A Systemmatic way to balance the load on cloud using Hybrid (OLB+LBMM) Technique

Navpreet kaur aulakh¹, Rishideep Singh²

North West Institute of Engineering & Technology,
¹naviaulakh@yahoo.in, ²hod_it@northwest.ac.in

Abstract: Cloud computing is an emerging computing paradigm with a large collection of heterogeneous autonomous systems with flexible computational architecture. Load Balancing is an important step to improve the overall performance of the cloud computing. Load Balancing is also essential to reduce power consumption and improve the profit of service providers by reducing processing time. In cloud computing environment, user services always demand heterogeneous resources (e.g., CPU, I/O, Memory 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 can take into consideration different types of attributes in the system both prior to and during run-time. Load balancing aims to optimize resource use, maximize throughput, minimize response time, and avoid overload of any single resource. In the paper, we have studied and implemented three algorithm using Java language and simulate using CloudSim. CloudSim also offers 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. Hence it is proven that proposed algorithm OLB+LBMM gives the better results as compared to Honeybee Foraging Behavior and Active Clustering. We also analyzed the research results on the basis of distinct performance parameters such as Response time, Tofal Execution time, Throughput, Processing Time and Energy consumption.

Indexed Terms: Load balancing, OLB+LBMM, Active Clustering, Honeybee, Virtual Machines, API.

I. INTRODUCTION

Today Cloud Computing is very vast technology among new generation. It is Very Popular because of its new features. It Provides Very interesting facilities so it is most commonly use. Cloud Computing is also known as Internet Based Computing because in this technology Resources are shared over the internet So it is called “Internet Based Computing”. Cloud Computing is combination of two words Cloud and computing where cloud is used as metaphor for “Internet” and computing means “to compute”. Cloud computing means that instead of all the computer hardware and software you’re using sitting on your desktop, or somewhere inside your company’s network, it’s provided for you as a service by another company and accessed over the Internet, usually in a completely seamless way. Exactly where the hardware and software is located and how it all works doesn’t matter to you, the user. Cloud Computing is very much similar to Grid Computing but the difference is in Grid Computing resources are shared over the grid like Structure but in cloud computing Resources are shared over the internet. The main enabling technology for cloud computing is virtualization. Virtualization software separates a physical computing device into one or more “virtual” devices, each of which can be easily used and managed to perform computing tasks. The number of users working on the cloud is increasing day by day because of its efficiency. But to handle the resources over the cloud is also become important. To solve this problem Load Balancing is use. Load balancing is dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time and, in general, all users get served faster. On the Internet, companies whose Web sites get a great deal of traffic usually use load balancing. To balance the load of Resources over the cloud i.e. Internet in this paper we will use “Cat Swarm Optimization Technique” to overcome the problem of “Cloud Resource Balancer”.

A. Cloud Computing Application Architecture

As we know that cloud computing is the shift of computing to a host of hardware infrastructure that is distributed in the cloud. The commodity hardware infrastructure consists of the various low cost data servers that are connected to the system and provide their storage and processing and other computing resources to the application. Cloud computing involves running applications on virtual servers that are allocated on this distributed hardware infrastructure available in the cloud. These virtual servers are made in such a way that the different service level agreements and reliability issues are met. There may be multiple instances of the same virtual server accessing the different parts of the hardware infrastructure available. This is to make sure that there are multiple copies of the applications which are ready to take over on another one’s failure. The virtual server distributes the processing between the infrastructure and the computing is done and the result returned.

Figure 1. Cloud Computing Application Architecture
There will be a workload distribution management system, also known as the grid engine, for managing the different requests coming to the virtual servers. This engine will take care of the creation of multiple copies and also the preservation of integrity of the data that is stored in the infrastructure. This will also adjust itself such that even on heavier load, the processing is completed as per the requirements. The different workload management systems are hidden from the users. For the user, the processing is done and the result is obtained. There is no question of where it was done and how it was done. The users are billed based on the usage of the system - as said before - the commodity is now cycles and bytes. The billing is usually on the basis of usage per CPU per hour or GB data transfer per hour.

B. Enabling Technologies

Before going into the idea of cloud computing, two technologies will be introduced that made the way of distributed computing and therefore cloud computing realizable.

