Implementing a Fault Tolerance Enabled Load Balancing Algorithm in the Cloud Computing Environment

G.Gayathri[1], R.Latha[2]
[1]Research Scholar, [2]Professor/Head
Dept. of MCA,
St.Peter's University, Chennai, TamilNadu, India

Abstract - The development of cloud computing supported virtualization technologies brings enormous opportunities to host virtual resource at low price without owning any infrastructure. Virtualization technologies help users to adopt setup and be charged on pay-per-use basis. However, Cloud data centers largely comprise heterogeneous servers hosting multiple virtual machines (VMs) with potential varied specifications and unsteady resource usages, which can cause unbalanced resource utilization among servers which may result in performance degradation and service level agreements (SLAs) violations. To attain effective scheduling, these challenges ought to be self-managed and resolved by load balancing schemes. This paper proposes a load balancing algorithm which balances the load on the host as well as preserves the fault tolerance level of the system based on virtual machine migration.

Keywords **- cloud computing, Load balancing, fault tolerance, performance, SLA, VM, Scheduling.**

I. INTRODUCTION

Cloud computing is a computing paradigm that centers on leveraging a varied range of users with access to scalable, virtualized resources over the internet. Cloud computing is a model to extract and manage IT resources. In data centers applications are run in physical servers that are often provisioned with high workload. This configuration makes data centers expensive to maintain with significant overhead. Data centers are more flexible, secure and provide good support for on-demand resource allocations. It abstracts server diversity, performs server consolidation and enhance server utilization. A host can run multiple VMs with different resource specifications and varied workloads. Physical servers hosting inhomogeneous VMs with unpredictable workloads may cause a imbalance in resource usage which in turn results in performance deterioration and service level agreements (SLA) violation.

Cloud data centers are highly unpredictable due to 1) irregular consumer request pattern 2) wavering resource usage by VMs 3) diverse rates of arrivals and departure of consumers and 4) the performance of the host may vary when handling different load levels. These reasons are enough to trigger unbalanced loads in the data center, which requires load balancing mechanism to avoid performance degradation and service level agreement violations.

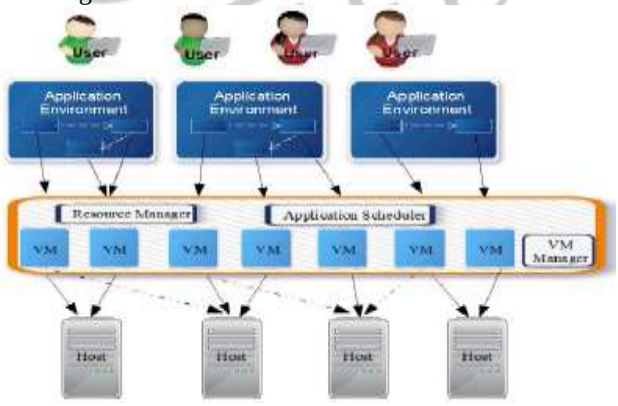

Figure 1. Datacenter Model

LOAD BALANCING

Cloud load balancing is a mechanism which distributes the dynamic workload evenly across all the host in the data center. Load balancing schemes are applied to achieve user satisfaction and better resource utilization. It ensures that no host is overloaded and thus improves the overall system performance. The main aim of load balancing is to assign VMs to appropriate host and balance the resource usage among all of the hosts. Efficient load balancing algorithms helps in optimal utilization of available resources and helps in implementing failover, enabling scalability, avoiding bottlenecks and reducing response time. Figure 1. Shows the

application, VM and host representation in cloud data centers. The host comprises of resources like CPU, memory and storage. Above the host the server platform manages the VMs run by the hosts. The applications are executed inside the VMs and have predefined dependencies between them.

