Secure Resource Aware Load Balancing through VM Migration in Cloud Data Center

Riddhi Trivedi[1] Mr. Miren Karamta[2] Mr. Hardik Upadhyay[3] Dr. M. B. Potdar[4]

[1]*Student, GTU PG School-Ahmedabad, Gujarat, India*
[2] *Project Scientist, Bhaskaracharya Institute for Space Applications and Geo-Informatics, Gandhinagar, Gujarat, India*
[3]*Assistant Professor, Gujarat Power Engineering & Research Institute, Mehsana, Gujarat, India*
[4]*Project Director, Bhaskaracharya Institute for Space Applications and Geo-Informatics, Gandhinagar, Gujarat, India*

ABSTRACT

Improvement in cloud networking, cloud infrastructure and processing data demands for flexibility of resource management. For maintaining the high availability of cloud data center, fault tolerance systems should be the main requirements which include the effective load balancing and migration of applications by not interrupting any other services running. The limitation of the available Virtual Machine Load balancing policies for the cloud is to gain resource utilization ratio by safeguarding an efficient and upright allocation of computer resources. VM Migration allows virtual machines to be migrated from one host machine to another host machine without interfering with software or processes running on those virtual machines. During this VM migration process, the data which is migrated over network becomes vulnerable and confidentiality of user can be breached. Therefore, it is required to develop an efficient solution to migrate data securely over the cloud. The proposed framework is capable to balance the load with the help of round-robin algorithm using it by HAProxy load balancer and securely migrating the application requests if any of the load balancer server is overloaded or would not be able to take requests from the clients. The proposed model also has Ganglia Monitoring implemented which can intelligently monitor all the tasks which are performed by all the servers located in the same cluster and shows it at the main server where the gmetad services is implemented and it shows all the data which are collected from the nodes located in the cluster. The target of achieving fault-tolerance cluster and highly available cluster is being fulfilled by this proposed framework.

Keyword: - *High Availability, Fault Tolerance, Ganglia Monitoring, Migration, Load Balancing*

1. Introduction

Cloud computing is an evolving computing paradigm that has influenced every other entity in the globalized industry, whether it is in the public sector or the private sector. Considering the growing importance of cloud, finding new ways to improve cloud services is an area of concern and research focus. Flexibility of resource management is being demanded for the improvement in the networking o cloud, infrastructure and processing of data. For maintaining the high availability of cloud data center, fault tolerance systems should be the main requirements which include the effective load balancing and migration of applications by not interrupting any other services running.

For effective load balancing and migration process we are installing Ganglia Monitoring System for monitoring the cloud data center which is scalable and allow us to view variety of system metrics of Linux servers and clusters in real time. Ganglia allow us to set up grids (locations) and clusters (groups of servers) for better management. Thus, you can create a grid composed of all the machines in a remote environment, and then group those machines into smaller sets based on other criteria. In the back end ganglia is made up of four components like gmond, gmetad, RRD and PHP web front-end.

Every node which you want monitored has gmond installed. Every server node uses gmond to send the data to the master node running gmetad. Gmetad collects the data of all the nodes and send it to the RRD tool to be stored. The stored data can be viewed by the web browser with the help of PHP web front-end.

The data which we can view by web front-end is being organized on several levels. Ganglia itself organizes nodes which can be individually monitored into clusters which are groups of similar nodes. Collection of clusters also can be organized on a higher level into grids.

2. Problem Statement

2.1 Problem Identified

Load reconciliation and VM migration with cloud information centers are a unit such a big amount of issues coming back within the means, very less analysis work is completed with this and ideas projected with the Ganglia Monitoring System enforced with Load Balancer.

However, at the current stage this analysis remains suppositious with solely suppositious models and systems being projected. As we discover there are some drawbacks in current state of affairs that is enforced and also the analysis that is completed until the date during this domain. Conjointly the security concern is high as a result of we'll use the live migration technique for migrating the applications of the VM.

We will concentrate on the load balancer technique and therefore the Ganglia Monitoring System which may work along and therefore the Ganglia monitoring system that is analysing the load balancer performance and checking each ten seconds is in a position to give notice antecedently concerning the alert for migrating the VM applications for continuity and high handiness of the applying running on the VM.

