An Efficient Approach for Load Balancing in Cloud Environment

Balasundaram Ananthakrishnan

Abstract— Cloud computing has become one of the key paradigms in distributed computing today. It provides the end users to access the resources in a distributed, scalable and virtualized fashion. The distributed nature of cloud in itself poses lots of challenges. Resource availability and load management across resources available in cloud environment pose a serious challenge. Thus load balancing plays a vital role in the performance of cloud computing. This paper focuses on listing the various techniques available for improving load balancing in cloud environment. Also, a comparative study of the load balancing techniques have been performed and tabulated. In addition, a new strategy to enhance load balancing in cloud environment has also been suggested.

Index Terms— Load Balancing, Cloud Computing, Load Balancing Algorithms, Cloudsim, Round Robin, Token Routing, Response Time.

1 Introduction

CLOUD computing refers to performing computing tasks using various services provided over the Internet. It promotes sharing of resources, software, and information through Internet. Cloud computing provides a shift away from applications that have to be installed on an the user's computer towards the applications being hosted online. The information is stored on physical servers that are maintained and controlled by a cloud service provider. Whenever information is needed, the user can access the stored information on the cloud using the Internet. Some common cloud computing based applications include email services such as Gmail and Hotmail, social applications such as Facebook, Twitter, LinkedIn etc., media services such as Youtube, VoIP services such as skype, cloud platforms such as Google Cloud Platform and cloud database such as CloudSQL.

1.1 Delivery Models in Cloud Computing

Cloud computing can be broadly classified into three major delivery models. Software as a Service (SaaS) is primarily focused on providing web services that cater to the user needs. Examples for SaaS include all the applications such as mail services, social network sharing services, media and VoIP services. Infrastructure as a Service (IaaS) aims at providing the necessary hardware and accessories over internet. A typical example for IaaS is Amazon Web Services which provides virtual server storage. Platform as a Service (PaaS) is more concentrated towards providing the development platform on top of which the cloud based applications can be built and implemented. A shining example for PaaS is Google Apps whose platform provides the necessary software and product development tools for creating different applications.

• *Balasundaram Ananthakrishnan is currently pursuing masters degree program in Computer Science and Engineering in B.S. Abdur Rahman University, Chennai, India, PH-09841288826.*
E-mail: balasundaram2682@gmail.com

1.2 Delivery Models in Cloud Computing

Fig. 1 shows the layered architecture of cloud computing implementation.

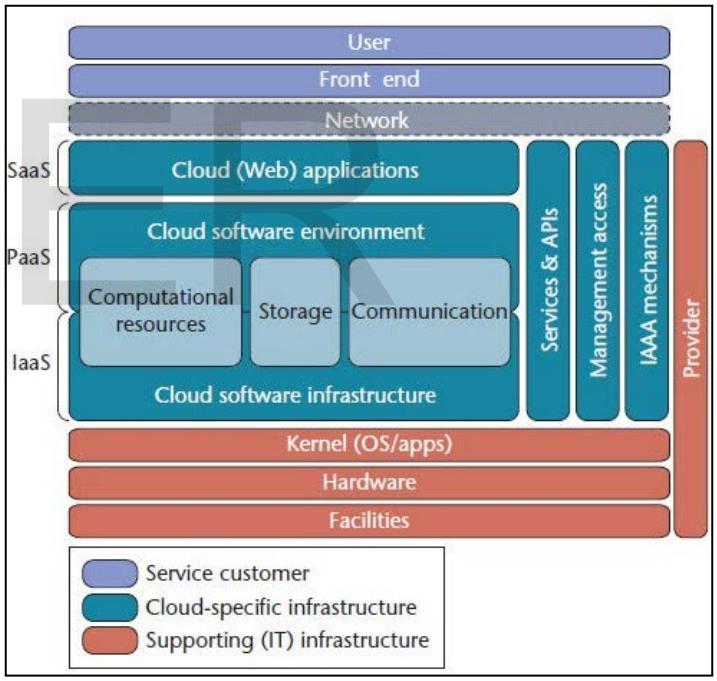

Fig. 1 Layered Architecture for Cloud Computing

It can be seen that the user accesses cloud specific layers (SaaS, PaaS and IaaS) through the network (Internet). The cloud related layers are generally managed by the service provider who is also responsible for taking care of the infrastructure and hardware components required to set the cloud environment.

