Ensuring High-Level System Availability of Computing Services

Yatrik Patel, Scientist D (CS) and Raja V Scientist B (CS)

Today number of institutes and their day-to-day activities are solely depended on IT. Almost every aspect of the institutional activity and services needs Internet connectivity and computing infrastructure to function. IT made the things faster, easier. At the same time, the responsibilities of implementer has increased multi-fold to ensure less downtime of applications and services. The cost of downtime is very high and may spoil the reputation of institution and its services. The best way to overcome the failure of the system or applications is setting up the high-level system availability. High-level system availability refers to a server or an application that ensures uninterrupted operational for a maximum possible time. In fact, a fault-free system or application that is expected to run without fail has to be well-designed and thoroughly tested before its deployment. Nowadays servers and underlying network have to manage a large volume of traffic to serve millions of clients. Perhaps, a single server may not sustain such a heavy load. In this computer era, it is very challenging task to achieve "100% Operational" or "never failing" of any computing environment. This can be achieved by implementing various redundancy mechanisms for hardware as well as software. Most of the organization prefers to deploy their applications on cloud so that the disaster recovery can be achieved with immediate effect without much complexity and safeguarding investments. High-level system availability can avoid having single point-of-failure. If failure occurs restoration can be performed in microseconds. This article attempts to give glimpses and technical highlights on achieving maximum uptime with minimal chances to failure of computing environment using presently available technologies.

Keywords—High-Level System Availability, Cloud Computing, Virtualization, Failover Cluster, Server Failover, Load Balancer, Virtual Network

1. Introduction

High Availability (HA) can be seen as the "holy grail" in the world of cloud computing[1]. Once high availability is ensured the requirement of disaster recovery can also be made part, so that end users will always expect uninterrupted service from the service providers without worrying about underlying server infrastructure. This can be achieved in many ways depending on the facility available with service providers as well as infrastructure. For cloud service providers, such as Azure, Google and Amazon Web Services, it is very easy to set it up by using their built-in tool(s). In case of any data centre maintained by the institution can also avail this infrastructure by using mechanisms like setting up high availability proxy such as HA proxy or mod_proxy(apache) which is open source and capable of running failover actions in case of unplanned IT service outages[1]. High Availability Proxy is used to provides a better accessibility[1]. Let's have an popularly available approaches, tools and technologies to ensure a nearly failsafe computing infrastructure.

2. Virtualization Techniques

Virtualization is technique to create a virtual version of resources or server. It is applied for sharing the capabilities of physical computers by splitting the computing resources among Operating Systems[2]. Using this technique, one can create multiple number of operating systems within a single OS mainly based on the storage, CPU and memory availability. Main server will be the base of all other servers. These computing techniques includes storage virtualization, server virtualization, operating-system level virtualization, network virtualization and application virtualization. Predominantly, most of the data leading virtualization companies like Microsoft Azure, Citrix, Oracle, Google, IBM, VM and Amazon are giving good services to the users. The most popular virtualization techniques are Full Virtualizations (main physical server to support applications and software to operate in a much similar way on virtualized divisions), Virtual Machines (VMs), Para-Virtualization and OS level Virtualization.

2.1. Cloud Computing

Cloud platforms provide geographically dispersed regions around the world[3]. It is a class of network-based computing. User may avail cloud computing on the basis of demand, resource pooling, elasticity, anywhere, anytime and anyplace. It has been classified as service models and deployment models, each service and deployment model offers different level of control and autonomy.

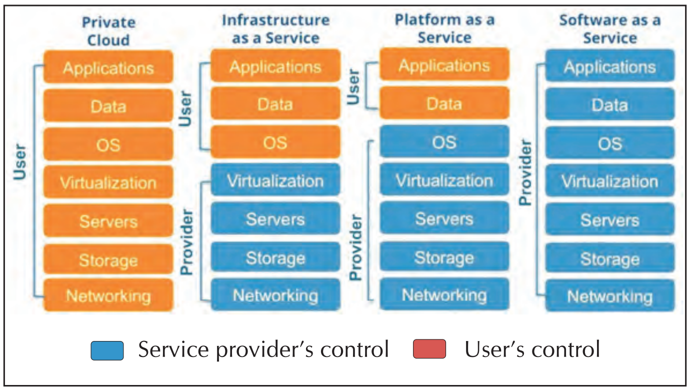

Fig. 1: Cloud Service Models and Controls Cloud Service

Cloud service models and controls Cloud service providers offer their services according to several fundamental models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS).There are four primary cloud deployment models(On premises /Private Cloud, Community cloud, Public cloud, and Hybridcloud). Each deployment model necessarily exhibits the previously discussed essential characteristics. The basic differences lie in the scope and access of cloud services, as they are made available to end users.