➢ Virtualization

With virtualization, applications and infrastructure are independent, allowing servers to be easily shared by many applications where applications are running virtually anywhere in the world. This is possible as long as the application is virtualized. Virtualizing the application for the cloud means to package the bits of the application with everything it needs to run, including pieces such as a database, middleware and an operating system. This self-contained unit of virtualized application can then run anywhere in the world. Virtualization also allows building so-called sandboxes. Sandboxes assure a higher degree of security and reliability by providing a mechanism to run programs safely. It is commonly used to "execute untested code, or programs from unverified third-parties, suppliers and un-trusted users".

Two types of virtualization are found in case of clouds as given in [1]:

- Full virtualization
- Para-virtualization

➢ Full Virtualization

In case of full virtualisation a complete installation of one machine is done on the other machine. It will result in a virtual machine which will have all the softwares that are present in the actual server.

➢ Para-virtualization

In para-virtualization, the hardware allows multiple operating systems to run on single machine by efficient use of system resources such as memory and processor, e.g. VMware software. Here all the services are not fully available, rather the services are provided partially.

C. Load Balancing On Cloud Computing

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.

D. Dynamic Load Balancing Algorithm

In a distributed system, dynamic load balancing can be done in two different ways: distributed and non-distributed. In the distributed one, the dynamic load balancing algorithm is executed by all nodes present in the system and the task
of load balancing is shared among them. The interaction among nodes to achieve load balancing can take two forms: cooperative and non-cooperative [4]. In the first one, the nodes work side-by-side to achieve a common objective, for example, to improve the overall response time, etc. In the second form, each node works independently toward a goal local to it, for example, to improve the response time of a local task. Dynamic load balancing algorithms of distributed nature, usually generate more messages than the non-distributed ones because, each of the nodes in the system needs to interact with every other node. A benefit of this is that even if one or more nodes in the system fail, it will not cause the total load balancing process to halt, it instead would effect the system performance to some extent. Distributed dynamic load balancing can introduce immense stress on a system in which each node needs to interchange status information with every other node in the system. It is more advantageous when most of the nodes act individually with very few interactions with others.

In non-distributed type, either one node or a group of nodes do the task of load balancing. Non-distributed dynamic load balancing algorithms can take two forms: centralized and semi-distributed. In the first form, the load balancing algorithm is executed only by a single node in the whole system: the central node. This node is solely responsible for load balancing of the whole system. The other nodes interact only with the central node.

In semi-distributed form, nodes of the system are partitioned into clusters, where the load balancing in each cluster is of centralized form. A central node is elected in each cluster by appropriate election technique which takes care of load balancing within that cluster. Hence, the load balancing of the whole system is done via the central nodes of each cluster [4]. Centralized dynamic load balancing takes fewer messages to reach a decision, as the number of overall interactions in the system decreases drastically as compared to the semi-distributed case. However, centralized algorithms can cause a bottleneck in the system at the central node and also the load balancing process is rendered useless once the central node crashes. Therefore, this algorithm is most suited for networks with small size.

II. RELATED WORK

Load balancing in the cloud computing environment has an important impact on the performance. Good Load balancing makes cloud computing more efficient and improves user satisfaction. There have been many studies of load balancing for the Cloud environment.

Bernardetta Addis et al. [4] in 2014 proposed a new optimization framework for the management of the energy usage in an integrated system for Cloud services that includes both service centers and communication networks for accessing and interconnecting them. The optimization framework considers a PaaS scenario where VMs serving an application can be allocated to a set of SCs geographically distributed and traffic load coming from different world regions can be assigned to VMs in order to optimize the energy cost and minimize CO2 emissions. Numerical results, on a set of randomly generated instances and a case study representative of a large Cloud provider, shows that the availability of green energy have a big impact on optimal energy management policies and that the contribution of the network is far from being negligible.

Rajesh Gorge Rajan et al. [5] in 2013 have 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.

Suriya Begum et al. [6] in 2013 have described the random arrival of load in such an environment can cause some server to heavily loaded while other server is idle or only lightly loaded. equally load distributing enhance performance by transferring load from heavily server. Efficient scheduling and resource allocation is a critical characteristics of cloud computing based on which performance can be estimated. It is required to Distribute the dynamic load workload evenly across all the nodes to achieve the high user satisfaction and resource utilization ration by making sure that every computing resource is distributed efficiently and fairly.

Argha Roy et al. [7] in 2013 presented the researcher proposed the idea of dynamic load balancing. Researcher concluded that dynamic load balancing is a technique to use the cloud computing in efficient manner .The virtual machine algorithm used in the approach can automatically monitor the load balancing with the use of load balancer. The researcher has avoided the data migration .Job migration and static load balancer.