1.3 Cloud Computing Advantages

Cloud computing offers numerous advantages to end users and also to businesses of all sizes. The biggest advantage is that there is no necessity for initial infrastructure setup or the knowledge necessary to develop and maintain the infrastruc-

ture, development environment or application. This provides the business enablers the flexibility to focus on their core business by outsourcing all IT infrastructures to the cloud service providers. The advantages that cloud computing provides are:

- Cost Saving: This is achieved by the elimination of the investment in stand-alone software or servers. By leveraging cloud's capabilities, companies can save on licensing fees and at the same time eliminate overhead charges such as the cost of data storage, software updates, management etc.
- Ubiquitous and Continuous availability: Cloud services are available wherever the end user might be located, thus enables easy access to information irrespective of different time zones and geographic locations. Moreover, cloud vendors typically use multiple servers for maximum redundancy. In case of system failure, alternative instances are automatically invoked on other machines.
- Backup and Recovery: Backing up and recovering of data is simplified since they now reside on the cloud and not on a physical device. The cloud providers offer reliable and flexible backup/recovery solutions. In some cases, the cloud itself is used solely as a backup repository of the data located in local computers.
- Resiliency and Redundancy: The cloud deployment is usually built on a robust architecture thus providing resiliency and redundancy to its users. The cloud offers automatic failover between hardware platforms out of the box, while disaster recovery services are also often included.
- Scalability and Performance: Cloud instances are deployed automatically only when needed and as a result, the user has to pay only for the applications and data storage they use.
- Increased Storage Capacity: Cloud computing offers unlimited storage capacity. It eliminates worries about running out of storage space and at the same time spares businesses the need to upgrade their computer hardware, further reducing the overall IT cost.

1.4 Cloud Computing Disadvantages

Any new concept has its share of advantages and disadvantages. The disadvantages of cloud computing are listed below.

- Cloud computing is totally dependent on internet connection. When the internet connection or network is down or when there is any network glitch, it has a direct bearing on the cloud environment as it may become inaccessible or may experience slowness.
- Cloud vendors ensure they have the latest, most sophisticated data security systems as data security is always a big concern for businesses which in turn accounts for additional cost. But there are still serious security concerns posed by hackers.
- Privacy is another area of concern. If a user can log in from any location to access data and applications, it's possible the user's privacy could be compromised.
- Since the Cloud infrastructure is entirely owned, managed and monitored by the service provider, it transfers minimal control over to the customer.
- The dynamic nature of cloud computing itself poses a huge challenge in managing the resources joining or leaving the cloud environment.
- Similarly, managing the load shared across the resources in cloud environment is another big challenge. Proper management of resources and the load they bear has a direct impact on the performance of cloud environment.

1.5 Need for Load Balancing

The main purpose of load balancing is to distribute the load evenly among the available resources thereby reducing the average response time. The jobs should not be let to wait in queue. The primary objectives of load balancing are listed below.

- Improve the cloud performance significantly by even distribution of load across available resources.
- To improve the overall stability of system.
- To accommodate fluctuating traffic and resources.
- To provide suitable backup in case of any resource failures.
- Ensuring that all the resources are optimally utilized and there is no over utilization or under utilization of resources.

This paper is focused towards making a detailed study of various load balancing algorithms available and to compare their benefits and drawbacks. Also, a new approach to perform efficient load balancing across cloud environment has been suggested.

2 Metrics And Policies

2.1 Metrics Related To Load Balancing

The following metrics [1] are used during the course of implementing the algorithms used for load balancing in cloud environment.

- Wait Time: This is the time consumed by a process waiting in the queue in ready state and about to get started.
- Reaction Time: This is the time consumed by the process to get started after it has been assigned to a resource.
- Context Switch: A context switch is the switching of the CPU from one job or thread to another. In this process, the state of the task or thread is stored and swapped quite frequently.

- Throughput: This is defined as the number of tasks that are completed in a given unit time.

2.2 Load Balancing Policies

Fig. 2 shows the different load balancing policies [2] employed in cloud computing.

