PROJECT ON

”ELASTIC CLOUD SERVICES”

PRODUCT DEVELOPMENT LABORATORY

BY

ASMITA CHIHNARA 115CS0204
DEBASIS PANDA 115CS0228

UNDER THE GUIDANCE OF
PROF. PABITRA MOHAN KHILAR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA , INDIA

2017-2018

ELASTIC CLOUD SERVICES

ASMITA CHIHNARA
DEBASIS PANDA

ABSTRACT

Cloud computing in other words refers to a model that provides a shared pool of configurable computing resources that dramatically simplifies infrastructure planning and can be released with minimum service provider interaction. The two main capabilities of cloud system for which it is being appreciated widely by consumers are ease of use and cost effectiveness. But loss of application data and data breaches are the recent headaches to be brought to the focus of the developers. Even load management of web traffic requests to make the entire system cost efficient with better reliability should be focused.

This paper implements both load balancing and fail over support system in order to make the system more robust plus to decrease the possibility of data loss using a mechanism that prioritises the user needs regarding the type of application data.

Keywords: : Virtual machines, Load balancing, Fail over support, Distributed Network, Application and database servers.

INTRODUCTION

Rapid development in technological spectrum has made computing resources powerful and more ubiquitously available than before. This has evolved the concept of cloud computing in which the resources including CPU and storage can be accessed by the user over the internet depending upon their demand [2]. There are several definitions of cloud computing but among them most accepted one is given as follows.

NIST(National Institute of Standards Technology) cloud computing definition: Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources(e.g. servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.[3]

The virtual storage capability of cloud services has made tremendous impact on IT sector. Large companies like Google, Amazon and Microsoft use cloud platforms to make their services more reliable. It is also greatly preferred by business professionals to run their applications on a cost-efficient platform with proper security from loss of data. Here our objective is to emphasize more on users choice for required services and simultaneously design the system with better security provisioning.

Chapter 1

1.1 Related Works

As our work mainly focuses on web traffic load balancing and fail over support system, we first discuss here about existing methodologies in this field and their limitations. Load balancing techniques are basically classified into two categories; Static and Dynamic load balancing.

- **Static Load Balancing:**
  As the word static suggests, this type of load balancing entirely depends on the information provided to the system prior to performance. And the performance is determined during execution. Finally all the results of work done by nodes are submitted to the remote node. Depending on the performances, load is distributed initially without considering current node. [5]
  This method has lesser execution time as compared to dynamic load balancing; however while making load allocation decisions, current state of the system is not taken into account. This affects overall performance of the system because of load fluctuation in distributed system.

- **Dynamic Load Balancing:**
  This algorithm continuously monitors the real time changes on work load and redistributes the work. It basically follows three strategies: transfer, location and information. [5]
  Transfer strategy decides which tasks are eligible for transfer to other nodes for processing. Location strategy decides a remote node to where the transfer is to be done. Finally information strategy works as an information centre for this algorithm.
  This method provides more reliability as it analyses the real time load, but the maintenance cost is very high (higher energy consumption) as compared to static load balancing. Again the centralised algorithms cause bottle neck in the system.

1.2 Motivation and Objective

As we have discussed earlier about the two main disadvantages of static and dynamic load balancing techniques, our objective here is to integrate both the models and propose a model that can eliminate the limitations of those two. So, here we propose a model that monitors real time work load as well as implements a static round robin algorithm to distribute web traffic uniformly that helps in cost-reduction.

Again based on the type of services provided by cloud, cloud computing is categorised into three different types [4].

- **Software as a Service (SaaS):**
  Here a single instance of the service runs on cloud server and multiple end users are serviced with a lower cost as only one application is hosted and maintained. E.g. Google, Salesforce, Microsoft, Zoho etc.
- **Platform as a Service (Paas):**
  Here an entire development environment is provided as a service upon which other higher levels of services can be built. Customers can build their own applications on providers platform. It also provides predefined combination of OS and application servers. E.g. Googles App Engine, Force.com etc.
- **Infrastructure as a Service (Iaas):**
  It provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure [4]. E.g. Amazon, GoGrid, 3 Tera etc.

