IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

A Scalable and Highly Available Distributed Architecture for e-Governance Applications on Private Cloud Platform

Priyeshkumar T. S.[1*], Dharmendra Devaka[2]

[1, 2]National Informatics Centre (NIC), Govt. of India, Gandhinagar (Gujarat), India

[*]*Corresponding Author: priyesh.ts@nic.in, Tel.: +91-95102-46341*

DOI: https://doi.org/10.26438/ijcse/v7i3.811814 | Available online at: www.ijcseonline.org

Accepted: 12/Mar/2019, Published: 31/Mar/2019

***Abstract*—** e-Government improves the efficiency of public administration, increasing transparency, deducing administrative corruption, improving service delivery, citizen's empowerment and improving government finance. The private cloud provides the ideal environment for scalable e-Government applications because of its scalable and elastic characteristics. The scalability can be achieved horizontally and vertically within short time in cloud environment. During design and development phase of application, special consideration is needed for scalability and high availability on Application and database layer. Application layer high availability and scalability can be achieved with a server load balancer. Database layer high availability and scalability can be done by database sharding within a database fail-over cluster. In this paper, we propose a scalable highly available architecture for e-Governance applications on private cloud environment.

***Keywords*—**Availability, Database sharding, Failover Cluster, Load Balancer, Scalability.

I. Introduction

An application is scalable if it gracefully handles dynamic load by adding resources to application infrastructure horizontally or vertically. We can use cloud elastic and scalable characteristics to achieve it effectively. In Vertical Scaling, more resources are added to a virtual machine [1, 2, 13]. So, if we have a large dataset, this could mean extending disk drives. It could also mean moving the compute operation to a larger server with more memory or speedier CPU. The point is that we take a single resource and increase its handling capability. In Horizontal Scaling, we add more virtual machines [1, 13]. So, we might add another server to store some parts of a large dataset, or in the case of a computing resource, we would split the load over additional virtual servers. We will need to include it in our initial design with a distributed application architecture.

An application is highly-available if it continues to function despite expected or unexpected failures of components in the application infrastructure. If one or more components fail, a resilient architecture keep application fault tolerant—continuing to function with rest of the system without noticeable service impact. A resilient application requires planning from both a software development phase and an application architecture design phase. In this paper, we focus on application architecture design. Scalability and high availability (fault resilience) are two key infrastructure requirements that organizations must consider in the architectural design of their critical e-governance applications.

Rest of the paper is organized as follows, Section II contain the related work of application and database technologies to make application highly available with scalability, Section III introduce distributed application architecture, Section IV contains scalability and high availability impact at application and database layer and Section V concludes research work with future directions.

II. Related Work

NIST defines cloud computing as "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [3]. The cloud technology standardizes and collects IT resources to automate many of the maintenance activities otherwise performed manually under conventional or client/server model of computing [4, 14]. It offers a great advantage in terms of immediate as well as long term cost savings for governments. As the model offers services based on a pay-as-you-go and pay-per-use basis, there are no upfront costs involved in buying IT equipment. The savings can allow cost effectiveness come in these financially challenging times. Cloud computing can be the backbone on which governments can create a more trusting environment for e-governance and provide the benefits of cost savings, efficiency, improve delivery mechanisms etc. [5].

Now a days, Private Cloud is implemented in most of the government owned state and national data centers [1, 6]. In private cloud, data and processes are managed by organization itself. It provides better and controlled infrastructure for security.

To design a scalable and resilient application architecture, we use the following technologies in proposed architecture.

A. Server Load Balancer

Server Load balancers (SLB) are critical component of architecture for scalable application deployment [7]. The concept is to distribute the load as the numbers of simultaneous connections increase and to route connections to request virtual machines. Thus an application can increase services just by adding virtual machines. SLB monitor health of virtual machines or services running on it and distribute load across multiple virtual machines that can best handle the requests.