2.2. Network Virtualization

All network resources can be deployed and managed as logical services. System administrator can easily improve network efficiency, minimize the capital and operation cost, scalable, highly secure and availability throughout the system design. It is possible to make from multiple physical networks into one virtual network. Partitions can be possible in this network as per computing needs. In virtual network, all networked system does not have any physical (wired or wireless) connection between themselves. These systems are linked using network virtualization. System administrator can fully control the IP address blocks, DNS settings, security policies, and route tables within this network[4]. There are some important advantages by creating virtual networks such as Isolation, access to the public Internet, access to Virtual Machines (VM) within the VNet, name resolution, security and connectivity[4].

2.3. Storage Virtualization

It is a process of collecting the physical storage from various network storage devices into a single storage device that can be centrally managed. It is also known as cloud storage. Management of storage and data is bit difficult and more time-consuming process. But, it helps addressing the problems, easy backup, archive and fast recovery. It can be implemented by using software applications or appliances[5]. There are some valid reasons behind for this storage virtualization such as better storage utilization, heterogeneous IT environment and estimation of down time with automated management. Storage Area Network(SAN)is divided into four main layers (i.e. Storage devices, Block aggregation layer, File/record layer and Application layer).

3. How the IT Scenario of Organization Change because of Virtualized Environment and Cloud?

Cloud computing is the recent and advanced technology where institutes may host their services on shared basis. It is used to give better and faster performance to both the organizations and the users. Today's users are expecting speed, high reliability and safety from the service provider or institutions[6].

In early 1980s, users and institutes were fully dependent on IT department in case of any application development or IT operations. After 1990s, the scenario changed from IT Department to IT Support Services on base of user's demand. Now that the cloud computing is being made available to the users with user-friendly features, users have started using the applications and developing programs and applications on their own. Cloud makes it simple to maintain or handle the server. Now user can fix their stuff with more ease. Computing infrastructure is transparent, security breaches has declined, language barriers have diffused and everyone can bootstrap. It is very easy to have a high-level system infrastructure in cloud computing. There is no method to prevent failure in electronic component, at the same time we can have alternative solutions to overcome such failures using various tools available in open domain.

4. Tools available for High Availability Computing

The term "available" describes a system that provides a specific level of service as needed and "highly available" characterizes a system, which is designed to avoid the loss of service by reducing or managing failures as well as minimizing planned downtime for the system. Tools available for high-availability computing are described below:

4.1. High Availability Proxying

A proxy server is a kind of agent or intermediate hardware device that acts for another server. The main purpose of using proxy server is to provide better performance and consistency of a server environment by dispensing the workload to multiple servers (i.e. applications, web servers and databases). There are some open source proxying mechanism and tools like HA Proxy, NGINX, Dispatch-proxy or mod_proxy for apache are freely available. Predominantly, HA Proxy and mod_proxy (apache) are widely used by several organization because configuration part is much simpler than other tools. Public clouds like AWS and Azure have their own tool available for proxying.

4.2. Load Balancer

The load balancer provides facility to setup high availability application by managing or distributing incoming traffic requests from multiple clients. It can be achieved in two ways i.e. Internet facing load balancer and Internal load balancer. For public cloud users, this can directly implemented by using their inbuilt load balancer tools for application deployment. Some of the important load balancing algorithms are as follows:

4.2.1 Round Robin(RR) - This algorithm is commonly used in most of the organizations. Each and every request from the client machine is distributed to different servers across the network. It also adjusts the request based on the current server load[7].

4.2.2 Static Round Robin - It is more or less similar to Round Robin, each server is used in turns per their weights. There are no design limitations in this server environment. When a server load is high, it will immediately reintroduce the new server once the full map is recomputed.

4.2.3 Least Connections – This algorithm is dynamic in nature just like Round Robin. The server with the less number of inputs receives the connection. This approach is useful where sessions are long such as LDAP, SQL, TSE, etc.[7].

Source, URI and **URL parameters** algorithms are used by default in static mode.

