

# An Autonomous Approach for High Availability and Fault Tolerance using Effective Monitoring in Cloud Data Center

Riddhi Trivedi  
ME Student  
GTU PG SCHOOL  
Gandhinagar, India

Miren Karamta  
Project Scientist  
BISAG  
Gandhinagar, India

Hardik Upadhyay  
Assistant Professor  
GPERI  
Mehsana, India

Dr. M. B. Potdar  
Project Director  
BISAG  
Gandhinagar, India

---

**Abstract:** The reason of grid monitoring and management is to detect offerings in grid surroundings for fault detection, performance evaluation, performance tuning, load balancing and scheduling. We have reviewed and worked on a self-managing fault-tolerance framework for high availability, scalability and reliability in cloud data center with proper monitoring tool which is deployed for Cluster monitoring and balancing purpose. This framework introduces the load balancers combined with Ganglia Monitoring System. The proposed framework is designed to overcome the issues regarding the high availability, fault tolerance, major downtime, self-healing, etc in cloud data center. HAProxy has been used to offer scaling to the servers for load balancing in proactive way. It additionally monitors the web servers for fault prevention on the client level. Our structure works with self-managed mirroring and load balancing in database servers with the help of MySQL master, replication of master database and Nginx consequently. Administrator supervise the running of servers through Ganglia Monitoring System because 24x7 manual monitoring cannot be possibly done by the cloud data center. For that we have carried out this scenario by using the virtualization technique in cloud environment.

**Keywords:** grid monitoring, Ganglia Monitoring System, cloud data center, HAProxy, load balancing, high availability, fault tolerance

---

## 1. INTRODUCTION

It's far a totally hard job in the cloud data center that how and where the computing is to be performed inside the cloud data center. Which will make a cloud dependable for the genuine customers, so many different issues are there like load balancing, fault tolerance, availability, safety in cloud data centers, etc. If by some means even after the occurrence of incidences of a few flaws in the system, if our architecture keeps running then the given structure is highly available and fault tolerable. So many strategies has been invented with a purpose to make cloud data center more reliable still it's far a totally hard activity to make it efficient. The components like very huge infrastructure, on call for carrier requirement inside the cloud data center consequences lots in the efficiently running of the cloud. In our proposition work using automated running, scaling, virtualization of a system, replication of data, monitoring of an environment and inside the computer code repository dynamic automation of software is finished using presumption fulfilled through an additional technique. In this given outlook dedicated approach has been used for high availability.

So as to representing TCP and HTTP based applications, load balancing between more than one servers a freely accessible High Availability Proxy is being used which presents distinguished accessibility. Through using the weight alternative supplied by means of the HAProxy corresponding to separate servers' solution provider can migrate the weight according to the requirements. Client side servlets are being used to send and accept the requests over HTTP. For making connections to the database iptable is being used. MySQL port is used in iptable to store the data in the concurrent database by taking the help of Nginx. Nginx is used as a load balancer in HAProxy which manages each of the database servers. The load between database servers are controlled by Nginx according to their precedence that is assigned to them. Live replication of the MySQL database is being performed by using master-master methodology for the purpose of

deployment over special virtual personal servers. We can copy the server data from one database server to another by using the replication. This facilitates us in including redundancy and allows in growing efficiency when users are looking to get access to the statistics. Then ganglia monitors the comprehensive infrastructure, recognizing of the trouble before its authentic occurrence, distinguishing the breaches associated with the safety and many others. VMware creates safe Linux Container for the applications so that the administrator and developers can port the applications together. The main goal of using VMware is to allow them deploying across systems.

