IJCSE International Journal of Computer Sciences and Engineering Open Access

Review Paper Vol.-6, Issue-5, May 2018 E-ISSN: 2347-2693

Online Database Load Balancer to Collaborating With Existing Database

Shital Pawar[1], Sadiya Shaikh[2], Priyanka Wadagave[3], Megha Chavan[4*], Deepika Tambat[5]

[1,2,3,4,5] Dept. of Computer Engineering, Bharati Vidyapeeth College of Engineering for Women, Pune, India

**Corresponding Author: meghachavan1997@gmail.com*

Available online at: www.ijcseonline.org

Accepted: 17/May/2018, Published: 31/May/2018

Abstract: According to literature meaning of Cloud Computing is distributed computing, storing, sharing, and accessing data over the internet. Cloud is the platform which provides numerous type of resources where the end user may use the resource for developing own software and even their own cloud and even include a new resource to the existing once. The biggest issue for a cloud datacentre is to tackle with billions of request coming dynamically from the end users to handle their database in efficient and effective manner. To achieve this goal, various load balancing approaches have been proposed in past years. Database load balancing strategies aim at achieving high software developer satisfaction by producing service like auto scale of their data in database, zero-downtime, multiple database choices, multi tenancy support. Load balancing in this environment means equal distribution of workload across instances. End users needs ample space to store their database data to decrease the maintenance cost and buying cost of servers and area required to assemble them this paper focus on the balancing the data in database. This paper, focus on database based load balancing which works well in cloud environment, considers resources specific demands of the tasks and reduces overflow of data overhead by dividing the data on running instances.

Index Terms **-**Auto scalability, Cloud computing, load balancer, Multi tenancy, Snapshot, Zero downtime

I. INTRODUCTION

Now a days cloud computing is the new era of computing these days. Cloud computing is capable of providing an easy way to store and access the stored data files and files from data centres which are situated at different geographical locations. The basic services of cloud infrastructure as a service (Iaas), platform as service (Paas), Software as a service (Saas), anything as a service (Xaas).These services provide a pay per use model to its user. Cloud Computing is having different deployment models. The deployment models are categorized in four different models: Private cloud, public cloud, community cloud, hybrid cloud and it possess five unique characteristics i.e. on demand self-service, elasticity, auto scaling, resources pooling. We have chosen AWS (Amazon Web Services) which is a public cloud for deployment of our proposed system. Our proposed system will be a web application. Our application will provide different services like compute services, storage services and management services and security services, these services are used to handle network traffic or data traffic.Our system will provide load balancing and auto scaling schema for large amount of data handling purpose.

II. OBJECTIVE

Our proposed system will be a web application. It provides zero downtime service to user, automatically scales the data and user can upload their files in multiple availability zones. Also it supports for multiple databases i.e. relational or non-relational databases.

III. SCOPE

This project will consist of creating the load balancer which will balance the load on different target i.e. Instances.

- It acts as the abstraction between the application and database.
- Rapid performance levels within the same datacenter or different datacenter.
- Reduce maintenance cost.
- Backups are provided.
- Scalability.

IV. SYSTEM FEATURES

List of the features:

1. **Elastic load balancing**

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon Elastic Compute Cloud (Amazon EC2) instances. You can set up an elastic load balancer to load balance incoming application traffic across Amazon EC2 instances in a single Availability Zone or multiple Availability Zones.

2. Auto Scaling

Elastic Load Balancing also offers integration with Auto Scaling, which assures that you have the back-end capacity available to meet varying traffic levels. Let's say that you want to make sure that the number of healthy EC2 instances behind an Elastic Load Balancer is never fewer than two. You can use Auto Scaling to set these conditions, and when Auto Scaling finds that a condition has been met, it automatically adds the requisite amount of EC2 instances to your Auto Scaling Group. Here's another example: If you want to make sure to add EC2 instances when the latency of any one of your instances exceeds 4 seconds over any 15 minute period, you can set that condition. Auto Scaling will take the appropriate action on your EC2 instances, even when running behind an Elastic Load Balancer. Auto Scaling works equally well for scaling EC2 instances whether you're using Elastic Load Balancing or not.