Meysm Orouskhani et al. [8] in 2013 proposed the improved CSO algorithm namely “Adaptive dynamic cat Swarm Optimization”The Paper described the addition of a new adaptive interia weight to velocity equation and use of adaptive acceleration ratio. The proposed CSO take less time to converge and can find best solution in less iteration.

Amir Nahir et al. [9] in 2013 presented the approach which is based on creating several replicas of each job and sending each replica to a different server. Upon the arrival of a replica to the head of the queue at its server, the latter signals the servers holding replicas of that job, so as to remove them from their queues. They show, through an analysis and simulations, that this scheme improves the expected queuing overhead over traditional schemes by a factor of 9 (or more) under various load conditions. In addition, we show that our scheme remains efficient even when the inter-server signal propagation delay is significant (relative to the job’s execution time). They provided heuristic solutions to the performance degradation that occurs in such cases and show, by simulations, that they efficiently mitigate the detrimental effect of propagation delays. Finally, they demonstrated the efficiency of proposed scheme in a real-world environment by implementing a load balancing system based on it.
deploying the system on the Amazon Elastic Compute Cloud (EC2), and measuring its performance.

Parveen Patel et al. [10] in 2013 proposed an approach that shows that Layer-4 load balancing is fundamental to creating scale-out web services. We designed and implemented Ananta, a scale-outlayer-4 load balancer that runs on commodity hardware and meets the performance, reliability and operational requirements of multi-tenant cloud computing environments. Ananta combines existing techniques in routing and distributed systems in a unique way and splits the components of a load balancer into a consensus-based reliable control plane and a decentralized scale-out data plane.

Isam Azawi Mohialdeen [11] in 2013 has discussed, the behavior of four job scheduling algorithms, namely: Random, Round-Rubin (RR), Opportunistic Load Balancing and Minimum Completion Time have been investigated and examined in a Cloud computing environment. Based on the results, it can be also concluded that there is not a single scheduling algorithm that provides superior performance with respect to various types of quality services. This is because job scheduling algorithms needs to be selected based on its ability to ensure good quality of services with reasonable cost and maintain fairness by fairly distribute the available resources among all the jobs and respond to the constraints of the users.

III. RESEARCH PROBLEM
Load Balancing is defined as a process of making effective resource utilization by reassigning the total load to the individual nodes of the collective system and thereby minimizing the response time of the job. Load Balancing algorithms are classified as Static and Dynamic algorithms. Static algorithms are most suitable for homogenous and stable environments. However, they cannot match the dynamic changes to the attributes during execution time. Dynamic algorithms take into consideration different types of attributes in the system both prior to and during run time. These algorithms can adapt to changes and provide better results in heterogeneous and dynamic environments.

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).

In this research, we are proposing the OLB + LBMM to balance the load on the cloud and compared it with the existing load balancing methods such as Honeybee Foraging Behavior, Active Clustering. 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 method. Also we have to prove that our proposed technique is 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 re-sources. 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.

B. 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.

1. To optimize the performance of cloud architecture.

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

3. To compare the OLB + LBMM with exiting load balancing techniques such as Active Clustering and Honeybee Foraging Behavior for energy efficiency.

4. To analyze the behavior of proposed method using following parameters-
   - Energy Efficient Metric
   - Execution time
   - Response time
   - Resource utilization
   - Fault Tolerance
   - Scalability
   - Throughput

5. To Evaluate the performance and behavior of proposed OLB + LBMM dynamic load balancing technique by comparing it with existing load balancing methods such as Honeybee Foraging Behavior and Active Clustering.

IV. RESULT AND DISCUSSION
Setup server_config.xml
- Initialize the Tomcat Server for project execution
- Send Request for the execution of the project
- Then select the specific algorithm for its execution
- Simulation Results using Tomcat Server
- CloudSim Results
- Output Tables
- Graphical Charts

A. Simulation Results

When all algorithm are selected for load balancing, simulation results are as depicted in figure 4 for 5 servers.

Figure 4. Simulation results of OLL+LBMM, Active Clustering and Honeybee Foraging for 5 servers

Figure 5. Simulations results of Algorithms for 6 servers

Figure 6. Simulations results of Algorithms for 8 Servers

Figure 7. Simulations results of Algorithms for 10 Servers

B. Output Tables

All three algorithms are implemented and compared on CloudSim tool for energy efficiency and load balancing. Table1. Depicts the result of different parameters for five jobs. During the comparison, OLB+LBMM technique is counted as best model for producing the good results according to the user requirements.