Focus is on the load balancer technique and the Ganglia Monitoring System which can work together and the Ganglia monitoring system which is analysing the load balancer performance and checking every 10 seconds is able to notify previously about the alert for migrating the VM applications for continuity and high availability of the application running on the VM.

2.2 Problem Statement

To design and develop a new framework for fault tolerance cluster in order to mitigate the high availability issue and load balancing issue in cloud data center through monitoring the cluster nodes in cloud data center by Ganglia Monitoring and migrating the application data to less loaded node if any load balancer server is fully loaded. The load balancing is performed by Round Robin algorithm and HAProxy. Hence, my objective can be stated as follow "Secure Resource Aware Load Balancing through VM Migration in Cloud Data Center".

2.3 Existing System

High availability is a performance of system that permits associate the application to mechanically restart or reroute work to a different capable system within the event of a failure. In terms of servers, there square measure a couple of completely different technologies required to line up an extremely obtained system. There should be a part which will redirect a work which is called as a migration and there should be a mechanism to observe for failure and transition the system if disruption is detected which is any monitoring system.

The keepalived daemon may be accustomed monitor services or systems and to mechanically failover to a standby if issues occur. The keepalived is used to set up high availability server for the software load balancers. There must be floating IP address which can be moved between multiple capable load balancers. All the set up can be configured to

divide the traffic between multiple backend web servers. If the first load balancer goes down, the floating information processing are affected to the second load balancer mechanically, permitting service to resume.

A Floating IP Address is associate IP address which will be instantly transfer from one drop to a different drop within the same data center.

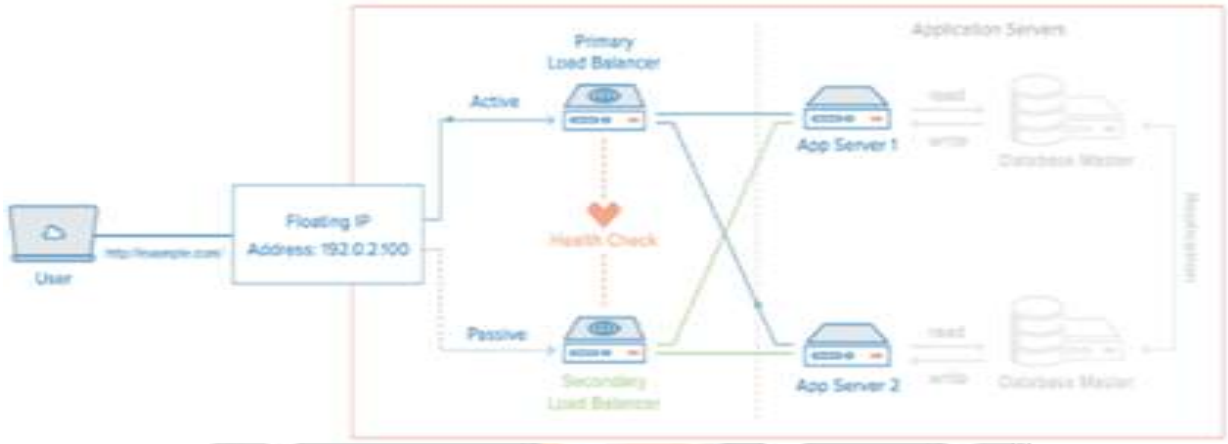

Figure 2.3.1 Active Primary Load Balancer[26]

As shown in the Figure 2.3.1, the user request goes to the web server which has floating IP Address. The Web server is connected with the primary and secondary load balancers. And the both primary and secondary load balancers are connected through Health Check. The health check between two load balancers checks if both the load balancers are alive or not. Here the figure shows that the primary load balancer is alive and can take the traffic of the main web server and proceed the request via primary load balancer and the application server 1 with database master in which we can read and write the data.

Figure 2.3.2 Active Secondary Load Balancer[26]

As shown in Figure 2.3.2, the primary load balancer goes down because of the overloading of traffic so the traffic passes to the Secondary load balancer which is active and connected to the primary load balancer via Health check. The secondary load balancer is alive and can take traffic from the main web server and the traffic passes to the secondary load balancer which is connected with application server 2 with master database through which we can read and write the data same as the Primary load balancer.