FAULT TOLERANCE

As the cloud computing systems continue to grow in scale and complexity, it is important to ensure the stability, availability, and reliability in such systems. The diverse execution environments, addition and removal of components, intensive workload on servers, are the primary reasons that can produce failures in dynamic environments of cloud computing. The reliability of such systems can be easily endangered if the proactive measures are not taken against the possible failures in cloud systems. To promise reliability and availability of services running in the virtual machines a fault tolerant enabled load balancing algorithm is implemented in this paper. At some time there may be many customers to register the data center or free their resources to exit from the data center. So, there would be unbalanced loads among the hosts in the data center. To reach load balancing, some VMs should be migrated from the high-load hosts to the low-load hosts. The fault tolerance level of the service is guaranteed by distributing the VMs of the service onto various physical hosts. Fault-tolerant level can be described as: if service *i* can work normally when *ki* hosts break down, the fault-tolerant level of service *i* is defined as *ki* . To provide reliable services, the fault-tolerant level should be ensured while VMs are migrated to balance loads.

Before getting into the implementation details of the load balancing algorithm we first discuss the challenges and the related technologies of a good load balancing algorithm. The challenges of load balancing algorithms for VM placement on host depends on the following factors:

Overhead: It is the cost due to VM migration cost as well as communication. A load balancing algorithm should be designed in a way it reduces overhead suffered by the system. .

Performance: It marks the efficiency of the system. Performance can be measured from user satisfaction. Load balancing plays a vital role in improving performance of the system. It can be realised by following factors.

- ***Resource Utilization***
- ***Scalability***
- ***Response Time***

RELATED TECHNOLOGY

Before discussing the VM load balancing algorithms, some related technologies for load balancing is introduced.

Virtualization technology: Virtualization is a technique to divide single physical infrastructure in to various dedicated small machines. These machines are named as Virtual Machines (VMs). Virtualization makes it possible to run multiple applications and its related operating system on the same server at the same time. It capacitates businesses to reduce IT cost and increases the efficiency, utilization and flexibility of their existing computing hardware.

VM Migration: Live migration of virtual machines [1] is the process of moving a running virtual machine between different physical machines without suspending the application. Memory, storage and network details are moved to the destination machine from source without disturbing the users interaction. Nowadays the live migration technology is widely adopted in the current cloud computing data centers.

II. VM LOAD BALANCING ALGORITHM MODELING IN CLOUDS

This section, discusses the details of designing of VM load balancing algorithm. The algorithm should decide about the VM resource type, VM type, VM allocation, optimization strategy and scheduling process.

VM Resource Type: When designing load balancing algorithm based on live VM migration, the resource type should be decided. It refers to the resources held by the Virtual machines. Usually the VM resource utilization is marked by its CPU utilization. Some algorithms consider more than one resources such as bandwidth and memory. The proposed work in this paper takes into account only the CPU utilization in to account. The selection and migration of a particular VM is decided by CPU utilization of the host.

VM Type: Assumptions could be made for VM type as homogeneous or heterogeneous when scheduling VMs for load balancing. Many algorithms assume a homogenous VM type for simplicity. But in real time VMs with same characteristics cannot be expected. The proposed work in this paper uses heterogeneous VM Type to take full advantage of the heterogeneous nature of cloud resource.

VM Allocation Dynamicity

In load balancing algorithm VM allocation can be classified as static and dynamic categories. Static algorithms are offline algorithms that requires the information of VMs to be allocated to host is known in advance.. But in actual cloud implementation the demands will change over the time. These conditions require algorithms which allocate VMs according to the loads at each time interval. Such types of algorithms are capable of configuring the VM placement phase with VM Migration technique. The proposed algorithm proactively schedules the incoming request at the VM allocation stage by a Modified Round Robin (MRR) scheme which effectively avoids the fault tolerance degradation at the initial stage itself.

Scheduling Process Modelling
The load balancing scheduling is mainly divided into scheduling at the,

i) VM initial placement stage.
ii) VM live migration stage.

VM initial placement stage
At the VM initial placement stage, the important component of the scheduling process is the VM acceptance policy, which decides the initial VM to host placement. The policy takes the host remaining resource into consideration.

VM live migration stage
As for the live migration stage in scheduling process, it mainly considers following aspects:

(1). A VM migration policy dictates when to trigger a VM migration from one source to destination host. Generally, a VM migration policy adopts a threshold mechanism to trigger migration operations, and the threshold value is fixed by a data center administrator based on the computing capabilities of each host. For example, if the CPU utilization of a particular host exceeds the threshold level then the migration policy will be triggered. The resource utilization data is tracked to take necessary steps. In this proposed work, as the algorithm assumes single resource type (CPU Utilization) the hosts are grouped under three categories like HEAVY, MODERATE and LIGHT based on the CPU usage. If any hosts slips into HEAVY category, then scheduling process will trigger the migration of VM.

