Intelligent Dynamic Load Balancing Approach for Computational Cloud

A. Paulin Florence
Research Scholar
Sathyabama University Chennai, India
Associate Professor, Department of MCA
St. Joseph's College of Engineering, Chennai, India

V. Shanthi, Ph.D
Professor,
Department of MCA
St. Joseph's College of Engineering, Chennai, India

ABSTRACT
Cloud computing is a new technology, which enables provisioning of resources on demand. It caters anything as service. Clients can scale up or scale down their requirements as per their demand. Load balancing is essentially, complex problem in computational cloud. A computational cloud differs from traditional high performance computing systems because of its heterogeneity among the computing nodes. In order to realize the full potential of cloud computing virtualization is widely used. Through virtualization, it is possible to meet the demands from multiple tenants, without switching on many physical nodes. In this paper, we propose Intelligent Dynamic Load Balancing(IDLB) algorithm for computational cloud. IDLB uses cloud machines, when the local processor becomes overloaded. The objective of IDLB is to provide fairness to all the jobs in the cloud by balancing the load between the virtual machines(VM). In this respect our algorithm uses ram size, bandwidth and image size in determining a balance threshold value of each VM for scheduling the jobs. The IDLB distributes the load evenly to all the virtual machines thus it solves the problem of load imbalance and high migration cost incurred by traditional algorithms.

Keywords
Cloud computing; Load balancing; heterogeneity; Virtualization; Virtual machines

1. INTRODUCTION
Cloud computing is the use of computing resources that are delivered as a service over a network. The internet is often represented as a cloud and the term cloud computing arises from that analogy. Cloud computing has started to obtain its significance in corporate data centers as it enables the data center to operate like the Internet through the process of enabling computing resources to be accessed and shared as virtual resources in a secure and scalable manner.

1.1 Characteristics of cloud computing
Cloud computing comprises of five essential characteristics which distinguishes cloud from other computing paradigms. Peter Mell et al., [1] list the essential characteristics of cloud as follows :

- On-demand self-provisioning: It fulfills the needs of customers without need of anyone's help.
- Broad network access: Resources can be availed over the network and accessed through standard mechanisms that promote use by heterogeneous platforms.
- Site autonomous Resource pooling: Pertaining to this various resources such as storage, processing, memory, network bandwidth, and virtual machines can be registered to a cloud provider. These physical as well as virtual resources are dynamically assigned and reassigned according to consumer demand independence of location. Customer generally has no control or knowledge over the exact location of the provided resources.
- Rapid elasticity: Customers need could be served rapidly. When the demand increases resources could be quickly scaled out and at once it is released it could be scaled in.
- Pay as you go model: Resource usage can be monitored, controlled, and reported consumers are charged fees based on their usage of a combination of computing power, bandwidth use and/or storage.

1.2 Service Models
There are different kinds of services provided by the cloud for the cloud users. They are Software as a Service (SaaS) is referred to as "on-demand software", Platform as a Service (PaaS) delivers a computing platform and/or solution stack as a service, Infrastructure as a Service (IaaS) [2] is a model which allows the provider to provision processing, storage, networks, and other fundamental computing resources.

1.2.1. The Deployment Models of the cloud
- Private cloud: It is managed by the organization or a third party and the cloud infrastructure is operated solely for an organization. It may exist on premise or off premise.
- Public cloud: The cloud services are offered over the internet and are made available to the general public or a larger industry group.
- Hybrid cloud: It is the composition of private or public that remains unique entities but is bound together.

2. ABOUT LOAD BALANCING
Cloud is a heterogeneous environment able to meet out the needs of customers on a pay-as-you-go basis. As per the requirements of the customers, resources will be allocated dynamically. Hence it is necessary to balance the workload between the nodes in order to utilize the resources efficiently. Thus load balancing plays a vital role in this environment which attempts to maximize system throughput by keeping all processors busy. Load balancing is done by migrating tasks from overloaded nodes to other lightly loaded nodes to improve the overall system performance. Load balancing is typically based on a load index(i.e length of CPU queue,

speed, memory capacity etc.), which provides a measure of the work load at a node relative to some threshold value which governs the actions taken once a load imbalance is detected. There are two types of load balancing algorithms [4] existing such as:

- Static – Task is always executed on the node to which it is allocated. It is non-preemptive. Work load is distributed to the slave nodes by the master depending upon their performance. Final selection of a host for process allocation is made when the process is created and cannot be changed during process execution to make changes in the system load. It fails to adapt to the dynamic workload
- Dynamic – Work load is distributed among the nodes at runtime. Master allocates the processes dynamically when one of the processors becomes under loaded.