Here both the databases of the application servers are connected with each other with the replication process and if any database goes down than the other database can receive the read-write request and can proceed to that requests also.

3. Proposed Solution

Geospatial data involves the things which are related to space and time also latitude and longitude coordinates. This data is collected from the satellite imaginary and remote sensing. The increasing data rate hints upon the challenges of analyzing, managing, storing, processing and visualized the large amount of data. Geospatial data collection has been shifting from a data sparse to a data rich paradigm. [] the data sets are categorized as vector data and raster data.

3.1 Proposed Framework

Proposed structure imagined as a key segment. It is an end to end fault tolerance and highly available system that does the load balancing and migration of the application if any node is under highly loaded condition. Figure demonstrates the proposed framework, which consist three server load balancers segments, three applications servers with replication database and one main server with the floating IP.

Different Phases of Proposed Architecture:

- Main Server: The main application server would gather all the user/client requests from the internet. The IP of this machine would work as a floating IP which is passing all the client request to the Load Balancer Servers.
- All Load Balancer Servers: The Load Balancer Server accept the client request from the main server and handles the request according to its priority. It handles the applications according to the availability of resources in Load Balancer Servers. All the Load Balancer Servers have the HAProxy implemented which balances the load according to its priority.
- All Application Server: The Application Servers accept the requests for application installed on the machine with replicated database. The accepted request is being processed in all Application Servers and the reply is generated from the Application Servers and send back to the main server through Load Balancer Servers.

Now as shown in Proposed Model, we have used three load balancers instead of two to make it highly available and fault tolerance which help us to manage load in a smooth manner. High Load Balancer is used to balancer very high and very low load categories. Medium Load Balancer is used to balance medium or average kind of load categories and at last low load balancer is used to balancer normal high and normal low load categories.

The Ganglia Monitoring System is being implemented in Main Web server which has the floating IP address and through Ganglia Monitoring System we can get the updates of the server for every 10 seconds. Ganglia Monitoring System monitors every cluster and grids. It will generate the report of each and every node of the cluster and the report will be generated in XML format and the admin can see that report and make changes in any system or server if required.