4.3. Database Replication

Data replication is an important concept when databases are deployed over cluster of servers. Implementation of replication should not affect the current production server performance. Because of this difficulty, one have to be very careful and need to give guarantee for server consistency. In order to overcome the existing protocols, current data replication should be easier and incur less cost. Two approaches for database replication are as follows:

4.3.1 Master – Master

In RDBMS, one need to create failover mechanism using Master-Master replication method. It is also known as a mirror, which is by far the simplest technique one can use to increase the performance and the reliability of RDBMS server installation. In order to implement it one should have a secondary server[8]. Application can read the data from both master(s).

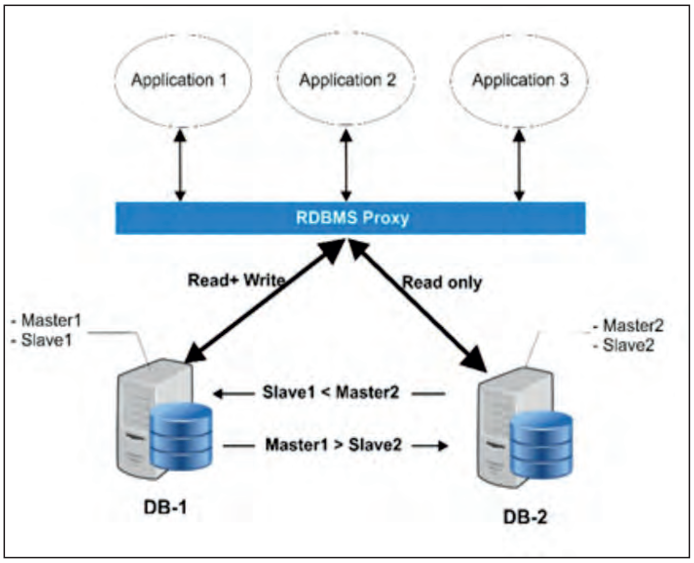

Fig 2: Multi Master Topologies

To setup a cluster environment one should have at least two servers or virtual machines (or hosting servers, each one providing an http web server (such as IIS or Apache) containing one or more web applications connected to a local RDBMS server instance on the same machine. Once a Master-Master replication mechanism between the two RDBMS instances has been established which basically means that a real-time web mirror has been setup, that can be used as a disaster recovery (see fig.2), and/or a 2-nodes web cluster managed by load-balancer – or any other balancing technique one might want to adopt[8].

4.3.2 Master-slave

It is a model of communication where the master will serve the data to slave. Application can read the data directly from slave(s) without impacting master server. Slave(s) can be taken offline and resync is possible without any downtime of master. But in case of failure a slave will take up the responsibility of master. Downtime and data loss may be possible when a master fails.

4.4. Failover Cluster

The Cluster is a group of similar hardware and other resources that acts as a single system. In the same way, failover cluster technology provides high availability and high throughput with low latency, while allowing for near linear scalability. High Availability cluster consists of at least two-nodes , since that is the minimum required to provide redundancy, but many clusters consist of many more nodes. Such configurations can sometimes be categorized into one of the following models.

4.4.1 Active/Active Cluster

Traffic intended for the failed node is either passed onto an existing node or load balanced across the remaining nodes. Each server will be configured with a specific application or service and provide redundancy for its relative server.

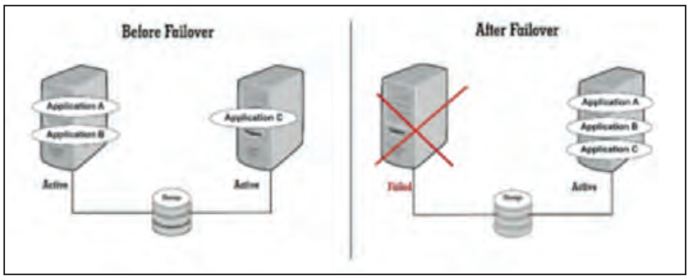

Fig. 3: Active/Active Cluster

4.4.2 Active/Passive Cluster

In an Active/Passive (or asymmetric) configuration, applications run on a primary, or master, server[9]. A dedicated redundant server is present to take over on any failure but apart from that it is not configured to perform any other functions.