3. Monitoring the Environment

One of the benefits of Elastic Load Balancing is that it provides a number of metrics through Amazon Cloud Watch. While you are performing load tests, there are three areas that are important to monitor: your load balancer, your load generating clients, and your application instances registered with Elastic Load balancing (as well as EC2 instances that your application depends on).

4. Monitoring Elastic Load Balancing

Elastic Load Balancing provides the following metrics through Amazon

- Cloud Watch
- Latency
- Request count
- Healthy hosts
- Unhealthy hosts Backend 2xx-5xx response count
- Elastic Load Balancing 4xx and 5xx response count

As you test your application, all of these metrics are important to watch. The particular items of interest are likely to be the Elastic Load Balancing 5xx response count, the backend 5xx response count, and latency.

5. Monitoring the Load Generating Clients

It is also important to monitor any clients that are generating load to ensure that they are able to send and receive responses at the load you are testing. If you use a single test client running in Amazon EC2s, make sure it is sufficiently scaled to handle the load. At higher loads, it is necessary to use multiple clients, because the client may be network-bound and may be affecting the test results.

6. Security

Security is provided to user through IAM (Identity and Access Management) to user's data.

V. EXISTING SYSTEM

Big – IP F5 is dell's product which is a software based product but it handles only one database at a time hence it provides a single tenancy support. Single tenancy support means it allows only one user to upload file at a time. Also it doesn't provide elasticity.

KEMP is one of the load balancer. The main drawback of it is that it is a hardware based load balancer hence it doesn't provide Auto scaling and elasticity. Sometimes because of some technical problem (such as network connection) user will not able to balance its data, this is the major drawback of KEMP. As it is hardware based load balancer it costs more than a software based load balancer application (User need to buy it for load balancing).

Scale arc is used in market now a days but it has some drawback. It is a software based load balancer .It handles load balancing of only relational databases it does not support non-Relational databases such as Mongo DB, Dynamo DB etc. User need to specify each and every action while using Scale arc such as whether user wants to balance a load on his data, or he wants to scale his data, whether he wants to back up his data or he wants some disaster recovery sites to store his data, whether he wants a snapshot service from his application or not. The major drawback of scale arc is, it is having poor identification of target groups and it is seamlessly providing target groups. Target group are the different media constraints such as images, web, videos, news, etc. While using scale arc user cannot redirect its control from one media to another media. User need to specify in which media he wants to work before using it. By controlling these entire Medias user may not immediately land on the media on which he wants to land.

VI. PROPOSED SYSTEM

The proposed system is software based so users need not to buy hard disks for large amount data storage purpose. In this paper we have demonstrated that how the proposed web application provides elastic load balancing and auto scaling for management of huge amount of database.

Our system architecture consists of four main modules. They are as follows

1. Potential growth servers.
2. Proxy servers.
3. Master slave
4. Cache memory

These modules define the function of our system. Our system is web application where user will enter into our system by creating his account by providing essential credentials. Then he will be logged into our system by providing username and password. As he is logged in our system he can see the dashboard for his personal account where he can upload his data, view data and can see how much data he has used and how much remaining storage which he can use in future.

In our system there is event monitoring happening which monitors all action performed by the user. User can retrieve his data or he can view his data .This type of functionality provided by CDM (Cloud domain function) by the system. The user's updated data will be get stored in cloud files. The system is having internet routing switch which is needed to land the user on load balancing function. In each region the system creates instances. The instance provides a platform for user's data storage purpose. Internet routing switch uses the IP address to land user data on load balancing functionality.