## 2. LITERATURE REVIEW

A massive amount of work has been completed within the location of toes in cloud data center however there are many demanding situations in it like high availability, fault tolerance, protection, records control and so forth. For the resolution of high availability many research work is performed with various different methodologies. Many proactive and retroactive tools and methods have been invented for the execution of high availability in cloud data center like appropriate migration of application to other properly working server, rebooting the whole system, self-repairing, etc. and tools are used like HAProxy, Hadoop, etc. Normal architecture of a web server with load balancer and application server cannot provide the high availability in cloud data center. Here load balancers performs as data center's single point of presence. It is installed between the internet and the backend servers. This architecture causes a single point failure for the network so the new approach came which is performed as duplicating the content on multiple host machines, and balancing the load between those host machines with the help of DNS with basic round-robin load balancing. But DNS load balancing has the problems like alternative IP cannot be tried by users. The approach named failover support in which increasing the availability they are

configuring another load balancer as a backup for the primary load balancer but they don't have any monitoring mechanism in their implemented scenario.

In the given approach, to make system highly available the proactive strategy is being used by taking the help of HAProxy. For gaining the superior performance of database servers and application servers the load balancer is used to manage the load between all the servers and Ganglia Monitoring System is also used over load balancers to monitor the architecture and to send or receive the real time status to the administrator 24x7.

## 2.1 Nagios

Nagios is a freeware to use and edit. It is very easy to add a custom script for extension of service availability. Anyone can monitor any system by using the SNMP protocol over their systems. It provides you variety of plugins and add-ons for better development and download. It also gives alerts, notifications or comments on the status of the architecture. But the drawbacks are like, many features like wizards or interactive dashboards are not available for the freeware of Nagios. Configurations of some files are very hard. It has very confusing interface. It can't monitor the network usage or bandwidth available. It can just monitor the network but cannot manage it.

## 2.2 Ganglia Monitoring System

Ganglia, a scalable dispersed monitoring system which was built to label these issues and challenges. Ganglia Scalable Monitoring Systems at different points in the architecture includes large scale clusters and nodes in an instrument room, computing grids, existing association of clusters and viewing application on an open and shared platform [2, 3]. This system is depends on a hierarchical structure which is targeted at association of clusters. It depends on a multicast based listen/announce protocol [4, 5, 6, 7] to monitor the situation within the clusters and it uses a tree of peer-to-peer connections between illustrative clusters nodes to associate clusters and entirely their state. It holds widely used tools and technologies like XML for data characterization, XDR for dense, transferable data transport, and RRDtool for statistics catch and visualization. It is using smartly organized algorithms and data structures to gain very less overheads per node and high compatibility[1]. The execution is robust, and been transported to set of large scale operation systems and architecture of high processors, and is used over 2000 clusters around the world[1].

A). Working of Ganglia[1]: Tracking on a single cluster is implemented through the ganglia monitoring daemon (gmond) (Fig: 1). gmond is ready as a collection of threads, each assigned a particular venture. Assemble and post thread is answerable for converging neighborhood node records, publishing it on a famous multicast channel, and sending periodic heartbeats. The listening threads are responsible for listening on the multicast channel for tracking records from other nodes and updating gmond's in-reminiscence storage, a hierarchical hash table of monitoring metrics. In the end, a thread pool of xml export threads are devoted to accepting and processing consumer requests for monitoring information. All facts stored by using gmond is gentle United States and nothing is ever written to disk. This, blended with all nodes

multicasting their area, method that a new gmond comes into a fact really via listening and saying.