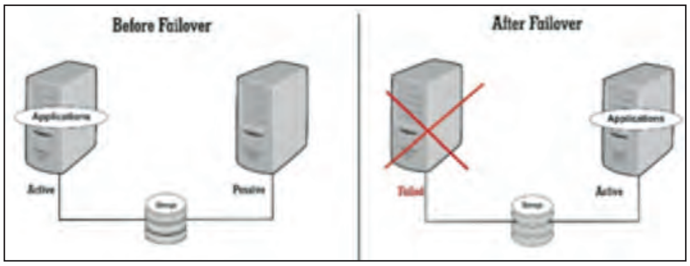

Fig. 4: Active/Passive Cluster

4.4.3 N-to-1 Cluster

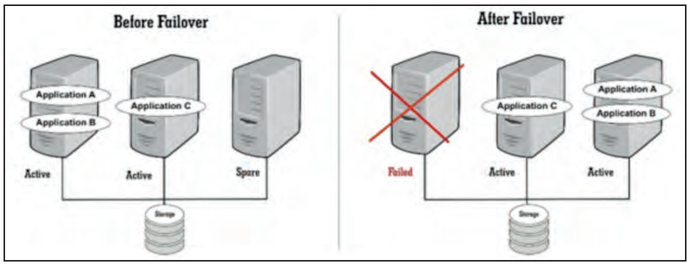

Fig. 5: N-to-1 Cluster

This configuration allows for the standby node to become the active one temporarily, until the original node can be restored and brought back online, at which point the services or instances must be failed-back to it in order to restore High Availability.

4.4.4 N+1 Cluster

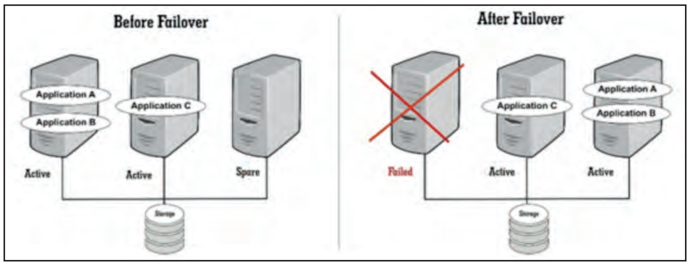

Fig. 6: N+1 Cluster

This configuration provides a single extra node that is brought online to take over the role of the node that has failed. This normally refers to clusters which have multiple services running simultaneously; in the single service case, this configuration degenerates to Active/Passive. In advanced N+1 configurations, an extra server in the cluster has spare capacity only[9].

4.4.5 N+M Cluster

In this case single cluster will be managed by many services. The drawback of this case would be having only one dedicated failover node and will not offer a quick redundancy. In such cases, more than one (M) standby servers are included and made available[9].

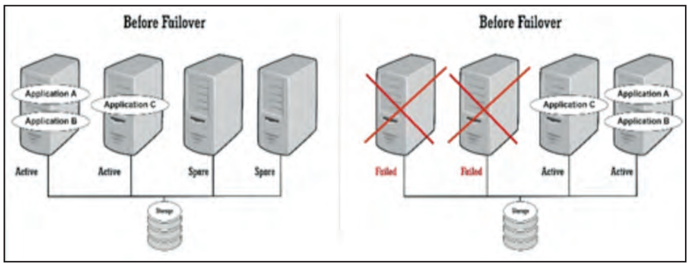

Fig. 7: N+M Cluster

4.4.6 N-to-N Cluster

It is a combination of Active/Active and N+M clusters. N-to-N clustering is the most complex of the failover configurations, and is typically used in a highly available architecture supporting multiple applications in a server consolidation environment[9].

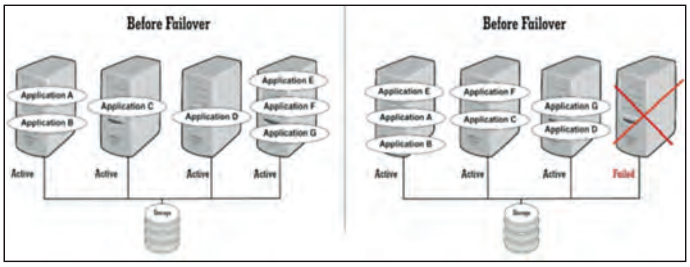

Fig. 8: N+N Cluster

This configuration is a logical evolution of N + 1 as it provides cluster standby capacity instead of a standby server in the cluster.

4.5. Load Testing

Load testing is the process of evaluating the capacity of a software system or computing device and measuring its response. It is performed to understand a system's conditions under both normal and anticipated peak load conditions. This performance testing is divided into four levels like performance test, load test, stress test and capacity test. Some of the additional concepts that are included in load testing are component test, investigation, smoke test, unit test and validation test. Performance testing is a broad and complex activity that can take many forms, address many risks, and provide a wide range of value to an organization[10].