In a cloud environment performance of the system can depend crucially on dividing up work effectively across the participating nodes [10]. Dynamic load balancing has the potential of performing better than static strategies, as they are inevitably more complex. The overheads involved in dynamic LBs are much more, which cannot negate their benefits. Cloud computing is a type of parallel and distributed system. It is a collection of interconnected virtual computers which are to be scaled up or scaled down dynamically (the processors or memory) with the increased demands. Load balancing serves two needs:

- Promotes availability of cloud resources
- Hence promotes performance

In order to balance the load it is necessary to keep the nodes busy, which substantially increases the power consumption. The impact is:

- Increases Total Cost of Ownership(TCO)
- Reduces Return On Investment(ROI)
- Increases Carbon footprints which have a significant impact on the climate change

Hence to attain sustainable growth in cloud computing and to gain the optimum potential of cloud, cloud resources need to be allocated not only to satisfy QOS (Quality Of Service) requirements [5] specified by users via SLA, but also to reduce energy usage. Virtualization plays a key role in this aspect. Virtual machine enables the abstraction of an OS and Application running on it from the hardware.

The rest of the paper is organized as follows: Section 3 discusses about related work, Section 4 discusses about proposed Dynamic Load Balancing Algorithm (IDLB), performance analysis of IDLB in section 5, finally we conclude the paper with summary.

3. Related Work

Brototi Mondal et. al., [11] have developed a soft computing based local optimization load balancing approach. It was used for allocation of incoming jobs to the servers or virtual machines(VMs). Performance of the algorithm was analyzed both qualitatively and quantitatively using Cloud Analyst.

Bin Dong et. al., [12] have presented a dynamic and adaptive load balancing algorithm (SALB) which was totally based on a distributed architecture.

Various most commonly used load balancing algorithms such as throttled, active monitoring load balancing, Round Robin and FCFS are also analyzed which are discussed below:

3.1 Throttled Load Balancer

It maintains a record of the state of each virtual machine (busy/ idle), if a request arrives concerning the allocation of virtual machine, the throttled load balancer sends the ID of idle virtual machine to the data center controller and data center controller allocates the idle virtual machine [8].

3.2 Active Monitoring Load Balancer

Active VM Load Balancer maintains information about each VM and the number of requests currently allocated to the pertaining VM. When a request to allocate a new VM arrives, it identifies the least loaded VM. If there are more than one least loaded VM, the first identified is selected. Active VM Load Balancer returns the VM id to the data center controller. The data center controller sends the request to the VM identified by that id. The data center controller notifies the Active VM Load Balancer of the new allocation.

3.3 Round Robin Load Balancer

In this, the datacenter controller assigns the requests to a list of VMs on a rotating basis. The first request is allocated to a VM picked randomly from the group and then the data center controller assigns the subsequent requests in a circular order. Once the VM is assigned the request, the VM is moved to the end of the list. In this RRLB; there is a better allocation concept known as weighted round robin allocation in which one can assign a weight to each VM so that if one VM is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the data center controller will assign two requests to the powerful VM for each request assigned to a weaker one. This approach does not take focuses on the parameters such as ram size, bandwidth etc. [3].

3.4 FCFS Load Balancer

In this, the datacenter controller assigns the first request to a first idle VM and the next incoming request to the next idle VM and so on. It does not consider the parameters such as Ram, Bandwidth and image size.

4. Intelligent Dynamic Load Balancing Algorithm

Intelligent dynamic load balancer is developed using CloudSim. CloudSim goal is to provide a generalized and extensible simulation framework that enables modeling, simulation, and experimentation of emerging Cloud computing infrastructures and application services. CloudSim is a toolkit (library) for simulation of Cloud computing scenarios. It provides basic classes for describing data centers, virtual machines, applications, users, computational resources, and policies for management of diverse parts of the system (e.g., scheduling and provisioning). These components can be put together for users to evaluate new strategies in utilization of Clouds (policies, scheduling algorithms, mapping and load balancing policies, etc).

The interior hardware infrastructure services interrelated to the cloud is modeled in the simulator by a data center entity for handling service requests. Datacenter is fully automated and offers remote control for the entire infrastructure which makes it ideal for cloud computing and archive storage. The data center entity manages a number of host entities. The hosts are assigned to one or more VMs based on a VM allocation policy. VM policy stands for provisioning of host to a VM, VM creation, VM destruction and VM migration.

