

When all algorithms are selected for load balancing, simulation results are as depicted in figure 2. for 5 servers.

Figure 2: Output of the algorithms for 5 Servers
B. Comparison Tables

All three algorithms are implemented and compared on CloudSim tool for energy efficiency and load balancing. Table 5.4 depicts the result of different parameters for five jobs. During the comparison, ACCLB technique is counted as best model for producing the good results according to the user requirements.

Table 1: Comparison between three algorithms in case of 5 servers

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Number of jobs/Tasks</th>
<th>Response Time</th>
<th>Execution Time</th>
<th>Energy Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCLB</td>
<td>For #5</td>
<td>1.5</td>
<td>7.3</td>
<td>53</td>
</tr>
<tr>
<td>Join idle queue</td>
<td>For #5</td>
<td>1.6</td>
<td>13.85</td>
<td>11.18</td>
</tr>
<tr>
<td>Vector Dot</td>
<td>For #5</td>
<td>1.7</td>
<td>13.16</td>
<td>10.56</td>
</tr>
</tbody>
</table>

Table 2: Comparison between three algorithms in case of 10 servers

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Number of jobs/Tasks</th>
<th>Response Time</th>
<th>Execution Time</th>
<th>Energy Consumption</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCLB</td>
<td>For #10</td>
<td>2.1</td>
<td>10.75</td>
<td>0.67</td>
</tr>
<tr>
<td>Join idle queue</td>
<td>For #10</td>
<td>3.3</td>
<td>19</td>
<td>24.75</td>
</tr>
<tr>
<td>Vector Dot</td>
<td>For #10</td>
<td>3.4</td>
<td>58.44</td>
<td>45.89</td>
</tr>
</tbody>
</table>

All three algorithms are compared for energy efficiency and load balancing. The results show that ACCLB technique is the better because which consumes less energy and all the tasks are executed in less time with no delay. It concluded that ACCLB is best in the energy efficient technique in cloud computing.

C. Graphical Analysis

Graphical charts shows the comparison between different techniques based on distinct number of jobs considering discrete parameters like energy consumption, response time and execution time.

Figure 4 reveals the information about the energy consumption of datacenter and the results show that ACCLB technique consumed less energy as compared to Vector Dot and Join idle queue. Therefore our proposed model is best in case of energy efficient.
VI. CONCLUSION & FUTURE WORK

Cloud computing is one of the most prominent technologies. The fundamental idea behind cloud computing is to distribute an array of computing services by unifying and scheduling a pool of computing resources, thereby minimizing the burden on the users and helping them focus on their core businesses. These computing resources are hosted on virtual hosts and distributed on-demand to the users by cloud service providers. For efficient resource utilization, systematic load balancing of incoming user traffic across virtual hosts is imperative. Load balancing is defined as the allocation of the work of a single application to processors at run-time so that the execution time of the application is minimized. In the research work we have proposed and implemented an ACCLB on cloud environment using CloudSim Toolkit. And, also compared with the Vector Dot and Join Idle queue. The results show that proposed technique is much better than the existing load balancing methods in terms of Response time, Execution Time, and Throughput. The thesis also concluded that ACCLB technique consumes less energy than existing techniques.

A. Future Work

Cloud Computing is a vast concept and energy efficiency plays a very important role in case of Clouds. There is a huge scope of improvement in this area. We have implemented only two dynamic load balancing algorithms. But there are still other approaches that can be applied to balance the load and energy consumption in clouds. The performance of the given algorithms can also be increased by varying different parameters. We can also move our research work on any Private Cloud for the Security and further enhancements. In future work, we can enhance the performance of other cloud models and also by considering some other parameters according to environment. As a future scope we need to enhance the proposed ACCLB algorithm with wider perspective and can be experimented with machines with different configurations. Further we need to optimize a cloud environment by incorporating bio-inspired algorithms along with proposed algorithm.

REFERENCES