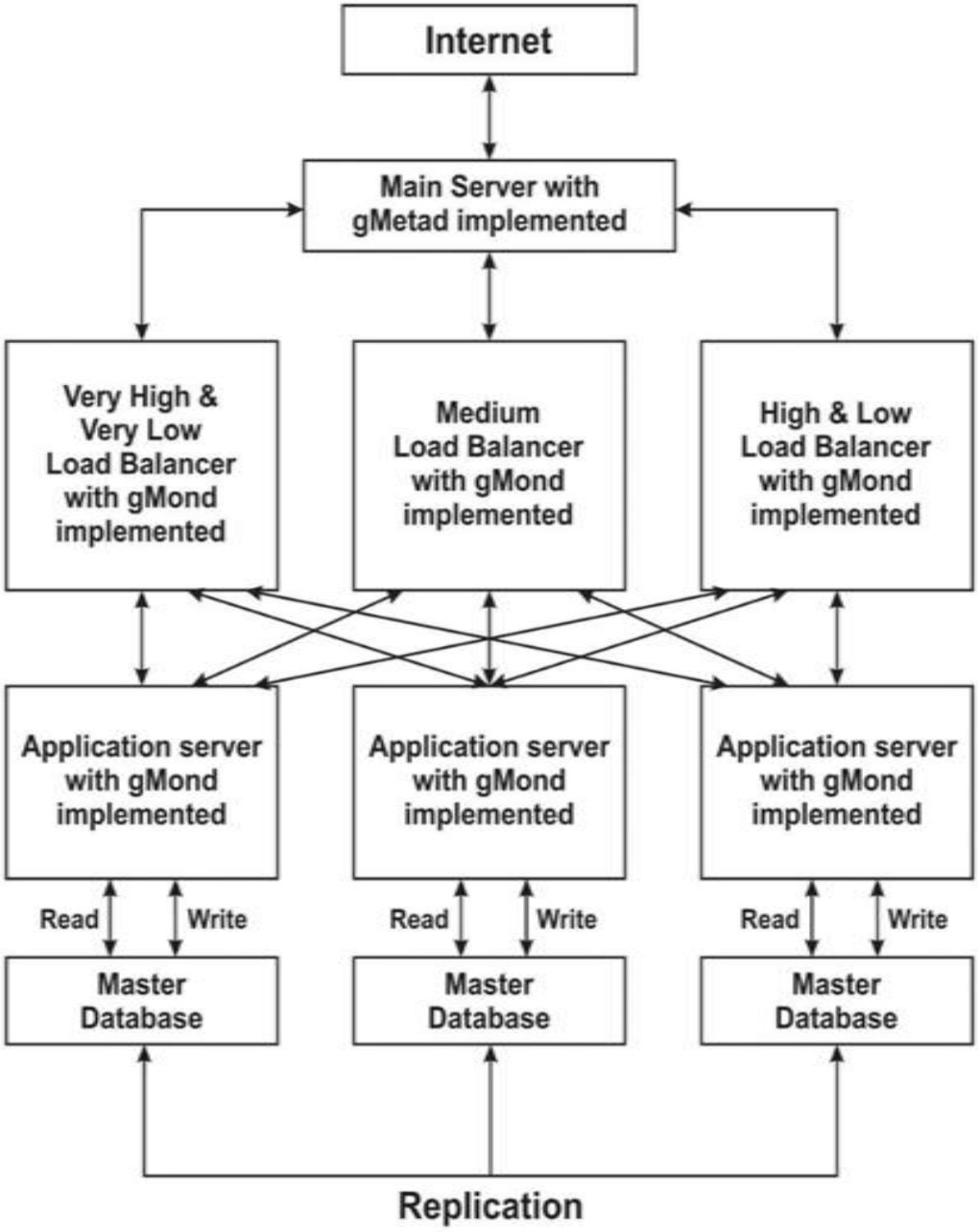

Figure 3.1.1 Proposed Framework

If any system is overloaded or near to overload than Ganglia Monitoring System is give the notification for that and admin can provide some protection over the problem or can switch the traffic between multiple web servers so that the problem does not occur and the availability of the service does not break.