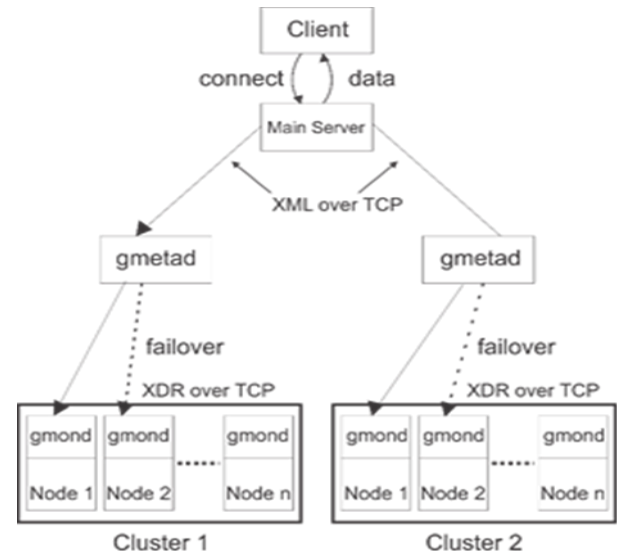


Fig. 1 Working of Ganglia

## 3. PROPOSED WORK

### 3.1 Coherent Benefits of New Approach

To gain high availability in cloud data center, the stated approach gives many amenities for the applications so the dynamic changes can be done very easily during the implementation. The facilities given are as below:

1) For the motive of quick and smooth revolution for any latest created server specimen and a load balancer, the VMware and HAProxy together is being useful. VMware provides OS level virtualization. The new specimen can be created very easily and dynamically by the VMware.

2) High Availability Proxy can manage the load with the help of weight option and Round-Robin method which have the priority options. If any one of the server fails to handle the request the requests can be handled by another servers.

3) Nginx is used as a load balancer in backend which helps in balancing the operations of database requests in a way that operations of database can be divided between a few database replicated servers in a better way by using different algorithms and weights alternatives.

4) By using the replication techniques number of MySQL node has been made to work together as a cluster [8]. So the replicated copy of data will always be there to provide fault tolerance and high availability.

5) Seeing that the architecture is having the cluster of two different databases. The problems like integrity of data, data lost, consistency of data which is related to database management can be reduced a lot.

6) The forecast of the availableness, fault and ongoing monitoring has been done using Ganglia Monitoring System. It acknowledges 24x7 to the administrator about the database servers, application servers and the load balancers because the administrator can't handle the large amount of data manually.

### 3.2 Lineaments of New Perspective

The proposed model which is used for the purpose of high availability is shown on the below flowchart in Fig 2. The framework provides high availability, fault tolerance and better Quality of service for the web applications which will run on cloud data centers. Seeing that the load balancing, use of replication of data, monitoring and automation in operating system level virtualization has been responsible for the application which is to be run in the highly available environment of cloud data center. When large number of users hits at the same time, the data should be available so the Operating system level virtualization has been implemented to create the dynamic reflection of the whole system and implement it in the same way the application works. The master-master replication has been implemented for the data to be secure and available for the clients 24x7. The high availability and fault tolerance can be protected up to higher scope for the reliable structure.



Fig. 2 Flowchart of Proposed Framework

1) Application Layer: The users can access the services which are located in cloud data center by this layer. By the use of HTML the user interface is being created which helps us to restrain the data which is to be sent as an input. Firstly the clients will interact with the HAProxy and after that the request will be forwarded to the upper division by the HAProxy. To transmit the data from web browser where the requested web page has been reloaded, the data blocks is being passed to the web server and at last to the process which is running in the background. The requests of the load balancers are being received and forwarded to the concurrent web server by this layer only.

2) Load Balancer: After accepting the requests from the application layer the request is being forwarded to the

concurrent web application server according to their weight and priority by the load balancer. The load balancer acts as an indicator of priority of the request. As per the configurations of the machines, the load can be assigned according to the computational powers of the running web servers.

3) VMware: The VMware which is open-source software helps in the deployment of the application by giving another layer of automation and abstraction of the operating system by using OS level virtualization. VMware can easily run multiple number of platforms so the deployment of application is easy in cloud data center.