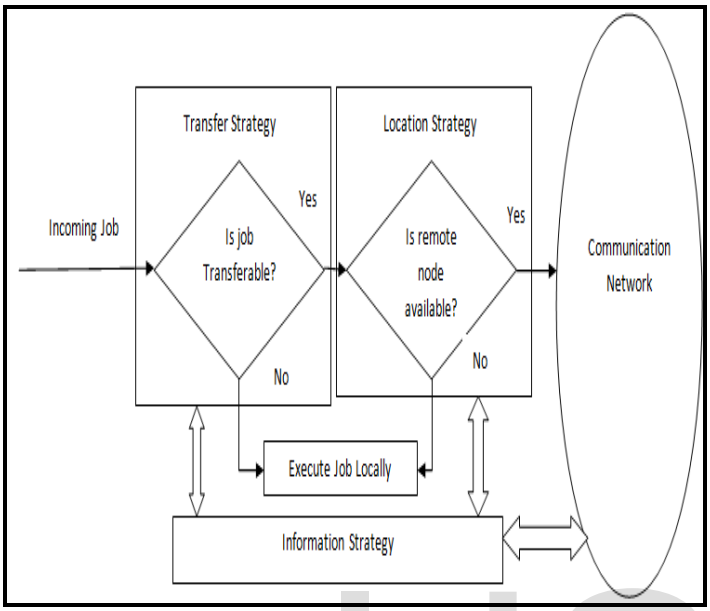

Fig. 2 Load Balancing Strategies

- Transfer policy: This policy is instrumental in determining whether the job has to be executed locally or whether it is transferable to any external nodes.
- Selection Policy: This policy denotes the processors involved in the load transfer.
- Location Policy: This policy determines the destination node where the job will be executed.
- Information Policy: This policy ensures all the necessary information pertaining to the individual nodes are collected.

3 Algorithms

Based on the nature of the algorithms, the load balancing algorithms can be classified into static and dynamic algorithms. Based on the environment deployed, the dynamic load balancing algorithms can further be classified into centralized and decentralized algorithms. Fig. 3 shows the classification of load balancing algorithms. Round Robin and Randomized algorithms are good examples for static load balancing algorithms while token ring, central queuing and least connection algorithms are typical examples of dynamic load balancing algorithms.

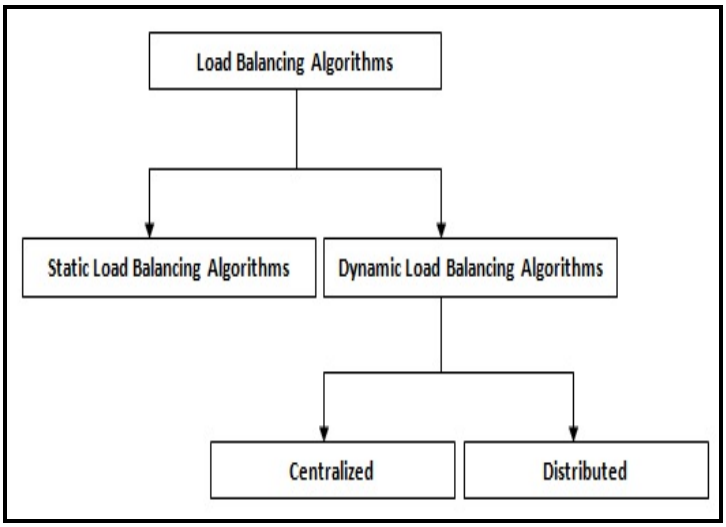

Fig. 3 Classification of Load Balancing Algorithms

3.1 Token Ring Algorithm

The primary objective of token routing algorithm [3] is to reduce the system cost by moving the tokens around the system. But in a scalable cloud system resources cannot have the enough information of distributing the work load due to lack or delay in communication. As a result, the load distribution among the resources is not fixed. The disadvantage of the token routing algorithm can be negated by using a heuristic approach of token based load balancing. This provides a fast and efficient routing decision. In this algorithm resources do not need to have the complete knowledge of their global state and work load of neighborhood resources. The decision where to pass the token is taken by referring to a knowledge base. This knowledge base is actually created from the previously received tokens. The primary advantage of this approach is that no communication overhead is generated.