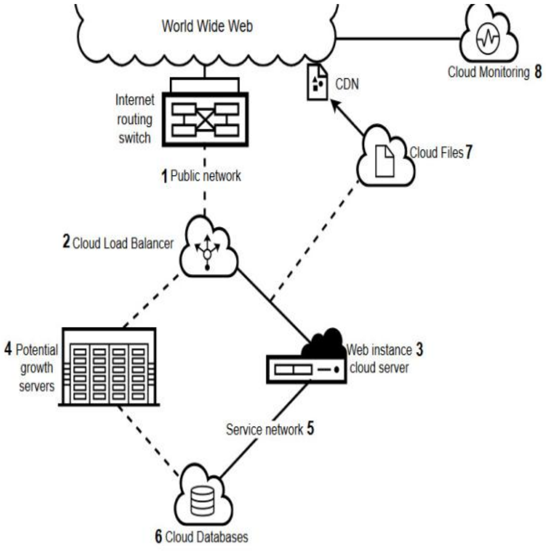

Fig.1 potential growth

The system provides load balancing function to balance the user's data. It also provides auto scaling function means according to user need and usage of memory. The storage will be provided to user by the system. The potential growth servers help the system to provide the dynamic allocation of storage to the user's data according to user need. Hence provides auto scaling functionality. Cloud database contain a data after every update on data performed by the user. This update is done through web application by user. All data related operations are managed through app instance in the system.

Second module of our system is reverse proxy .Reverse proxy chooses the area among various areas. The area will be chosen according free space available on it and users data. The data is temporary hold on one area and according to availability of storage on various areas the reverse proxy transfers the data to that area.

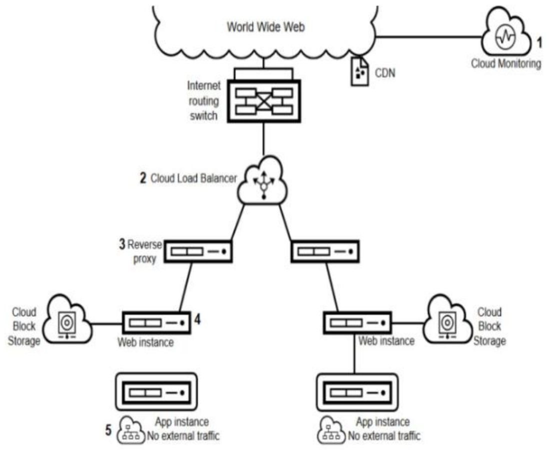

Fig.2 Reverse Proxy

The third module of the system is master and slave .This module uses cache instance to distribute the work among master and slave. If large numbers of users are performing same operations on their data at that time one query will be performed and cached to reduce burden on system. Instead of processing the same query every time, the query will get cached and single query is processed to avoid usage of many system calls. The cached query is handled by the master and rest of operations are distributed among the slaves.

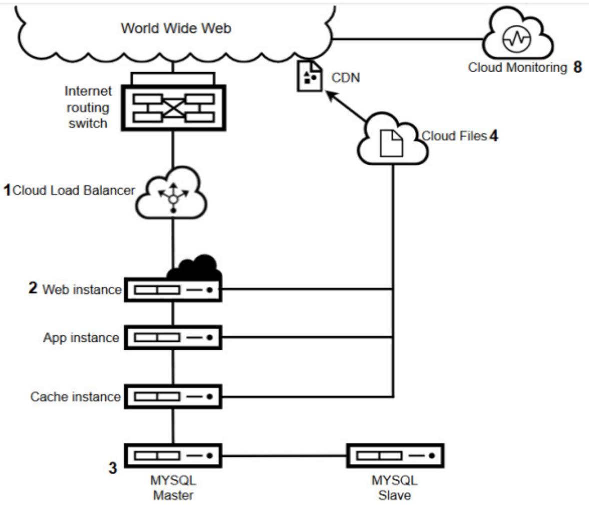

Fig.3 Master Slave

The last module of the system is memory cache. The memory cache contains various cached operations performed on data. It contains one block in which multiple block storage are provided. Block storage is assigned to each slave and each block storage space performs the same operation on different data. After all the operations performed on data, this data get stored on cloud database.