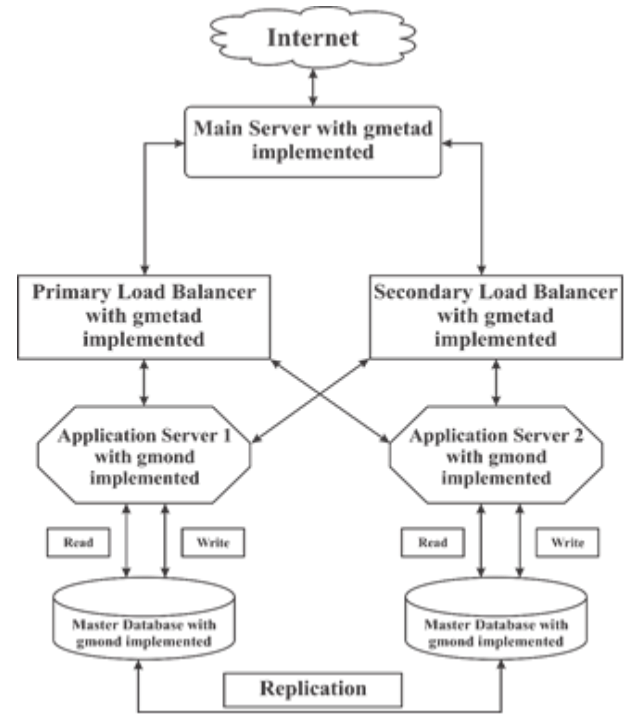


Fig. 3 Proposed Framework

4) Nginx: Nginx is very well known for the simple configuration, stability, better performance and very low memory usage, etc. The drivers sends the request to the Nginx for the corresponding database servers to save the data. After storing the data the status of both MySQL database servers is being checked and transfer the load between them in the manner of Round-Robin and transfers the priority of the load.

5) MySQL Database: MySQL master-master replication is being performed in order to provide redundancy and speed for the websites. The replication of MySQL emerges in formation of cluster that helps in gaining high availability in web configuration. The modification has to be done in the configuration file of MySQL which is my.cnf. to perform master-master replication [8]. The end of replication is performed by duplicating the status of master by both of the database servers that gets all the information about the database which has to be duplicated. The copied data through the servers by performing the replication are IP address of the servers, usernames, password, log files, etc.

6) Ganglia Monitoring System: Ganglia is a freeware system that helps in monitoring different services like CPU usage, network monitoring, aliveness of the nodes, etc. It examines the services which is working concurrently, the notifications

by any contacts in the situation when any problem gets arises or resolved. A health check program is being used to monitor the MySQL database and the connection between the active and passive load balancers. The parameters which is to be monitored are uptime, logs, master-slave sql running, connection time, connected threads, etc.

#### 4. OVERVIEW OF ARCHITECTURE

**A. Handling client request:** The client will send the request by getting into the IP address or Domain name system which is assigned to the application running. HAProxy server will forward the request to the web server without knowing any knowledge of the client.

**B. Balancing the incoming request between web servers:** After getting the request from the client HAProxy is to check the availability of the application servers and balancing the load between the application servers which are linked up with HAProxy [8]. According to the priority of the requests the HAProxy will forward the requests to the corresponding servers. The priorities are assigned to in the file of configuration with the help of weight option so load distribution is available as per the configuration of the system. The working of the servers are also being checked by the HAProxy. The statistical report is being generated when any of the server goes down. In HAProxy red colour indicates the failure of the server and the green colour running of the server.

**C. What if both the servers are fully occupied?:** HAProxy is efficiently balancing the load among the web servers. If at any case the number of requests are being increased which is more than the number of requests which is handled by the servers. In this scenario VMware can be used as an Operating system level virtualization. In this case VMware separates our infrastructure from the given infrastructure and handle our application in a managed way. Here, in such scenario VM image is generated which can hold our application. It results as dynamically more web servers can be installed by the use of VM images that's why the entire load can be managed easily.

**D. Load Balancing using Nginx and Ganglia:** By using Nginx for balancing the load and ganglia for managing it, we got the procedure through which we can distribute the incoming load of the traffic. The database requests among the different database servers as per the requirements. The load distribution can be decided by the configurations of VMs available at where database is loaded. So distribution of the requests between different VMs gives us the high availability and fault tolerance by giving the redundancy, stability to the application.