3.2 Round Robin Algorithm

As part of Round Robin algorithm [4], the tasks are shared between all resources available. Each task is assigned to a resource in a round robin order. The task allocation order is maintained locally and is independent of the allocations from remote locations. Though the work load distributions between resources are equal, the job processing time for different tasks varies. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers where Http requests are of similar nature and distributed equally.

3.3 Randomized Algorithm

Randomized algorithm [4] is static in nature. In this algorithm a task can be handled by a particular node n with a certain probability p. The task allocation order is maintained for each resource independent of allocation from remote resources. This algorithm works well in case of tasks that are of equal load. However, problem arises when loads are of different computational complexities. Randomized algorithm does not use deterministic approach. It works well when Round Robin algorithm generates overhead for process queue.

3.4 Central Queuing Algorithm

Central queuing algorithm [5, 7] works on the principle of dynamic distribution. Each new task arriving at the queue manager is inserted into the queue. When request for an task is received by the queue manager it removes the first task from the queue and sends it back to the requester. If no ready task is present in the queue the request is buffered, until a new task is available. But in cases where a new task comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new task is assigned to it. When a resource load falls under threshold value then the local load manager sends a request for the new task to the central load manager. The central manager then answers the request if any ready task is found otherwise queues the request until new task arrives.

3.5 Least Connection Algorithm

Least Connection Algorithm [6, 8] takes into account the count of the number of connections for each server dynamically and estimates the load accordingly. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it, and decreases the number when connection finishes or timeout happens. Table 1 shows the comparison of various load balancing algorithms discussed so far.

TABLE I. LOAD BALANCING ALGORITHMS COMPARISON

Algorithm	Job / Task Migration	Resource Utilization	Stability
Token Routing	Possible	More	Less stable
Round Robin	Not possible	Less	Stable
Randomized	Not possible	Less	Stable
Central Queuing	Not possible	Less	Less stable
Least Connection	Not possible	Less	Less stable

Based on the literature study performed earlier, three major factors have been identified to have a huge impact on the performance of cloud environment. Job Migration is an important factor that decides the effective utilization of available CPU in the cloud space. If Job migration is possible, it implies that the tasks can be swapped or moved between processors whenever they are available. As a result, the throughput of the system increases as well. The next factor considered is Resource Utilization. This provides a measure of how effectively the resources in the cloud environment are utilized. The system should precisely know when the resource is freed and is ready to accept any new task or job. The primary objective of considering this factor is to improve effective utilization of individual resources and to minimize the idle time of resources. The third factor under consideration is the stability of the system. The stability of the system is based on the nature of the algorithm employed. In general the static load balancing algorithms are robust and more stable when compared to the dynamic load balancing algorithms.

From the comparison it can be seen that token routing algorithm allows the job migration between resources which in turn increases the resource utilization. But the stability takes a dip due to frequent migration of jobs as this algorithm is dynamic in nature. This job migration is not possible in other algorithms which in turn mean less resource utilization.

4 PROPOSED APPROACH

The proposed load balancing approach is an enhancement made on top of the well known round robin load balancing algorithm. Fig. 4 shows the algorithm for the proposed approach.

Step 1: Initialize all the available Virtual Machines (VMs) to IDLE status and maintain in a VM_STATE list.

Step 2: Create an empty data structure – Hashmap (VM-MAP) without any entries.

Step 3: Receive the requests from the data center.

Step 4: Create a queue which collects the entire requests from the datacenter.

Step 5: Pull each request from the queue one by one.

Step 6: Check if the request can be catered by base location VM and whether any VM is IDLE.

Step 7: If VM found IDLE in base location assign the request to the corresponding VM.

Step 8: If no base location VM is IDLE assign the request to any other VM using Round Robin approach.

Step 9: Update the VM_MAP hashmap and VM_STATE list accordingly.

Step 10: Repeat step 5 to step 9 until all requests in the datacenter queue is processed.

Fig. 4 Proposed Algorithm for Load Balancing

While, the round robin method does not store the state of the resource (VM), the proposed approach stores the current state of the VM. The language used to implement this approach will be Java and the changes will be implemented using CloudSim simulator. The performance improvement of cloud load balancer will be observed based on response time of the request and processing time of the data center.