The updated data in memory cache can be retrieved or fetched in cloud files.

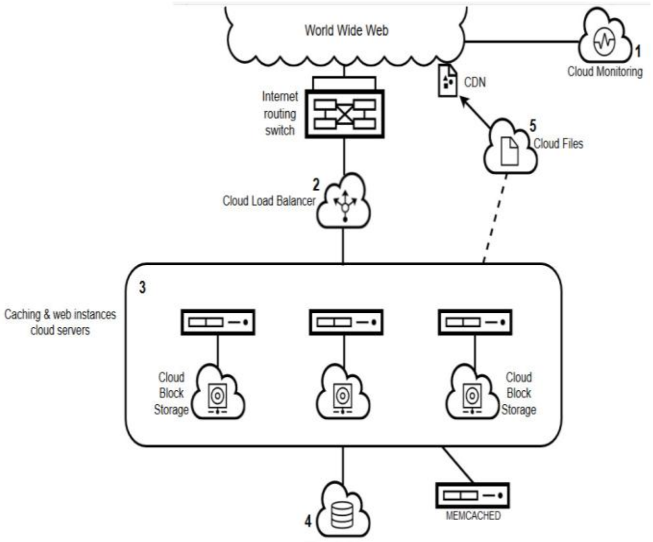

Fig.4 Memory Cache

VII. ADVANTAGES

- Auto scaling.

Elastic Load Balancing also offers integration with Auto Scaling, which assures that you have the back-end capacity available to meet varying traffic levels. Let's say that you want to make sure that the number of healthy EC2 instances behind an Elastic Load Balancer is never fewer than two. You can use Auto Scaling to set these conditions, and when Auto Scaling finds that a condition has been met, it automatically adds the requisite amount of EC2 instances to your Auto Scaling Group. Here's another example: If you want to make sure to add EC2 instances when the latency of any one of your instances exceeds 4 seconds over any 15 minute period, you can set that condition. Auto Scaling will take the appropriate action on your EC2 instances, even when running behind an Elastic Load Balancer. Auto Scaling works equally well for scaling EC2 instances whether you're using Elastic Load Balancing or not.Auto Scaling Groups are a wonderful service of cloud that allows us to automatically scale up or scale down our cluster of EC2 instances based on CPU or Memory consumption by using scaling policies.Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. The service provides a simple, powerful user interface that lets you build scaling plans for resources including instances and tasks, database tables and indexes, and Replicas. Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. With Auto Scaling, your applications always have the right resources at the right time. Auto Scaling lets you set target utilization levels for multiple recourses in a single, intuitive interface. Auto Scaling lets you build scaling plans that automate how groups of different resourses respond to changes in demand. You can optimize availability, costs, or a balance of both. Auto Scaling monitors your application and automatically adds or removes capacity from your resources groups in real time as demand change.

- Multi tenancy support.

Multi tenancy refers to software architecture in which a single instance of software runs on a server and serves multiple tenants. A tenant is a group of users who share common access with specific privileges to the software instance. With a multitenant architecture, a software application is designed to provide every tenant a dedicated share of the instance - including its data, configuration, user management, tenant individual functionality and non-functional properties. Multi tenancy contrasts with multi-instance architectures, where separate software instances operate on behalf of different tenants.Multi tenancy is for the benefit of the service provider so they can manage the resource utilization more efficiently, but multi-tenancy is not to the tenant's advantage at all.Multi tenancy" at the highest layer basically advocates a shared-DB approach.In a multi tenancy environment, multiple customers share the same application, running on the same operating system, on the same hardware, with the same data-storage mechanism. The distinction between the customers is achieved during application design, thus customers do not share or see each other's data. Compare this with virtualization where components are transformed, enabling each customer application to appear to run on a separate virtual machine. Multi tenancy allows for cost savings over and above the basic economies of scale achievable from consolidating IT resources into a single operation. . As there is a single software instance serving multiple tenants, an update on this instance may cause downtime for all tenants even if the update is requested and useful for only one tenant. Also, some bugs and issues resulted from applying the new release could manifest in other tenants' personalized view of the application. Because of possible downtime, the moment of applying the release may be restricted depending on time usage schedule of more than one tenant and hence zero downtime is use.