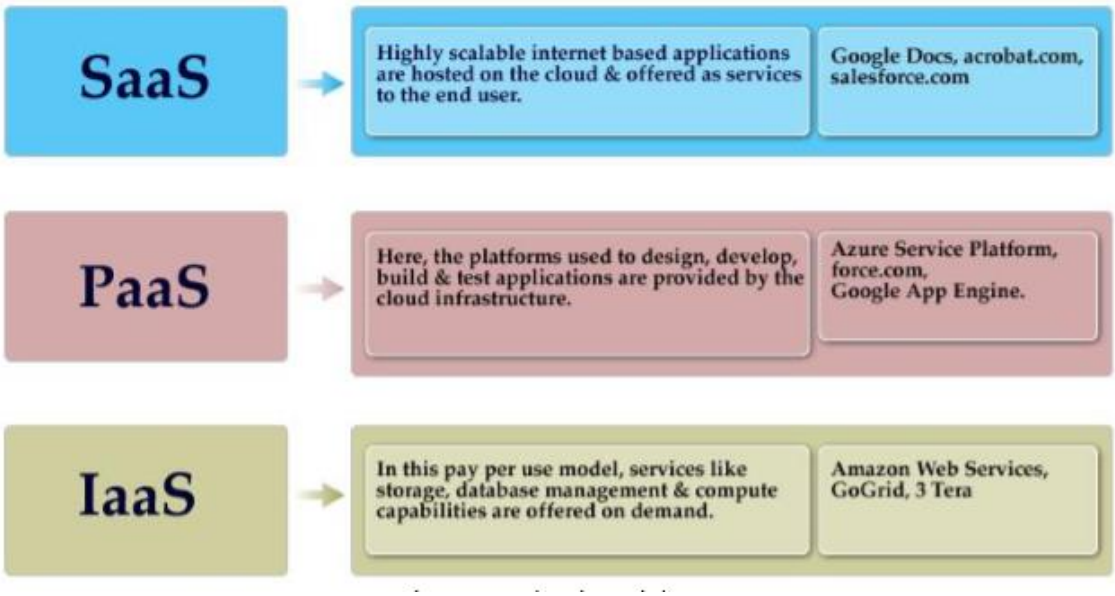

Figure 1.1: Categories of Cloud Computing

Our software implements the Iaas model. It has the feature of giving customer needs as the highest priority while deploying any application on the server. It provides different choices of application server and database servers in terms of their computation ability, storage space and privacy.

Chapter 2

2.1 Methodology

The overall system consists of a primary node where our software needs to be deployed and many secondary nodes connected to the primary node. The secondary nodes are distinguished in terms of four following categories:

- Shared application server and shared database server
- Shared application server and dedicated database server
- Dedicated application server and shared database server
- Dedicated application server and dedicated database server

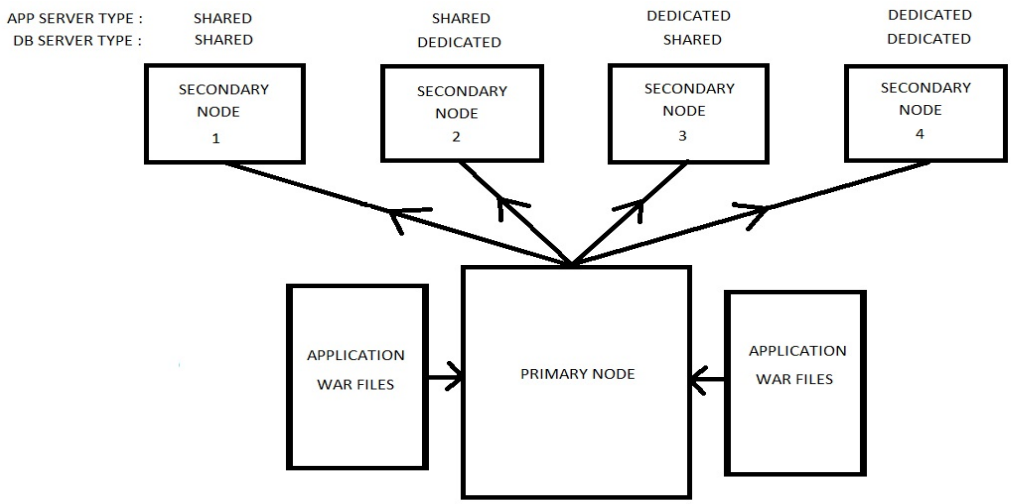

Figure 2.1: Schematic diagram of our system

Dedicated means that particular server is allocated to only one application where as in shared servers more than one application can be deployed. Backup of every application WAR file and SQL file is stored in the primary node.