For simulation purpose CloudSim is configured hence that there are two data centers under the control of IDLB. Each data center has been configured with two hosts, under each host there are two VMs. In the proposed work all the incoming cloudlets (jobs) are maintained in a cloudlet list and VM's are created and stored in VM list. When a client sends a job, it goes to a dedicated local processor. The jobs are processed in the local processor till the local processor becomes overloaded. Once it becomes overloaded, it switches the job request to the cloud server. When a job is received in cloud server, the cloud simulator creates VM's and cloudlets. VM's and cloudlets are stored in VM list and cloudlet list respectively. These details are submitted to the broker, the broker in turn submits the VM list and cloudlet list to the data center. The data center controller uses IDLB algorithm to determine which VM should be assigned the next request for processing. When one datacenter becomes overloaded, it shifts the load to another datacenter.

The IDLB dynamically allocates the cloudlets to a particular VM determined by VM's Ram, Bandwidth and image size. Once the job is processed, it evaluates the start and finish time of the job. The proposed algorithm solves the problem of load imbalance and high migration cost incurred by traditional algorithms.

4.1 Pseudo Code

```
Input the file size
Computes the length and output size
If job is submitted then
    connects to the local processor
    if  jobsize <=threshold
        Submit the job to the local processor
        Job is processed in local processor
  else
        Send the job to the cloud server
        Activate IDLB
          creates datacenters, Host, VMs and
          broker
           VMs and cloudlets are stored and
           submitted to broker
           Start the simulation
           Allocation of jobs
          Compute debt value of each VM
          Allocates job to a VM depending on
          the parameters
          Repeat job allocation until all the jobs
          are Processed
      Stop the simulation
    End if
```

The architectural view of IDLB algorithm is shown in Figure 1.

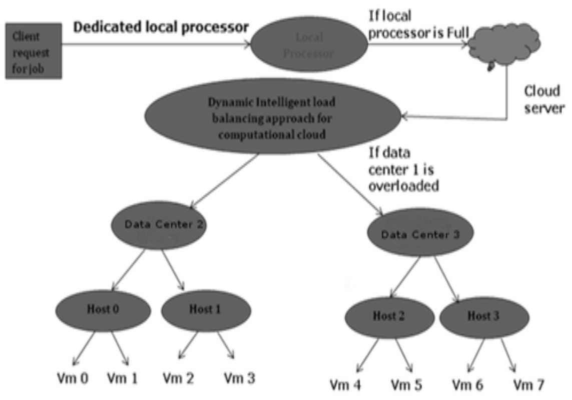

Figure 1. Architectural Diagram of IDLB

5. Simulation results

5.1 VM configuration

Configuration of VMs in data center 1 and data center 2 on behalf of the different parameters is shown in Table 1. The performance measurement of IDLB is based on three metrics: VM's Ram, Bandwidth and image size. The finish time per 1000MIPS of both algorithms are also tabulated. From the table it is observed that IDLB performs well than FCFS.

Table 1. VM configuration

	Data Center 1				Data Center 2			
	HOST 0		HOST 1		HOST 2		HOST 3	
	Vm 0	Vm 1	Vm 2	Vm3	Vm 4	Vm 5	Vm 6	Vm 7
VM RAM	1024	128	2048	256	1024	128	256	2048
Bandwidth	1000	500	1000	500	500	500	300	1000
Image size	500	1000	500	100	500	1000	100	500
Finish Time Per 1000 MI(IDLB)	1.25	2	1	1.75	1.50	2	1.95	1
Finish Time Per 1000 MI(FCFS)	2	2.04	2.1	2.14	2.17	2.26	2.27	2.31

The bar chart shown in Figure 2 explains about the comparison between IDLB and FCFS algorithms. X axis refers to virtual machines and Y axis refers to the processing time. The IDLB algorithm processes the job in a shorter period of time when compared to FCFS algorithm. I

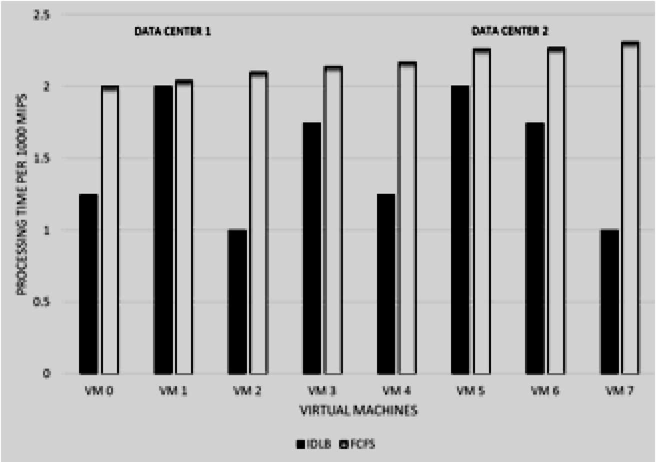

Figure 2. Processing Time

5.2 Allocation of jobs

Graph shown in Figure 3 explains about the allocation of jobs between IDLB and FCFS algorithms. In IDLB, considering the parameters such as VM's ram, bandwidth and image size, jobs are allocated to VMs. In FCFS algorithm, jobs are allocated based on FCFS order without considering any factors such as VM's Ram, bandwidth and image size.