3.2 Proposed Workflow

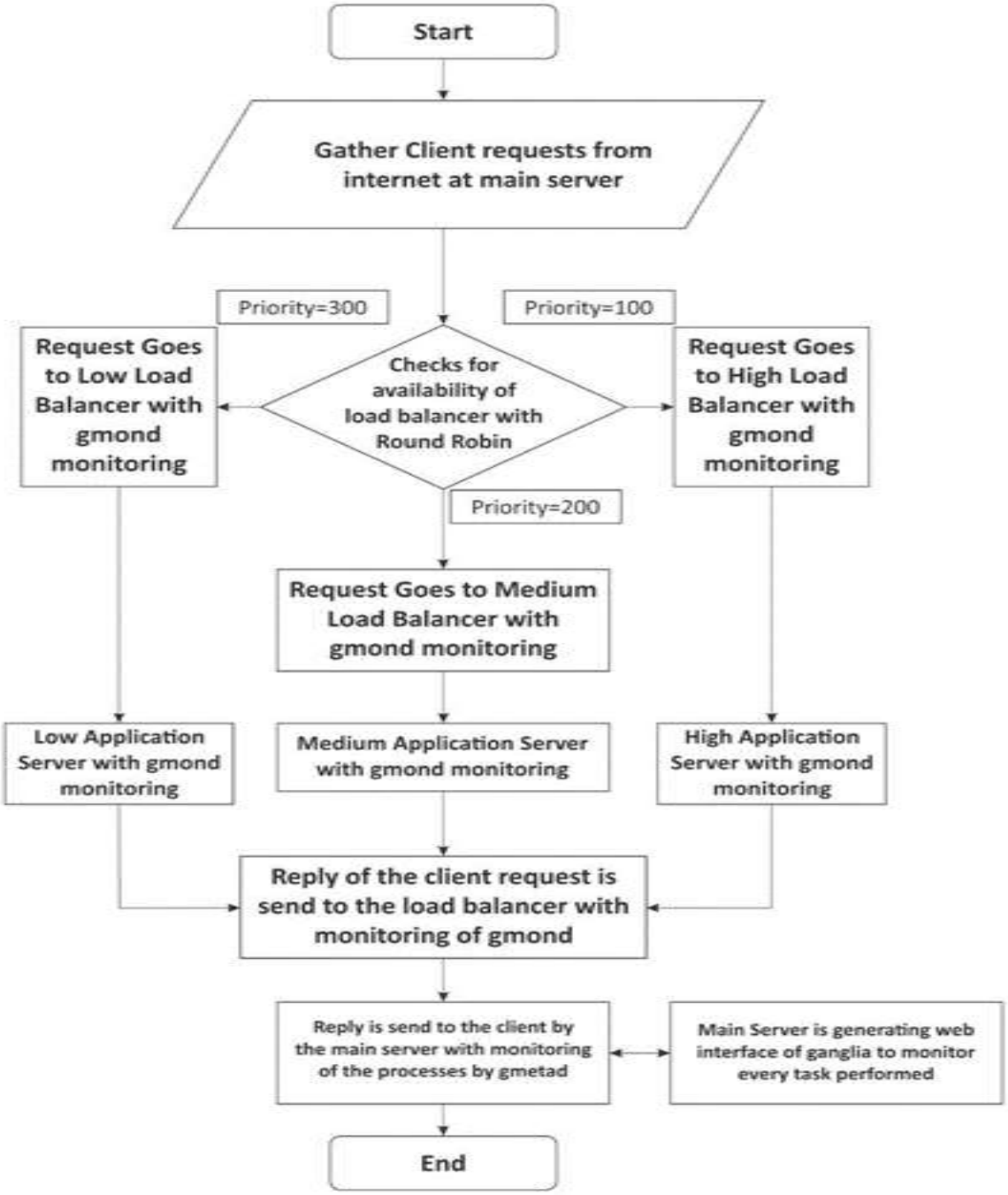

Figure 3.2.1 Proposed Workflow

3.3 Proposed Solution

Our proposed solution is seek to create viable solution that is load balancing and VM application migration based on round-robin algorithm way to deal with challenges distinguished in the balancing of load in cloud data center which

are highly loaded machines and needs to be distribute its load among all the servers located in cloud data center and also make the cloud data center highly available and fault tolerance in all conditions.

Steps for balancing the load and migrating the requests:

1. Input in the form of user requests from the floating IP for using applications installed in application server located in cloud data center.
2. Sorting the requests by the round-robin and haproxy among the load balancers located in cloud data center according to the priority of the load balancers and assignment of floating IP.
3. Request is handled by the choosen load balancer and the associate application server.
4. Response of the request is generated by application server and send it back to load balancer and load balancer sends the response to the user by the main server.
5. All the process is monitored by the Ganglia Monitoring by the services starting by gmetad, ganglia-monitor and apache.
6. The main server having the gmetad service on is able to show the web interface of ganglia and we can see the all process monitored.
7. All the load balancing servers, application servers, etc are giving the information of them to the main server by passing it to the main server by gmond.

4. Implementation

Implementation stage requires proper planning and understanding and limitations of existing system. It is the phase in which we apply our own methodology to mitigate the limitations and to enhance the existing mechanism. Below figure shows the cloud data center architecture setup to implement propose highly available and fault tolerance cluster in cloud data center.

Here we used Nginx for handling the requests, HAProxy to split the requests, Keepalived IP for failover capabilities, Floating IP for reassigning the IP address, Ubuntu 14.04 Operating System and Ganglia Monitoring System to implement a proper resource aware monitoring system in cloud data center.

4.1 HAProxy Configuration

Basically we installed and configured the web server with the help of Nginx which only listen for any requests on the private IP of the servers. After that we installed HAProxy which passes all the requests to all the web servers by HAProxy configurations. In HAProxy configuration file we set the traditional round-robin algorithm for balancing the load among the three servers and the location where HAProxy will pass the received traffic.