There are two stages- Application onboarding and Customer onboarding. In application onboarding the setup file(WAR file) and the SQL file of the application is uploaded into the primary node. When a customer is onboarded in the next step a particular application is mapped to that customer and the application WAR file as well as its databases are deployed into a particular secondary node as per the customer demand of storage and computational power.

Our software implements round robin algorithm for load balancing of the web traffic requests. This algorithm distributes the client requests to each parallel node in turn and after reaching the end of node list, it iterates again from the first node.

Chapter 3

3.1 Results and Discussions

For the purpose of simulating the application, we have developed an interface in order to communicate between admin and user. In our database all the application details, customer details and mapping between customer to specific accesssible applications are stored.

Admin End:

- An admin can log in to his/her account by entering valid credentials.
- Now, admin gets the option to onboard application and to onboard user.

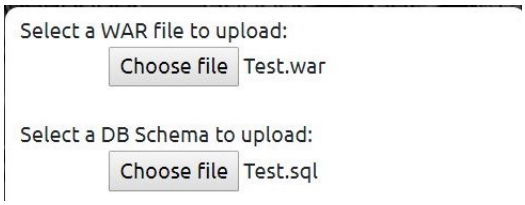

Figure 3.1:

- Admin has to select the WAR file for the application as well its SQL file with the name of the application. After submiting the page the files are uploaded onto the primary node.(Fig. 3.1)
- In customer onboarding procedure, the admin has to enter the details of the customer as well as the application it is to be mapped. The admin has to also select the app and database server type according to the customer demand and choose the corresponding node. (Fig. 3.2)

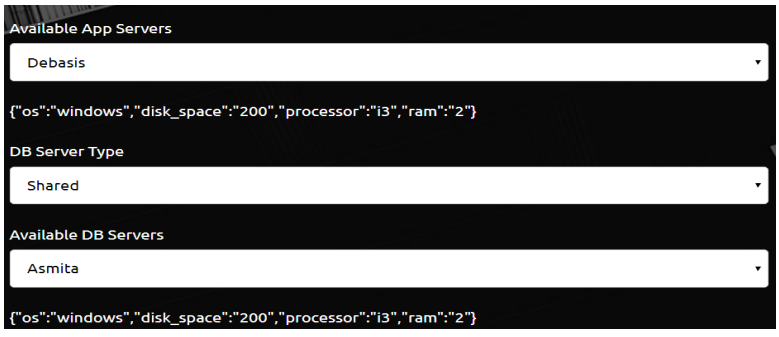

Figure 3.2:

Customer End:

- Each customer is given unique credentials using which he/she can log in to his/her account.
- In the customer homepage, all the applications allotted to him/her are display with a button to redirect to the corresponding site. (Fig. 3.3)

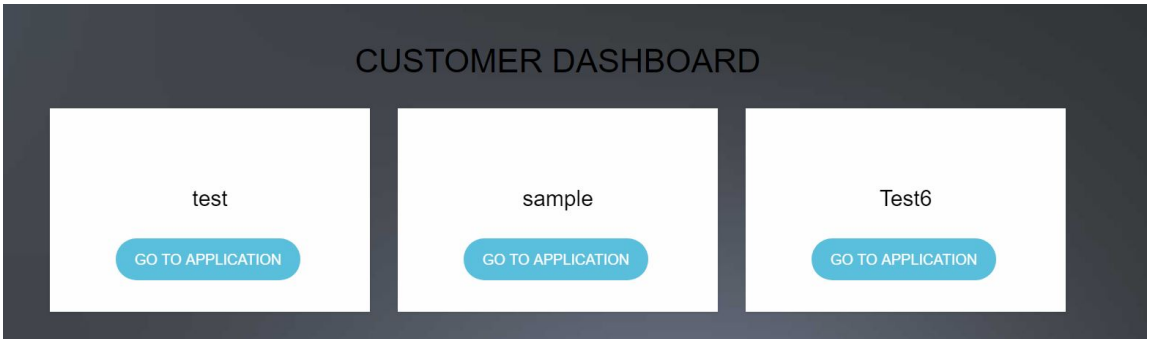

Figure 3.3:

- Upon clicking the specified application user wants to run, it executes the corresponding WAR file by fetching the details of the WAR file from the database of the primary node.