- Availability zone.

Cloud is composed of regions and availability zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as availability zones. EC2 provides you the ability to place resources, such as instances, and data in multiple locations.Resources aren't replicated across regions unless you do so specifically. Cloud Resources aren't replicated across regions unless you do so specifically. When you launch an instance, you can select an Availability Zone or let us choose one for you. If you distribute your instances across multiple Availability Zones and one instance fails, you can design your application so that an instance in another Availability Zone can handle requests.

You can also use Elastic IP addresses to mask the failure of an instance in one Availability Zone by rapidly remapping the address to an instance in another Availability Zone. For more information, see Elastic IP Addresses.

An Availability Zone is represented by a region code followed by a letter identifier; for example, us-east-1a. To ensure that resources are distributed across the Availability Zones for a region, we independently map Availability Zones to identifiers for each account. For example, your Availability Zone us-east-1a might not be the same location as us-east-1a for another account. There's no way for you to coordinate Availability Zones between accounts.

As Availability Zones grow over time, our ability to expand them can become constrained. If this happens, we might restrict you from launching an instance in a constrained Availability Zone unless you

- Zero downtime.

Zero downtime refers that website stays online during the whole process that means the system is made available 24*7 365 days.. Usually when you deploy without any strategy it implies downtime. Zero downtime describes a site without service interruption. Zero-downtime deploys - the ability to release a new version of your code to production without taking the site down - are key components of continuous delivery. Zero downtime, on the other hand increased productivity, higher revenue, and greater opportunities or it can be said its crucial to partner with a cloud computing provider that will give you peace of mind by keeping your business up and running at all times. It refers to a period of time that a system fails to provide or perform its primary function. Reliability, availability, recovery, and unavailability are related concepts.

VIII. APPLICATIONS

- Big enterprises.

Big enterprises required to store and manage their database and hence required load balancer to balance their data on database.

- Manufacturing.

Manufacturing organisation need to update and store the information of the product that is been manufactured is needed to be store and manage the database and hence required to balance their data by load balancer.

- Automobiles.

Automobiles organisation even required to store their content fo their product in database and need to maintain the database.

IX. CONCLUSION

In this paper, we are trying to overcome the drawbacks of existing similar systems. We have interpreted the aspect of load balancing which deals with distribution of incoming load of traffic on available regions. We are also dealing with the concept of auto scaling which scales the database size according to database requirement. Various algorithms have been reviewed for this service. These algorithms are not efficient in terms of memory requirement because user need to buy a new memory for scaling there existing data .This problem have been addressed in the proposed approach.

X. REFERENCES

[1]. Z. Gong, X. Gu, and X. Ma. Siglm: Signature-driven load management for cloud computing infrastructures. In Proc. IEEE International Conference on Quality of Service (IWQoS), Charleston, South Carolina, 2009

[2]. Ha'c and X. Jin. Dynamic load balancing in distributed system using a decentralized algorithm. In *Intl. Conf. on Distributed Computing Systems*, 1987.

[3]. Mahajan, K., & Dahiya, D., "A Cloud Based Deployment Framework For Load Balancing Policies" IEEE seventh International Conference on Contemporary Computing, pp. 565-570, August 2014.

[4]. Sharma, S., Singh, S., & Sharma, M. "Performance Analysis Of Load Balancing Algorithms" World Academy of Science, Engineering and Technology, 38, pp. 269-272, 2008.

[5]. Rahman, M., Iqbal, S., & Gao, J., "Load Balancer as a Service in Cloud Computing", IEEE 8th

[6]. Nuaimi, K. A., Mohamed, N., Nuaimi, M. A., & Al-Jaroodi, J., "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" IEEE second symposium on Network Cloud Computing, pp.137-142,December2012.