B. Failover Cluster

A failover cluster aims to provide high availability for services or applications that run within it [8]. It contains a group of independent nodes that work together to increase the availability of applications and services. Failover clustering can protect against hardware and software failures by failing over resources from one cluster node to another as required. Failover is the process of taking a clustered service or application offline on one node and bringing it back online on another node. This process is typically transparent to the users, who should experience a minimal disruption of service when a failover occurs.

C. Database Sharding

The most common horizontal scaling is the breaking up of services into partitions (shards) [9]. Some of the advantages of sharding are massive scalability, high availability, faster queries; more write bandwidth, reduced cost as databases can run on commodity servers. The basic concept of database sharding or horizontal partitioning is that take a large database and break it into a number of smaller databases across multiple servers [9, 10]. Each shard (running on its own node) contains a portion of the original monolithic database, and is sharded based on application-specific rules. For example, e-governance database shard by geographical zones, with each shard containing a specific group of user related information in that particular zone.

III. METHODOLOGY

A node is single entity machine or virtual server. A server farm is a group of servers serving identical content--otherwise the user might receive inconsistent content. Adding more servers to server farm will increase potential load capacity by spreading the load over multiple servers.

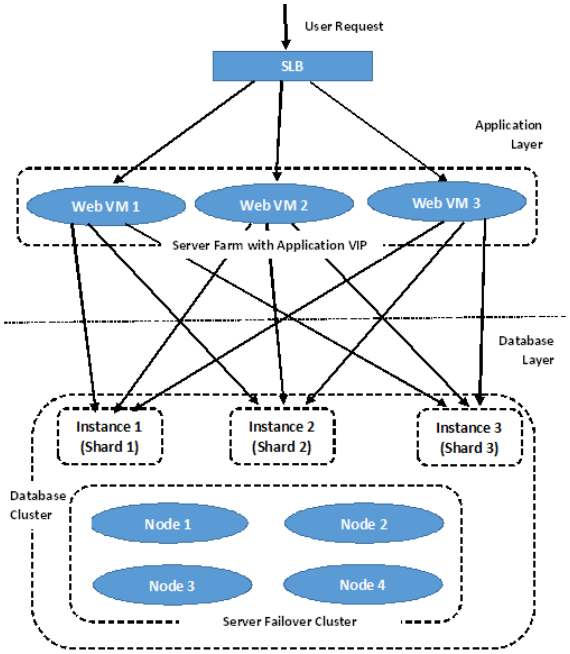

Figure 1. Distributed Application Architecture

Figure 1 shows scalable highly available distributed application architecture. There is a virtual IP (VIP) for each application's server farm. The user accesses the application through SLB, which forwards the user's request to one of the server in application's server farm. Whichever application server is selected will respond directly to the user's request. The SLB exposes this VIP address that customers use to access the application. This VIP address can be associated with a DNS entry. Incoming requests are distributed across the nodes in a farm based on load distribution algorithms [11]. The SLB has inbuilt health monitoring mechanism to automatically replace node that have failed or have become unavailable. An HTTP health check specifies the port and path to execute the health check against on each service running on node. An ICMP health check is done to check availability of node. When a new server is added to server farm, SLB discover the new member and start to sending requests to new server automatically. SLB maintain source IP persistency (sticky session) so that the same requester is always routed the same node based on persistency rule.

On database layer, database instances (shards) are running on top of server failover cluster. Each instance has its own virtual IP. Web server connect with database instance using this VIP. In general, if active node fails, all the database instances running on that node will move to passive node in the cluster. Passive node are optional. In the absent passive node, failover happens as per failover policy defined in the cluster configuration. Usually the number of cluster nodes are higher than number database Instances with few passive nodes to get equal or higher computing resources in order to avoid performance degradation after a failover. Application user can fetch data from one or more database instances based on application logic. Database queries and write operations are then performed in one of two modes:

- Single-Shard: Each database operation (read or write) is performed against a single shard, such as information about single citizen or group of citizen from a particular geographical zone (shard).
- Multi-Shard: This typically applies to analytic queries, performed in parallel across one or more shards, enabling impressive performance results. This is usually an input for management information system such as overall performance comparison across all geographical zones of a state or country.