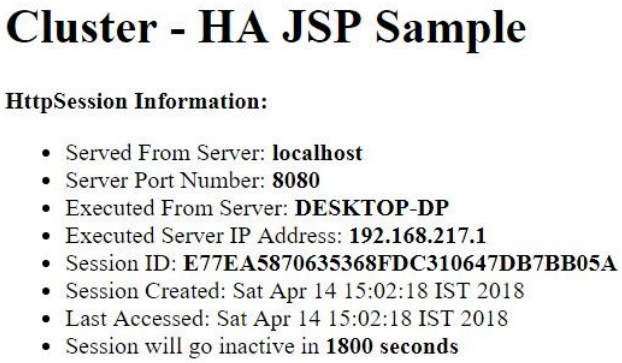

Figure 3.4:

LOAD BALANCING AND FAILOVER IMPLEMENTATION:

Using round robin algorithm, it distributes the client request to each available secondary node uniformly in turn. This satisfies our objective regarding load balancing. If any of the secondary nodes is not available or has failed then the system detects the failed server and does not redirects any client request to that node which implements failover support system as proposed above.Figure 3.5 shows the node status page. The available nodes are in green colour and the failed or unavailable nodes are in red colour.

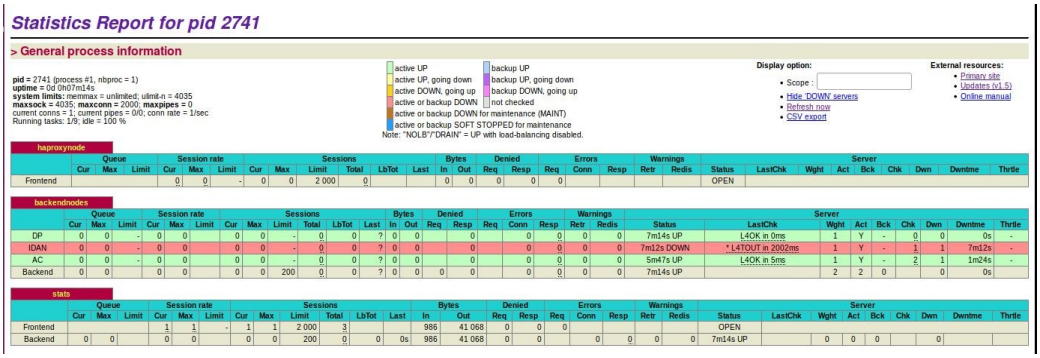

Figure 3.5:

CONCLUSION

In this paper we presented a model that runs static round robin algorithm yet takes real time web traffic load and redistributes the load. Thus it implements both the concepts of static and dynamic load balancing by eliminating their limitations. Again as it continuously checks the availability of the servers, during failure of one sub-node, it withdraws the request and assigns it to another available one. Thus it prevents loss of data due to server failure. As of now, there are no conflicts detected in the algorithm and its works efficiently.

In future, we aim to extend this project to implement newer and dynamic load balancing algorithms instead of static at the same time keeping it cost efficient.

REFERENCES

1 *Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, david Patterson, Ariel Rabkin, Ion Stoica and Matei Zaharia, A view of Cloud Computing in Communication of the ACM, 50-58, Vol 53, April 2010.*

2 *Qi Zhang-Lu Cheng-Raouf Boutaba, Cloud computing: state-of-the-art and research challenges, 25 February 2010.*

3 *Peter Mell(NIST), Tim Grance(NIST), The NIST Definition of Cloud Computing, publication : SP 800-145, September 2011.*

4 *Torry Harris, Cloud Computing - An Overview in www.thbs.com*

5 *Bhushan Ghutke and Urmila Shrawankar, Pros and Cons of Load Balancing Algorithms for Cloud Computing , March 2014.*