(2) VM selection policies establish polices to choose VMs to be migrated from overloaded hosts. The host which comes under HEAVY category has to be scanned and one or more of the VMs should be selected so that the host's CPU utilization comes below the threshold level.

(3) A VM acceptance policy concentrates on 1) the remaining resource of hosts, 2) an associated resource type (CPU usage). VM acceptance policies are executed by load balancing algorithms to determine whether to bid hosting for a given VM or not.

RELATED WORK

There have been many studies on load balancing based on virtualization technology. Based on VM migration, Wilcox (2009) proposed a load balancing scheme named modified central scheduler load balancing (MCSLB), which migrates VMs from the heaviest load host to the lightest host. Wang et al., (2010) proposed a scheduling algorithm to utilize better executing efficiency and maintain the load balancing of the system by combining opportunistic load balancing and load balance min– min scheduling algorithms. Zhang et al., (2010) presented a load balancing mechanism based on ant colony and complex network theory in the open cloud computing federation to realize load balancing in the distributed system.

In order to achieve the best load balancing and reduce or avoid dynamic migration, Hu et al., (2010) proposed a scheduling strategy on load balancing of VM resources based on a genetic algorithm with the consideration of system variation and historical data. Randles et al., (2010) investigated three possible distributed load balancing algorithms inspired by Honeybee Foraging Behavior, Biased Random Sampling and Active Clustering for cloud computing scenarios. By integrating live OS migration into the Xen VM monitor, Clark et al., (2005) discussed the procedure of live VM migration, and presented the design, implementation and evaluation of high-performance OS migration built on top of the Xen VMM. Lin Yao et al., (2013) proposed a novel guaranteeing fault-tolerant requirement load balancing scheme (GFTLBS). GFTLBS migrates the VMs to balance the load without violating the fault-tolerant requirement of all services.

Owing to hardware failure, communication link errors and malicious attack, fault tolerance is one of the most critical issues in distributed systems. However, most of the researches of load balancing discussed above do not take the fault tolerance of the services into account except the GFTLBS by Lin Yao et al., (2013). Different VM migration schemes result in various influences on the fault-tolerant level of the services.

In the cloud computing environment, the data center has many VMs to run the service, and the fault tolerant level of the service is guaranteed by distributing the VMs of the service onto various physical hosts. Fault-tolerant level can be described as: if service i can work normally when ki hosts break down, the fault-tolerant level of service i is defined as ki.[11]. To provide reliable services, the fault-tolerant level should be ensured while VMs are migrated to balance loads.

To guarantee the fault-tolerant level of all services provided by the data center while balancing the load based on VM migration among the hosts a Fault Tolerance enabled load balancing algorithm based on VM migration, is proposed in this paper.

III. LOAD BALANCING WITH FAULT TOLERANCE

The work in this paper focuses on maintaining the fault tolerance level of the overall system while load balancing. As discussed earlier in this paper, the load balancing comprise of two phases they are, scheduling at the VM initial placement stage and scheduling at the VM live migration stage.

Usually in initial placement stage of VMs, the well known Round Robin algorithm is used to place the VMs in the host. In basic round robin algorithm the Load balancer allocates a VM to the requesting node in cyclic manner equally among all available nodes. The advantage of this algorithm is that it utilizes the CPU resources in a balanced order and nearly equal numbers of VMs are assigned to all the nodes which ensure fairness. Even though the round robin algorithm allocates VMs equally to the available hosts, it does not check whether the VM of same service is already running in the host. If the VM of the same service is allocated to the same host then the fault tolerance level of that particular service is decreased. To avoid this situation a modified round robin algorithm is proposed.

MODIFIED ROUND ROBIN ALGORITHM

The modified Round Robin algorithm allocates VM to the host in a cyclic manner but before allocating the VM it checks whether the same service type is already running in the host. If so it skips the current host and selects the next host for allocating the VM. The Modified Round Robin Algorithm maintains the fault tolerance level of the services at the initial allocation stage itself.