**E. Replication of Database:** High availability and the fault tolerance in database is being performed by configuring MySQL master-master Replication [8]. Requests of the database between the servers is managed by doing load balancing with the help of Nginx [9]. It can also tackle the queries which is related to write operations in database

servers. If number of write operations are not very frequently performed and application is mainly deals with the read operations then with the help of our proposed work we can increase the client response to much higher level.

**F. Monitoring MySQL:** As the database is having large number of entities stored in it, it is very difficult to find any incompatibility manually in the database. For monitoring the database the ganglia monitoring system is being used for the purpose of fault tolerance and high availability. Ganglia will address the administrator about the condition of the database server and if it is working in a proper condition or not. It notifies the admin through email, remote access, web interface, etc. Though administrator is able to perform monitoring 24x7.

#### 5. CONCLUSION

In this paper the proposed model and implementation provides the high availability, fault tolerance by safeguarding availability, scalability and reliability. The different conditions shows different situations like unexpected hike in traffic, growth of traffic, many internal problems like maintenance of the components, failure of the servers can be easily handled by the HAProxy. HAProxy with the help of Nginx and Ganglia balances the load between servers. It is also capable to redirect the load to the other server if any web servers is going through the failure. VMware provides the OS level virtualization by separating the applications from the system. Mirror copy of the database is being maintained by the Master-Master replication. Nginx is used to balance the load between primary and secondary servers and database servers and able to redirect the requests if any of the server fails. Ganglia Monitoring System enables the feature of automatic monitoring of the servers that increases the efficiency and accuracy. For the future work various types of database locks can be used by the database server for the effective use of database.

#### 6. ACKNOWLEDGMENTS

We are thankful to Shree T. P. Singh, Director, BISAG for providing infrastructure and encouragements. Special thanks to Mr. Margam Suthar, Principal, GTU PG SCHOOL, Gandhinagar for initial guidance.

#### REFERENCES

- [1] Massie, Matthew L., Brent N. Chun, and David E. Culler. "The ganglia distributed monitoring system: design, implementation, and experience." *Parallel Computing* 30.7 (2004): 817-840.
- [2] Foster, Ian, and Carl Kesselman. "Globus: A metacomputing infrastructure toolkit." *The International Journal of Supercomputer Applications and High Performance Computing* 11.2 (1997): 115-128.
- [3] Foster, Ian, Carl Kesselman, and Steven Tuecke. "The anatomy of the grid: Enabling scalable virtual organizations." *The International Journal of High Performance Computing Applications* 15.3 (2001): 200-222.
- [4] Amir, Elan, Steven McCanne, and Randy Katz. "An active service framework and its application to real-time