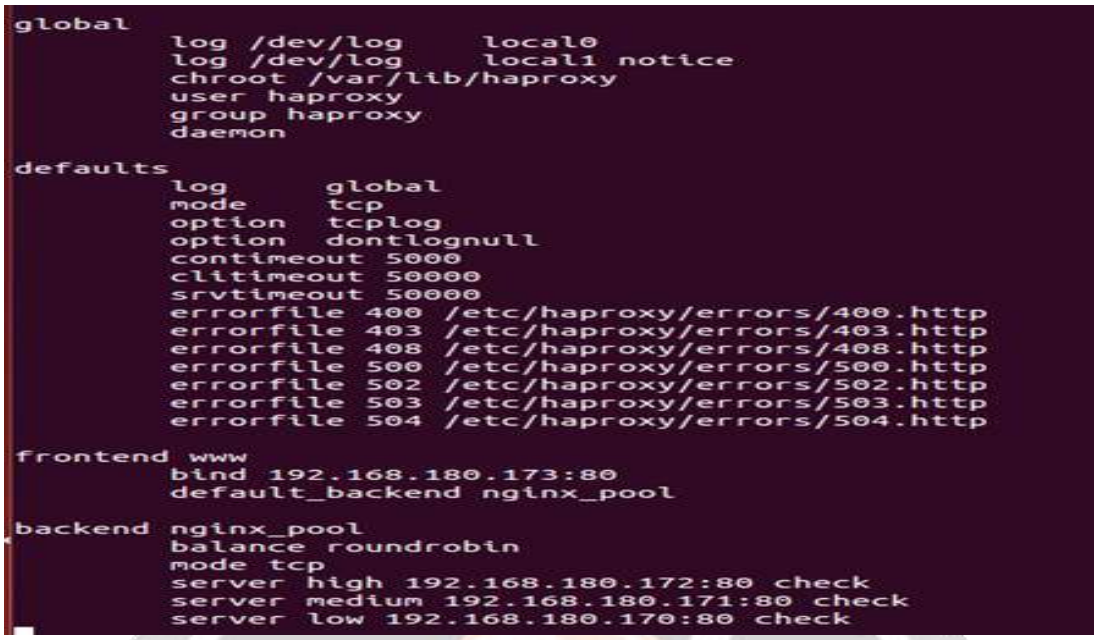

Figure 4.1.1 HAProxy Configuration File

4.2 Keepalived Configuration

Keepalived is installed and configured for the high availability of the cloud data center. In the configuration file of keepalived the heart beat is there to check for the availability of the server then there are server availability check, network check, etc. The network check will check for the network availability and if not then it transfers the application requests to the mentioned available Virtual Machines. There is also authentication block which is being used for the secure migration of the application. At last there a notify_master script which executes whenever the node becomes master in all load balancers. This scrip is responsible for triggering the reassigning of floating IP.