LOAD DETECTION

After the initial allocation the system is monitored at regular interval of time. The basic method of cloud computing dynamic resources scheduling is migrating virtual machine among hosts in the data center, so that every host can run at the optimal load states. If the load of a host is too heavy, it will violate the SLA service level agreement, which affects the Quality of Service (QoS) of the executing services. On the contrary, if the load of a host is too low, resources utilization of the host will be very low. Then the overhead due to these hosts is high. Therefore, the resource utilization thresholds of a host need to be monitored and recorded in the load balancing algorithm. The load of the host is measured in terms of its CPU usage. If any host which exceeds the CPU maximum threshold then the host is marked as “HEAVY”. If any of the host is running below the min CPU threshold then the host is marked as “LIGHT”. If any of the host is running between 40 to 55 % CPU usage then the host is marked as “MODERATE”. If any of the host is found to be “HEAVY” it is said to be overloaded so some or any of the VMs had to be removed to bring down the CPU Usage below the maximum threshold. This host is the source host for the migration.

Load Balancing based on VM Migration

If an host is overloaded then the VMs running inside the host should be migrated to the lighter host. To choose the VM to be migrated, all the VMs running in the source host is sorted in the descending order and the hosts which are marked as "LIGHT" are sorted in ascending order. Then the VM to be migrated in the source host must be chosen such that the destination host should not already run same type of VM itself (no two services of same type in one host) and after migration the CPU usage of the destination host should not exceed the CPU maximum threshold. If no migration is possible among the LIGHT group then the same procedure is repeated for MODERATE group.