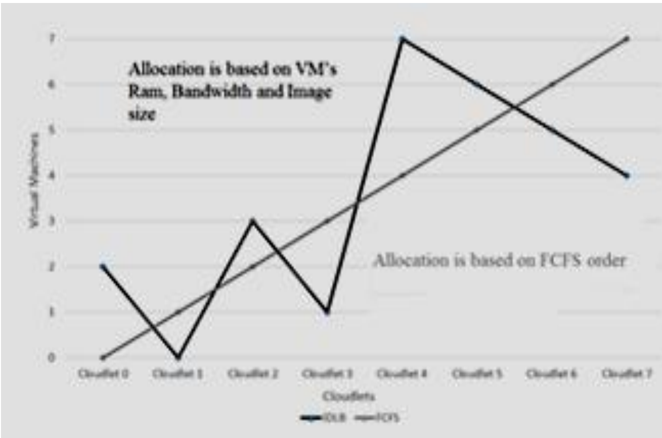

Figure 3. Allocation of jobs

6. CONCLUSION

In this paper, a new dynamic intelligent load balancer is proposed and then implemented in CloudSim, an abstract cloud computing environment using java language. This dynamic load balancer distributes the load evenly to all the virtual machines and also it compares each VM's performance and allocates job to the most powerful VM so that the job is finished in a shorter period of time. This algorithm proves to be more efficient in processing the job, when compared to FCFS algorithm.

7. REFERENCES

[1] Peter Mell, and Tim Grance, 2009. The NIST Definition of Cloud Computing. Information Technology Laboratory.

[2] Hayes, B. 2008. Cloud Computing. Communications of the ACM, 51, pp. 9 – 11.

[3] Qi Zhang, Lu Cheng, Raouf Boutaba, 2010. Cloud computing: state-of-art and research challenges. The Brazillian Computer Society.

[4] Abbas Karimi, Faraneh Zarafshan, Adznan, b. Jantan, "A new fuzzy approach for dynamic load balancing algorithm", International Journal of Computer Science and Information Security(IJCSIS), 2009, 6(1).

[5] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., Brandic, I., "Cloud Computing and emerging IT platforms: Vision, type, type, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, 25,(6), pp.599-616c.

[6] Ram Prasad Padhy, and Goutam Prasad Rao, P., Load balancing in cloud computing system. Proc. Department of Computer Science and Engineering National Institute of Technology, Rourkela Rourkela, Orissa, India, 2011, pp. 252-312.

[7] Rodrigo, N., Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar, A. F., De Rose, and Rajkumar Buyya, 2010. CloudSim: A Toolkit for Modeling and simulation of Cloud Computing environments and evaluation of resource provisioning algorithms.

[8] Meenakshi Sharma, Pankaj Sharma, Dr. Sandeep Sharma, Efficient load balancing algorithm in VM Cloud environment, 2012, IJCST, 3(1).

[9] Anthony, T.Velte., Toby, J.Velte., Robert Elsenpeter, Cloud Computing A Practical Approach. TATA McGRAW-HILL. First Edition.

[10] Branko Radojevic, Mario Zagar, 2011. Analysis of issues with Load balancing algorithms in Hosted(Cloud) environments. MIPRO.

[11] Brototi Mondal, Kousik Dasgupta, Paramartha Dutta, Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach. Procedia Technology, vol. 4, pp.783-789, 2012.

[12] Bin Dong, Xiuqiao Li, Qimeng Wu, Limin Xiao, Li Ruan, "A dynamic and adaptive load balancing strategy for parallel file system with large-scale I/O servers, Journal of Parallel Distribution Computing", vol. 72, pp.1254-1268, 2012.

[13] www.cloudbus.org

[14] Zhu Y, Hu Y, "Efficient, proximity-aware load balancing for DHT-based P2Psystems", IEEE Transactions on Parallel and Distributed Systems, vol.16, no.4, pp.349–61, 2005.

[15] Karger D, Ruhl M., "Simple efficient load-balancing algorithms for peer-to-peer systems", Theory of Computing Systems, vol.39, no.6, pp.787-804, 2006.