IV. Results and Discussion

We have implemented this architecture for two critical e-Governance projects – Public Distribution System (PDS) and Common Services Portal (CSP) for Government of Gujarat. In this section, we point out positive impact of proposed architecture.

A. Application Layer

Cloud has auto scaling feature to scale virtual machine horizontally or vertically [12]. Horizontal scalability can be easily achieved by adding a new server to server farm by replicating existing virtual machines. By configuring basic system configuration (like IP configuration, Hostname etc.) in replica virtual machine, we can straight away add new production virtual machine to server farm with zero application downtime. Application is available to access in case of one or more virtual machines are down. SLB detect faulty machines and don't route traffic those virtual machines. There might be slight performance impact in case of multiple virtual machines failures at the same time.

B. Database Layer

Database Instances are highly available in cluster environment. The advantage of the database sharding approach is improved horizontal scalability, growing in a near-linear fashion as more nodes are added to the failover cluster. There are several other advantages of shards,

- Administrative is easy - Production databases must be fully managed for regular maintenance activities such as backups, database optimization, indexing and other common tasks. By using the sharding approach, each individual shard can be maintained independently, performing such maintenance tasks in parallel with short time window.
- Performance Improvement - By hosting each shard database on its own server, the ratio between memory and data on disk is properly balanced, thereby reducing disk I/O and maximizing system resources. This results in less contention, greater join performance, faster index searches and fewer database locks. Therefore, not only can a sharded system scale to new levels of capacity, individual transaction performance is benefited as well.
- Less Cost - Sharding works well with commodity multi-core server hardware, systems that are far less expensive when compared to high-end, multi-CPU servers and expensive storage area networks.

In e-governance applications, load is dynamic as per time to time government policy changes. The proposed architecture is suitable to handle such situation effectively and efficiently.

In e-Governance environment, the following challenges on application and database layer has to be considered with proposed architecture.

Application execution code should be same in all members of server farm. When developer update application code to add additional functionalities or content changes, the code has to be updated and in sync in all machines without any manual mistakes. If auto sync software is using for application code synchronization, a manual testing on each server should be carried to ensure it. Another challenge is that in case one of the virtual machine is having some technical issue, there should be a mechanism to find out the problematic virtual machine. The problematic virtual machine has to isolate from production environment by removing from application server farm.

There are some challenges with partitioning because anytime you distribute data or functions among multiple servers, not the least of which is data locality. In case of multi-shard query, if data is not local when needed, servers will have to "go fetch", which lead to long query execution time and slow user response. A second challenge is an inconsistency. If different services are writing and reading from a shared source, there can be an incident in which someone is sending a request for something at the same time it is being updated by someone else.

V. Conclusion and Future Scope

In this paper we introduce a scalable highly available architecture for e-Governance applications on cloud environment using server load balancer and failover clustering. We have explained how availability and scalability is achieved on application and database layer. Application is available in the event if unexpected disaster or maintenance activity. The data in clusters dispersed as multiple shards. If a particular data node is down, shard will move to another node and continue to deliver data without noticeable service impact. The architecture ensure high application service delivery uptime and scalability in case of high application service demand. The architecture uses most of the advantageous features of cloud computing technology. Also we mention some of challenges and cloud auto scalability feature on which we will focus in future for further improvement.

REFERENCES

[1] Ab Rashid Dar and Dr. D. Ravindran, "*Survey on Scalability In Cloud Environment*", International Journal of Advanced Research in Computer Engineering & Technology, Vol.**5**, Issue.**7**, pp.**2124-2128**, **2016**.