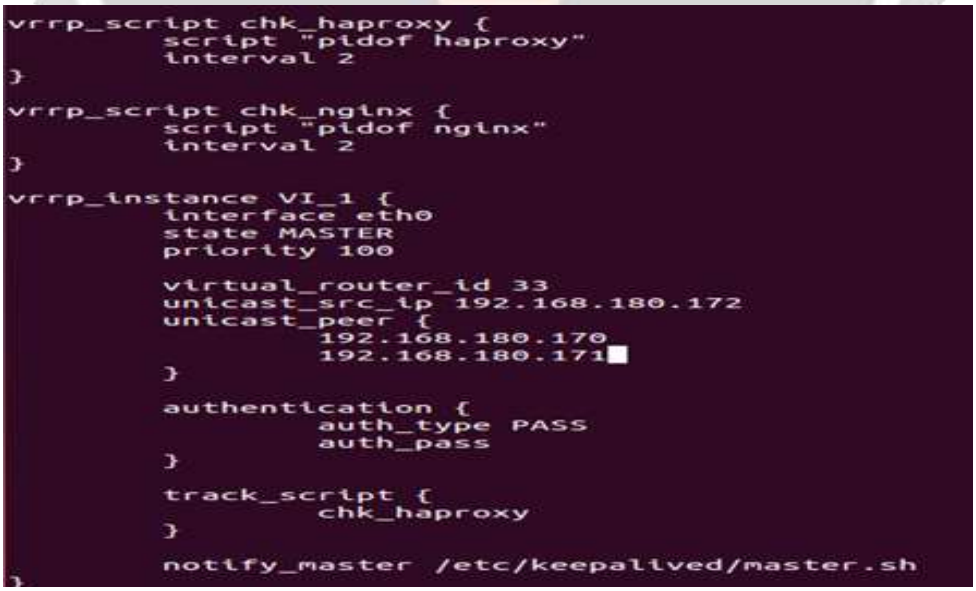

Figure 4.2.1 Keepalived Configuration File

The master.sh file is created on the all load balancers. Inside master.sh assign a variable called IP which holds the floating IP. Use curl to receive and assign metadata service for ID of the server we are currently on. Also check if this ID has the floating IP assigned to it or not and store the result in a variable named as HAS_FLOATING_IP. Now we ca use the variable to call the assign-ip script. We can only call the script in a scenario where floating IP is not already associated with ID. It helps to minimize API calls which helps to prevent the conflicting requests to the API in cases where the status of master switches between servers rapidly.

To handle the cases where floating IP already has an event in progress we will retry the assign-ip script for few times. Here we attempt to run the script for 10 times with an interval 0f 3 seconds between each call. The loop will terminate immediately if the floating IP move went successful.

```
IP='192.168.180.173'
ID=$(curl -s http://192.168.180.172/metadata/v1/id)
HAS_FLOATING_IP=$(curl -s http://192.168.180.172/metadata/v1/192.168.180.173/ipv4/active)

if [ $HAS_FLOATING_IP = "false" ]; then
        n=0
        while [ $n -lt 10 ]
        do
                python /usr/local/bin/assign-ip $IP $ID && break
                n=$((n+1))
                sleep 3
        done
fi
```

Figure 4.2.2 master.sh file

4.3 Configure the Ganglia Monitor system

On the master node we have installed ganglia-monitor, rrdtool, gmetad, ganglia webfrontend. We set the online graphical dashboard by copying the Ganglia web frontend and configuration file to Apache sites-enabled folder.

Figure 4.3.1 Ganglia Web Front-end

Set up your cluster named as bisagcluster in ganglia by editing in gmetad.conf file. It includes and configures where and how the gmetad daemon will collect data. And edit the gmetad.conf and gmond.conf file.

After starting the services of ganglia-monitor, gmetad and apache we can see the web front-end of ganglia monitoring system.

5. Conclusion

The paper addressed issue of resource aware load levelling for cloud data centers that we are able to thought-about because the serious issue of the cloud data centers. And currently a days a lot of and a lot of folks obtaining are becoming virtualized by IT infrastructure and getting committed cloud technologies therefore the use of cloud information centers square measure increasing day by day. We've got to use live migration techniques for migrating the VM application thus for that security is additionally main concern as a result of in migration security of VM can be compromised.

Therefore to create cloud data center a lot of economical and secure we tend to square measure addressing the higher than issue to implement the secure live migration techniques and essential load balancer and observation system that monitors the total cloud data center entities and provides some reasonably alert or notification for any issues which may be happens in future.

Acknowledgments

We are thankful to Shri T. P. Singh, Director, BISAG, for providing infrastructure and encouragement to carry out this project at BISAG for permitting to carry out the project at BISAG.

Reference

[1] Kumar, Ankit, and Mala Kalra. "Load balancing in cloud data center using modified active monitoring load balancer." Advances in Computing, Communication, & Automation (ICACCA) (Spring), International Conference on. IEEE, 2016.
[2] Shah, Syed Asif Raza, Amol Hindurao Jaikar, and Seo-Young Noh. "A performance analysis of precopy, postcopy and hybrid live VM migration algorithms in scientific cloud computing environment." High Performance Computing & Simulation (HPCS), 2015 International Conference on. IEEE, 2015.
[3] Soni, Gulshan, and Mala Kalra. "A novel approach for load balancing in cloud data center." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.
[4] Upadhyay, Ankit, and Prashant Lakkadwala. "Secure live migration of VM's in Cloud Computing: A survey." Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), 2014 3rd International Conference on. IEEE, 2014.
[5] Kharodia, Akbarali, et al. "A detailed analysis of distributed load balancing algorithm."
[6] McGilvary, Gary A., et al. "C2ms: Dynamic monitoring and management of cloud infrastructures." Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on. Vol. 1. IEEE, 2013.
[7] Zhao, Yi, and Wenlong Huang. "Adaptive distributed load balancing algorithm based on live migration of virtual machines in cloud." INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on. IEEE, 2009.
[8] Yang, Chao-Tung, et al. "Well-balanced allocation strategy for multi-cluster computing environments." Future Trends of Distributed Computing Systems, 2008. FTDCS'08. 12th IEEE International Workshop on. IEEE, 2008.
[9] Chen, Kun-Ting, Chien Chen, and Po-Hsiang Wang. "Network aware load-balancing via parallel VM migration for data centers." Computer Communication and Networks (ICCCN), 2014 23rd International Conference on. IEEE, 2014.
[10] Shrivastava, Vivek, et al. "Application-aware virtual machine migration in data centers." INFOCOM, 2011 Proceedings IEEE. IEEE, 2011.
[11] Tsygankov, Mykola, and Chien Chen. "Network aware VM load balancing in cloud data centers using SDN." Local and Metropolitan Area Networks (LANMAN), 2017 IEEE International Symposium on. IEEE, 2017.
[12] Vaidyanathan, Karthikeyan, H-W. Jin, and Dhabaleswar K. Panda. "Exploiting RDMA operations for Providing Efficient Fine-Grained Resource Monitoring in Cluster-based Servers." Cluster Computing, 2006 IEEE International Conference on. IEEE, 2006.