Table1. Different algorithms are compared for 5 Jobs.

<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>OLB+LBMM</td>
<td>For #5</td>
<td>1.5</td>
<td>7.3</td>
<td>391</td>
</tr>
<tr>
<td>Active Clustering</td>
<td>For #6</td>
<td>1.7</td>
<td>13.16</td>
<td>1056.30</td>
</tr>
<tr>
<td>Honeybee Foraging</td>
<td>For #5</td>
<td>1.6</td>
<td>13.85</td>
<td>1118</td>
</tr>
</tbody>
</table>

Table2. Depicts the result of different parameters for six jobs.

<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>OLB+LBMM</td>
<td>For #6</td>
<td>1.6</td>
<td>8.1</td>
<td>584</td>
</tr>
<tr>
<td>Active Clustering</td>
<td>For #6</td>
<td>2</td>
<td>15.10</td>
<td>1209.85</td>
</tr>
<tr>
<td>Honeybee Foraging</td>
<td>For #6</td>
<td>2.0</td>
<td>16.8</td>
<td>1331</td>
</tr>
</tbody>
</table>

Table 3. Different algorithms are compared for 10 Jobs.

<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>OLB+LBMM</td>
<td>For #10</td>
<td>1.0</td>
<td>10.75</td>
<td>67.77</td>
</tr>
<tr>
<td>Active Clustering</td>
<td>For #10</td>
<td>2.5</td>
<td>18.53</td>
<td>15.97</td>
</tr>
<tr>
<td>Honeybee Foraging</td>
<td>For #10</td>
<td>3.3</td>
<td>26.58</td>
<td>2092</td>
</tr>
</tbody>
</table>

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

C. Graphical Charts

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 8. reveals the information about the energy consumption of datacenter and the results show that OLB+LBMM technique consumed less energy as compared to Honeybee Foraging Behavior and Active Clustering. Therefore our proposed model is best in case of energy efficient.

Figure 8. Energy consumption comparison based on distinct number of jobs.

Response time is the time taken to respond by a particular load balancing algorithm in a distributed system. The response time is calculated by varying the number of jobs. Figure 9. depicts that the response time is minimum in case of OLB+LBMM technique.

Figure 9. Response time comparison based on distinct number of jobs.

Execution time is estimated by calculating the total number of jobs executed within a fixed span of time. Figure 10. shows the execution time taken by three techniques for distinct number of jobs. After implementation results show that the execution time of OLB+LBMM is less as compared to Honeybee Foraging Behavior and Active Clustering.

Figure 10. Execution time comparison based on distinct number of jobs.

V. CONCLUSION AND FUTURE WORK

Efficient load balancing of jobs in the cloud environment has become the important criteria to determine the system performance. To improve the user’s satisfaction in balancing of jobs there should be an efficient optimization algorithm to distribute the jobs in cloud environment which eventually leads to dynamic allocation of resources in distributed environment. Cloud computing has become a buzzword in the area of high performance distributed computing as it provides on-demand access to shared pool of resources over Internet in a self service, dynamically scalable and metered manner. Cloud computing is still in its infancy, so to reap its full benefits; much research is required across a broad array of topics. One of the important research issues which need to be focused for its efficient performance is scheduling. It has been shown that proper load balancing of computing resources can lead to a significant reduction of the energy consumption by a system, while still meeting the performance requirements. A relaxation of the performance constraints usually results in a further decrease of the energy consumption. Load balancing that is required to distribute the excess dynamic local workload evenly to all the nodes in the whole Cloud to achieve a high user satisfaction and resource utilization ratio.

In the research work we have proposed and implemented a OLB+LBMM on cloud environment using CloudSim Toolkit. And we have compared it with the Active Clustering and Honeybee Foraging Behavior. The results show that proposed technique is much better than the existing load balancing methods in terms of Response time, Execution Time, and Throughput. We also concluded that OLB+LBMM consumes less energy than Central Load Balancer.

A. Future Work

In these days, all the companies are using the concept of cloud computing. It is a new technology. It works based on Pay on Demand model. This model means, in which quantity user required the resources for a specific time to complete a task, user have to pay some money only for that
much of time. Due to this technology many benefits are applicable to get the better profit in the market. Profit in terms of time, cost, load balancing, and storage and so on. In this technology, all applications run on a virtual platform and all the resources are distributed among these virtual machines. Each and every application is different and is independent. We have implemented three 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.

REFERENCES