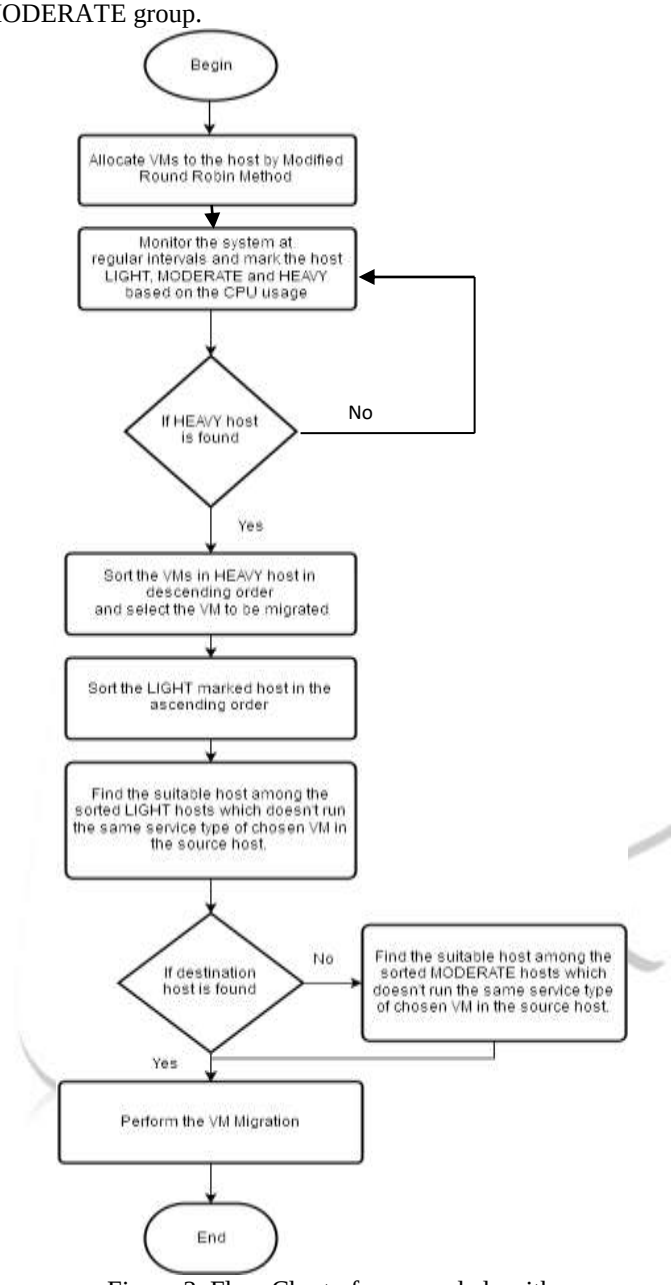

Figure 2. Flow Chart of proposed algorithm

IV. EXPERIMENT AND EVALUATION

CloudSim is simulation software of cloud computing, designed and implemented by The University of Melbourne, Australia. CloudSim has two new advantages: (1) modelling and simulation of large-scale cloud computing infrastructure; (2) a self-sufficient supporting Data Center, Data Center Broker, scheduling and allocation strategy, and it employs virtualization to provide resource. The resource of a host in a data center, such as physical host, could be mapped to several VMs based on the user requirement, thus there are possible different numbers of VMs running on the hosts. Therefore, VM migration is necessary to balance the load. The proposed algorithm is simulated and tested under CloudSim Environment.

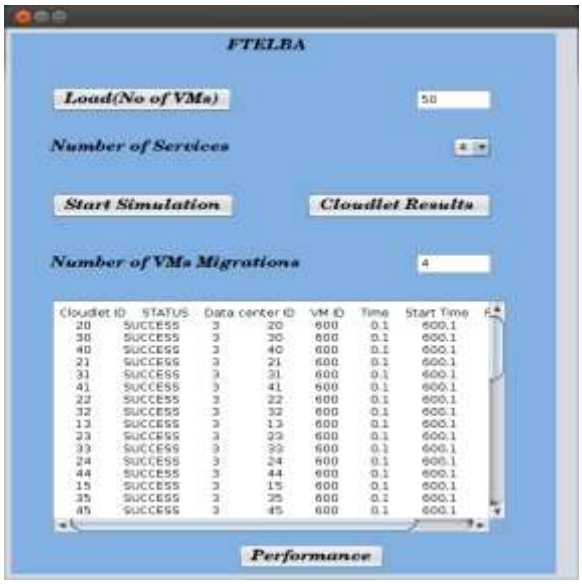

Figure 3. Simulation in CloudSim Environment

The Simulated results are as follows. The proposed algorithm is compared with GFTLBS. The execution time of the proposed algorithm is found to be less than GFTLBS. The proposed algorithm shows better results for SLA Violations, Throughput, and number of migrations.

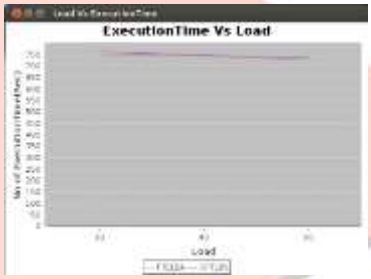

Figure 4. Comparison of Execution Time

Figure 5.Comparison of SLA Violations

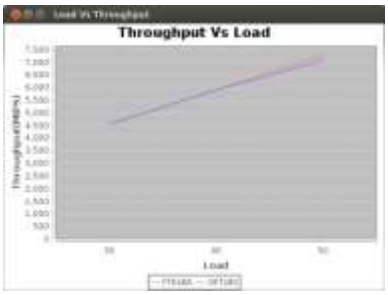

Figure 6. Comparison of Throughput

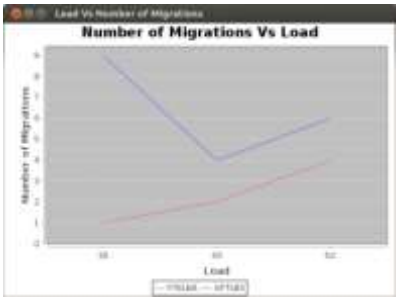

Figure 7. Comparison of Number of migrations

In order to show that the proposed algorithm achieves better fault tolerance it is evaluated in the CloudSim environment. The objective of the proposed work is to preserve the fault tolerance level of the services when the migration is carried out, i.e., no host should be allocated the VM with the same service type. This criterion ensures that if a host fails for some reason then the VMs running inside the hosts will be lost, in that case not more than one instance of any VM is lost. This helps to keep the fault tolerance level of the system. The comparison of the fault tolerance level of the system based on the proposed work and the existing GFTLBS can be seen in Figure 8. The proposed algorithm is found to give better fault tolerance level than the existing scheme.