5. Summary

From the above discussion one can easily observe that assuring 100% up time, or to design a completely failure-proof system is almost impossible. However, in-depth analysis and study of tools available reveals that it requires a many fold approach to ensure the infrastructure reliability and computing capacity needs.

A holistic approach has been depicted in following schematic to achieve maximum uptime.

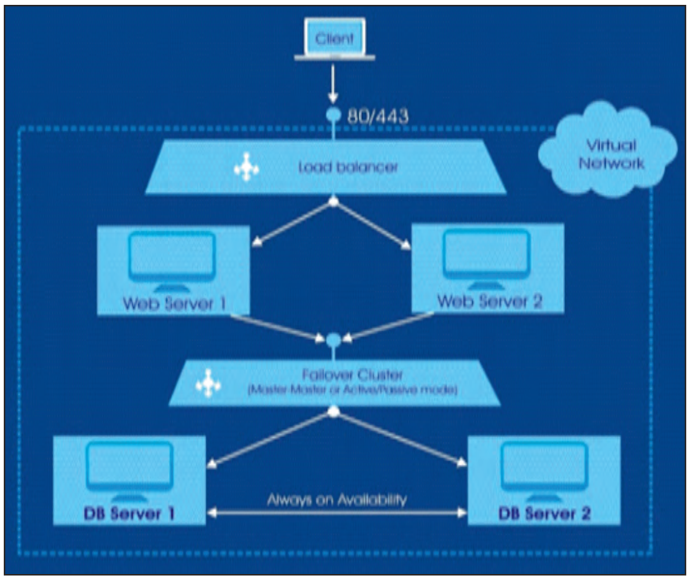

Fig. 9: High Level System Availability infrastructure

In above diagram, it has been assumed that the network is redundant (and virtual).The client facing web application can be made redundant by implementing high availability proxy and load balancing mechanism discussed earlier. The web application server can be clustered by taking any approach as discussed in topic "failover clusters" and database can be made redundant by deploying either of master-master or master-slave mechanism.

It should also be noted that before implementing fail-safe mechanism detailed requirement analysis considering multiple parameters e.g. services, expected availability, performance, cost, complexity should be carried out to choose the appropriate tool and approach for each segment.

References

[1] K. Benz and T. M. Bohnert, "Impact of pacemaker failover configuration on mean time to recovery for small cloud clusters," *IEEE Int. Conf. Cloud Comput. CLOUD*, pp. 384–392, 2014.

[2] F. Rodríguez-Haro *et al.*, "A summary of virtualization techniques," *Procedia Technol.*, vol. 3, pp. 267–272, 2012.

[3] Adam Glick, "Disaster Recovery and High Availability for Azure Applications | Microsoft Azure," 2016. [Online]. Available: https://azure.microsoft.com/en-in/documentation/articles/resiliency-disaster-recovery-high-availability-azure-applications/. [Accessed: 15-Aug-2016].

[4] Jim Dial, "Azure Virtual Network (VNet) Overview," 2016. [Online]. Available: https://azure.microsoft.com/en-in/documentation/articles/virtual-networks-overview/. [Accessed: 22-Aug-2016].

[5] D. A. S. T. Anu Mendiz. R, "Storage Virtualization in Cloud Computing," vol. 3, no. 3, pp. 63–65, 2016.

[6] Vittorio Della Rossa, "How will the IT organization change because of cloud innovation□ - Cloud computing news," 2014. [Online]. Available: https://www.ibm.com/blogs/cloud-computing/2014/10/will-organization-change-cloud-innovation/.

[7] Jovan Katic, "Load Balancing with HAProxy | MaxCDN," 2015. [Online]. Available: https://www.maxcdn.com/blog/haproxy-load-balancing/. [Accessed: 10-Oct-2016].

[8] Ryan, "MySQL Master-Master Replication setup in 5 easy steps," 2015. [Online]. Available: http://www.ryadel.com/en/mysql-master-master-replication-setup-in-5-easy-steps/.

[9] R. Natário, "Networks and Servers: Failover Clustering (IV)." [Online]. Available: http://networksandservers.blogspot.in/2011/09/failover-clustering-iv.html. [Accessed: 10-Nov-2016].

[10] D. R. J.D. Meier, Carlos Farre, Prashant Bansode, Scott Barber, "Chapter 2 – Types of Performance Testing," 2007. [Online]. Available: https://msdn.microsoft.com/en-us/library/bb924357.aspx.