5 RESULTS

The proposed algorithm has been implemented at task level and the results were studied based on the response time of the VM to which the task was allocated. The simulation scenario comprises of a datacenter which contains 5 virtual machines. Initially, 20 cloudlets are assigned to these virtual machines using the existing Round Robin approach. After observing the response time, the same set of cloudlet is assigned using the proposed algorithm and the response time is observed for the proposed approach.

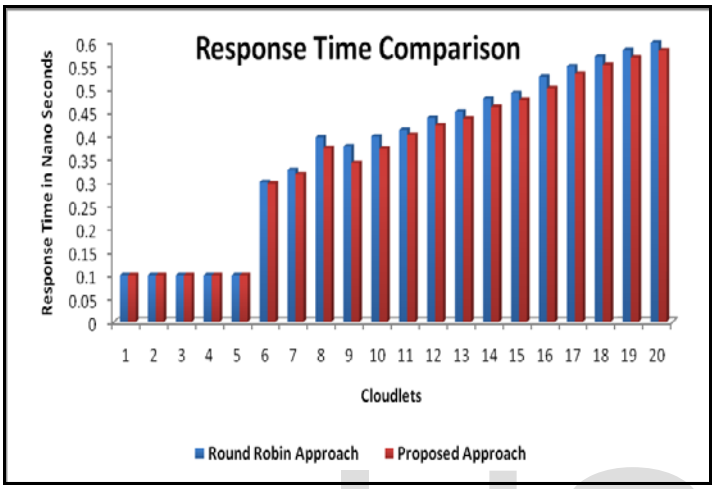

Fig. 5 Response Time Comparison Graph

Table 2 shows the values corresponding to the graph shown in Fig 5.

TABLE II. RESPONSE TIME COMPARISON

Cloudlet ID	Round Robin Approach	Proposed Approach
1	0.1	0.1
2	0.1	0.1
3	0.1	0.1
4	0.1	0.1
5	0.1	0.1
6	0.3	0.297
7	0.326	0.317
8	0.396	0.373
9	0.376	0.341
10	0.398	0.372
11	0.412	0.401
12	0.438	0.422
13	0.451	0.436
14	0.479	0.461
15	0.491	0.476
16	0.526	0.501
17	0.548	0.532
18	0.569	0.551
19	0.583	0.567
20	0.599	0.582

It can be seen that the response time for the proposed approach has decreased when compared to the traditional round robin approach.

6 CONCLUSION AND FUTURE WORK

An analysis of various existing load balancing algorithms available has been performed and a comparative study of these existing algorithms has been presented in this paper. Also, an enhancement to the existing Round Robin algorithm has been proposed. This enhancement maintains the VM state throughout the task assignment phase and hence will provide a significant improvement in the response time of individual requests. A sample set of 5 VMs and 20 cloudlets was considered and the response time comparison for round robin and the proposed approach has been performed. Results indicated that the response time decreased considerably for the proposed approach.

Future work will be directed towards implementing the proposed approach for more cloudlets and VMs and apply the same across multiple datacenters and study the response time behavior.

REFERENCES

[1] Yatendra sahu, M. K. Pateriya "Cloud Computing Overview and load balancing algorithms", International Journal of Computer Application Vol-65 No.24, 2013.

[2] Peter S. Pacheco, "Parallel Programming with MPI", Morgan Kaufmann Publishers Edition 2008.

[3] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid, "Availabity and Load Balancing in Cloud Computing" International Conference on Computer and Software Modeling IPCSIT vol.14 IACSIT Press,Singapore, 2011.

[4] Zhong Xu, Rong Huang,"Performance Study of Load Balanacing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report, 2009.

[5] M.Beltran, A.Guzman and J.L. Bosque, "Dealing withheteroginity in clusters" in proceeding of the Fifth International Symposium on Parallel and Distributed Computing, ISPDC, 2011.

[6] P.Warstein, H.Situ and Z.Huang, "Load balancing in a cluster computer" In proceeding of the seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE, 2010.

[7] Wang, S- C., K-Q. Yan, W- P. Liao and S- S. Wang, "Towards a load balancing in a three- level cloud computing network," in proc. 3rd International Conference on. Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1, pp: 108- 113, July 2010

[8] A. Khiyaita, M. Zbakh, H. El Bakkali, and D. El Kettani, "Load balancing cloud computing: state of art," in Network Security and Systems (JNS2), National Days of, pp. 106–109, IEEE, 2012.