[2] Kamyab Khajehei, "*Role of virtualization in cloud computing*", International Journal of Advance Research in Computer Science and Management Studies, Vol.**2**, Issue.**4**, pp.**15-23**, **2014**.

[3] Peter Mell and Tim Grance. "*The NIST Definition of Cloud Computing*", NIST Special Publication 800-145, **USA**, pp.**2**, **2011**.

[4] Haroon Shakirat Oluwatosin, "*Client-Server Model*", IOSR Journal of Computer Engineering, Vol.**16**, Issue.**1**, pp.**67-71**, **2014**.

[5] Anand More and Priyesh Kanungo, "*Use of Cloud Computing for Implementation of e-Governance Services*", International Journal of Scientific Research in Computer Science and Engineering, Vol.**5**, Issue.**3**, pp.**115-118**, **2017**.

[6] Solanke Vikas, Kulkarni Gurudatt, Maske Vishnu and Kumbharkar Prashant, "*Private Vs Public Cloud*", International Journal of Computer Science & Communication Networks, Vol.**3(2)**, pp.**79-83**, **2013**.

[7] P. Beaulah Soundarabai, Sandhya Rani A., Ritesh Kumar Sahai, Thriveni J., K.R. Venugopal and L.M. Patnaik, "*Comparative Study on Load Balancing Techniques in Distributed Systems*", International Journal of Information Technology and Knowledge Management, Vol.**6**, No.**1**, pp.**53-60**, **2012**.

[8] Dubravko Miljković, "*Review of Cluster Computing for High Available Business Web Applications*", Proceedings of MIPRO 2008/GVS, Opatija Croatia, pp.**261-266**, **2008**.

[9] Sikha Bagui, "*Database Sharding: To Provide Fault Tolerance and Scalability of Big Data on the Cloud*", International Journal of Cloud Applications and Computing, Vol.**5(2)**, pp.**36-52**, **2015**.

[10] Pankaj Deep Kaur and Gitanjali Sharma, "*Performance of Scalable Data Stores in Cloud*", International Journal of Engineering and Advanced Technology, Vol.**4**, Issue.**5**, pp.**212-216**, **2015**.

[11] Kandi Phani Sai, Sri Rohith and Abhineet Anand, "*Analytical Study of different Load balancing algorithms*", International Journal of Advanced Studies in Computer Science & Engineering, Vol.**7**, Issue.**1**, pp.**21-26**, **2018**.

[12] Pooja C.S and K.R Prasanna Kumar, "*Survey on Load Balancing and Auto Scaling Techniques for Cloud Environment*", International Journal of Engineering and Advanced Technology, Vol.**6**, Issue.**5**, pp.**28-30**, **2017**.

[13] C. Venish raja1 and L. Jayasimman, "*A Survey on Scalability in Cloud Computing*", International Journal of Computer Sciences and Engineering, Vol.**6**, Special Issue.**2**, pp.**471-474**, **2018**.

[14] Kalburgi Tayyaba, Ibrahim and Sajjan R.S, "*Cloud Resourse Virtualization*", International Journal of Computer Sciences and Engineering, Vol.**4**, Special Issue.**4**, pp.**21-27**, **2016**.

Authors Profile

Mr. Priyeshkumar T S pursed M.Tech. Computer Science from IIT, Guwahati in 2009. He is currently working as Scientist – C (Senior System Analyst) in National Informatics Centre(NIC), Gujarat. His main area of interests are Cloud Computing, Distributed Systems, High Availability Solutions and Big Data Analytics. He has 10 years of technical experience in Data Centre Technologies.

Mr. Dharmendra Devaka pursed B.E. Computer Engineering from L.D. College of Engineering , Ahmedabad in 1991. He is currently working as Scientist – E (Technical Director) in National Informatics Centre(NIC), Gujarat. His main area of interests are Information Security, Network Security, Cloud Computing, and Big Data Analytics. He has 16 years of technical experience in eGovernance Project Implementation and 6 years of experience in Data Centre Technologies.