- multimedia transcoding." *ACM SIGCOMM Computer Communication Review*. Vol. 28. No. 4. ACM, 1998.
- [5] Chun, Brent, and David Culler. "Rexec: A decentralized, secure remote execution environment for clusters." *Network-Based Parallel Computing. Communication, Architecture, and Applications (2000)*: 1-14.
- [6] Fox, Armando, et al. "Cluster-based scalable network services." *ACM SIGOPS operating systems review*. Vol. 31. No. 5. ACM, 1997.
- [7] Stumm, Michael. "The design and implementation of a decentralized scheduling facility for a workstation cluster." *Computer Workstations, 1988.*, Proceedings of the 2nd IEEE Conference on. IEEE, 1988.
- [8] (2015) Digital Ocean, Inc.(US), master-master replication. [Online]. Available: <http://www.digitalocean.com/>
- [9] (2015) Nginx: Load Balancer. [Online]. Available: [wiki.nginx.org/Main](http://wiki.nginx.org/Main)
- [10] Garg, Ashima, and Sachin Bagga. "An autonomic approach for fault tolerance using scaling, replication and monitoring in cloud computing." *MOOCs, Innovation and Technology in Education (MITE), 2015 IEEE 3rd International Conference on. IEEE, 2015.*
- [11] Tchana, Alain, Laurent Broto, and Daniel Hagimont. "Approaches to cloud computing fault tolerance." *Computer, Information and Telecommunication Systems (CITS), 2012 International Conference on. IEEE, 2012.*
- [12] Bala, Anju, and Inderveer Chana. "Fault tolerance-challenges, techniques and implementation in cloud computing." *IJCSI International Journal of Computer Science Issues* 9.1 (2012): 1694-0814.
- [13] Jhavar, Ravi, Vincenzo Piuri, and Marco Santambrogio. "A comprehensive conceptual system-level approach to fault tolerance in cloud computing." *Systems Conference (SysCon), 2012 IEEE International. IEEE, 2012.*
- [14] Das, Pranesh, and Pabitra Mohan Khilar. "VFT: A virtualization and fault tolerance approach for cloud computing." *Information & Communication Technologies (ICT), 2013 IEEE Conference on. IEEE, 2013.*
- [15] Malik, Sheheryar, and Fabrice Huet. "Adaptive fault tolerance in real time cloud computing." *Services (SERVICES), 2011 IEEE World Congress on. IEEE, 2011.*
- [16] Gupta, Dhananjya, and Anju Bala. "Autonomic Fault Tolerant Framework for Web Applications." *International Journal of Computer Science and Telecommunication (IJCST)* 4.2 (2013): 528-533.
- [17] Santhosh, R., and T. Ravichandran. "Pre-emptive scheduling of on-line real time services with task migration for cloud computing." *Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on. IEEE, 2013.*
- [18] Dai, Yuanshun, Yanping Xiang, and Gewei Zhang. "Self-healing and hybrid diagnosis in cloud computing." *Cloud computing (2009)*: 45-56.
- [19] Anderson, Eric, and David A. Patterson. "Extensible, Scalable Monitoring for Clusters of Computers." *LISA*. Vol. 97. 1997.
- [20] Brewer, Eric A. "Lessons from giant-scale services." *IEEE Internet Computing* 5.4 (2001): 46-55.
- [21] Chakrabarti, Soumen, et al. "Automatic resource compilation by analyzing hyperlink structure and associated text." *Computer networks and ISDN systems* 30.1-7 (1998): 65-74.
- [22] Buyya, Rajkumar. "PARMON: a portable and scalable monitoring system for clusters." *Software-Practice and Experience* 30.7 (2000): 723-740.
- [23] Chien, Andrew A., et al. "High Performance Virtual Machines (HPVM'S): Clusters with Supercomputing APIs and Performance." *PPSC*. 1997.
- [24] Czajkowski, Karl, et al. "Grid information services for distributed resource sharing." *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on. IEEE, 2001.*
- [25] Foster, Ian, Carl Kesselman, and Steven Tuecke. "The anatomy of the grid: Enabling scalable virtual organizations." *The International Journal of High Performance Computing Applications* 15.3 (2001): 200-222.
- [26] Harren, Matthew, et al. "Complex queries in DHT-based peer-to-peer networks." *Peer-to-peer systems (2002)*: 242-250.
- [27] Rowstron, Antony, and Peter Druschel. "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems." *IFIP/ACM International Conference on Distributed Systems Platforms and Open Distributed Processing*. Springer, Berlin, Heidelberg, 2001.
- [28] Sottile, Matthew J., and Ronald G. Minnich. "Supermon: A high-speed cluster monitoring system." *Cluster Computing, 2002. Proceedings. 2002 IEEE International Conference on. IEEE, 2002.*
- [29] Van Renesse, Robbert, Kenneth P. Birman, and Werner Vogels. "Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining." *ACM transactions on computer systems (TOCS)* 21.2 (2003): 164-206.
- [30] Zhao, Ben Yanbin, John Kubiawicz, and Anthony D. Joseph. "Tapestry: An infrastructure for fault-tolerant wide-area location and routing." (2001): 70.