Figure 8. Comparison of Fault Tolerance Level

V. CONCLUSION

In this paper an algorithm for fault tolerance enabled load balancing in the cloud environment is detailed. This algorithm is devised to maintain the fault tolerance level of the services running inside the VMs. VMs can be migrated from the overloaded host to the lightest one while preserving the fault-tolerant requirements of all the services. In addition, based on VM migration, the hardware utilization, power savings, availability, security and scalability can be increased without disturbing the customer applications running in the VMs. The simulation indicates that the algorithm works well and keeps stability in guaranteeing the fault-tolerant requirements of all services while balancing the load by VM migration.

References

1. Clark, C., Fraser, K., Hand, S., Hansen, J., Jul, E., Limpach C., Pratt, I. and Warfield, A. (2005) Live Migration of Virtual Machines. *Proc. 2nd Conf. Symp. Networked Systems Design & Implementation-Volume 2 (NSDI'05)*, Berkeley, CA, pp. 273– 286. USENIX Association, Berkeley, CA.

2. Daniels J. Server virtualization architecture and implementation. Crossroads 2009; 16(1):8–12.[2] Speitkamp B, Bichler M. A mathematical programming approach for server consolidation problems in virtualized data centers. IEEE Transactions on services computing 2010; 3(4):266–278.
3. Deng, J., Qiu, M. and Wu, G. (2010) Fault Tolerant Data Collection in Heterogeneous Intelligent Monitoring Networks. *2010 IEEE 5th Int. Conf. Networking, Architecture and Storage (NAS)*, Macau, China, July 15–17, pp. 13–18. IEEE, NewYork.
4. Gutierrez-Garcia JO, Ramirez-Nafarrate A. Agent-based load balancing in cloud data centers. Cluster Computing 2015; 18(3):104
5. Hu, J., Gu, J., Sun, G. and Zhao, T. (2010) A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment. *2010 3rd Int. Symp. Parallel Architectures, Algorithms and Programming (PAAP)*, Dalian, China, December 18–20, pp. 89–96. IEEE, NewYork.

6. Lin Yao, Guowei Wu, Jiankang Ren, Yanwei Zhu and Ying Li(2013) Guaranteeing Fault-Tolerant Requirement Load Balancing Scheme Based on VM Migration
7. Meng, X., Pappas, V. and Zhang, L. (2010)Improving the Scalability of Data Center Networks with Traffic-Aware Virtual Machine Placement. *2010 Proc. IEEE INFOCOM*, San Diego,CA, March 15–19, pp. 1–9. IEEE, NewYork.

8. Qiu, M., Liu, J., Li, J., Fei, Z., Ming, Z. and Sha, E. (2011) A Novel Energy-Aware Fault Tolerance Mechanism for Wireless Sensor Networks. *2011 IEEE/ACM Int. Conf. Green Computing and Communications (GreenCom)*, Chengdu, China,August 4–5, pp. 56–61. IEEE, NewYork.

9. Randles, M., Lamb, D. and Taleb-Bendiab, A. (2010) A Comparative Study into Distributed Load Balancing Algorithmsfor Cloud Computing. *2010 IEEE 24th Int. Conf. Advanced Information Networking and Applications Workshops (WAINA)*,Perth, Australia, April 20–23, pp. 551–556. IEEE, NewYork.

10 Soumya Ray and Ajanta De Sarkar(2012) Execution analysis of load balancing algorithms in cloud computing environment.
11. Wang, S.,Yan, K., Liao,W. andWang, S. (2010) Towards a Load Balancing in a Three-level Cloud Computing Network. 2010 3rdIEEE Int. Conf. Computer Science and Information Technology (ICCSIT), Chengdu, China, July 9–11, pp. 108–113. IEEE,NewYork.
12. Wilcox Jr, T.C. (2009) Dynamic load balancing of Virtual machines hosted on Xen. Master's Thesis, Brigham Young University, USA.
13. Zhang, Z. and Zhang, X. (2010) A Load Balancing MechanismBased on Ant Colony and Complex Network Theory in OpenCloud Computing Federation. 2010 2nd Int. Conf. Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 30–31, pp. 240–243. IEEE, NewYork.

14. Zhu, X., Qin, X. and Qiu, M. (2011) Qos-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters. *IEEE Trans. Comput.*, **60**, 800–812.