[13] Wang, Bo, et al. "MiCA: Real-Time Mixed Compression Scheme for Large-Scale Distributed Monitoring." Parallel Processing (ICPP), 2014 43rd International Conference on. IEEE, 2014.
[14] Singh, Gursharan, and Pooja Gupta. "A review on migration techniques and challenges in live virtual machine migration." Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2016 5th International Conference on. IEEE, 2016.
[15] Shethwala, Ridhvesh R., and Miren Karamta. "A Survey on cloud migration: Frameworks and security issues."
[16] Elsaid, Mohamed Esam, and Christoph Meinel. "Live Migration Impact on Virtual Datacenter Performance: Vmware vMotion Based Study." Future Internet of Things and Cloud (FiCloud), 2014 International Conference on. IEEE, 2014.
[17] Patel, Pradip D., et al. "Live virtual machine migration techniques in cloud computing: A survey." International Journal of Computer Applications 86.16 (2014).
[18] Wood, Timothy, et al. "CloudNet: dynamic pooling of cloud resources by live WAN migration of virtual machines." ACM Sigplan Notices. Vol. 46. No. 7. ACM, 2011.
[19] Elsaid, Mohamed Esam, and Christoph Meinel. "Multiple Virtual Machines Live Migration Performance Modelling--VMware vMotion Based Study." Cloud Engineering (IC2E), 2016 IEEE International Conference on. IEEE, 2016.
[20] Harney, Eric, et al. "The efficacy of live virtual machine migrations over the internet." Proceedings of the 2nd international workshop on Virtualization technology in distributed computing. ACM, 2007.
[21] Zhang, Bowu, et al. "Towards security-aware virtual server migration optimization to the cloud." Autonomic Computing (ICAC), 2015 IEEE International Conference on. IEEE, 2015.
[22] Wu, Hanqian, et al. "Network security for virtual machine in cloud computing." Computer Sciences and Convergence Information Technology (ICCIT), 2010 5th International Conference on. IEEE, 2010.
[23] Acharya, Shreenath, and Demian Antony D'Mello. "A taxonomy of Live Virtual Machine (VM) Migration mechanisms in cloud computing environment." Green Computing, Communication and Conservation of Energy (ICGCE), 2013 International Conference on. IEEE, 2013.
[24] Mansour, Ibrahim Ejdayid A., Kendra Cooper, and Hamid Bouchachia. "Effective Live Cloud Migration." Future Internet of Things and Cloud (FiCloud), 2016 IEEE 4th International Conference on. IEEE, 2016.
[25] Techopedia, definition, 29883, “data center virtualization”, accessed on 22nd November 2017, https://www.techopedia.com/definition/29883/data-center-virtualization
[26] Digitalocean, community, tutorials, how to set up highly available haproxy servers with keepalived and floating ips on Ubuntu 14.04 https://www.digitalocean.com/community/tutorials/how-to-set-up-highly-available-haproxy-servers-with-keepalived-and-floating-ips-on-ubuntu-14-04