Survey On e-ticket Generation through Cloud Computing

P.Sreedevi1, M.V.Bramhananda Reddy2, A. Rama Mohan Reddy1, M.GaneshKumar4
1,4Dept. of CSE, Holymary institute of technology and science, Hyderabad, A.P. INDIA
2 Dept. of CSE, VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY, Proddatur, A.P., INDIA
3Dept. of CSE, S.V.U. College of Engineering, Sri Venkateswara University, Tirupati, INDIA

ABSTRACT

Service-oriented architecture (SOA) paradigm for planning large-scale distributed applications offers significant cost savings and minimizing hardware resources by reusing existing services. However, the high irregularity of client requests and the distributed nature of the approach may lead to service response time and availability. Static replication of components in data centers for accommodating load spikes requires proper resource planning and underutilizes the cloud infrastructure. Moreover, no service availability guarantees are offered in case of datacenter failures. In this paper, we propose a cost-efficient approach for dynamic and minimizing the hardware resources in a cloud computing infrastructure that effectively adapts to load variations and offers service availability guarantees. In our virtual economy, components rent server resources and replicate, according to self-optimizing strategies. We can prove it as cost efficient. We experimentally prove that such an approach outperforms in response time even full replication of the components in all servers, while offering service availability guarantees under failures. A successful online application should be able to handle traffic spikes and flash crowds efficiently.

Keywords—Distributed, Reliability, Service oriented, significant, Virtual Economy.

I. INTRODUCTION

Cloud computing is deemed to replace high capital expenses for infrastructure with lower operational ones for renting cloud resources on demand by the application providers. However, with static resource allocation, a cluster system would be likely to leave 50% of the hardware resources (i.e. CPU, memory, disk) idle, thus baring unnecessary operational expenses without any profit (i.e. negative value flows). Moreover, as clouds scale up, hardware failures of any type are unavoidable.

A successful online application should be able to handle traffic spikes and flash crowds efficiently. Moreover, the service provided by the application needs to be resilient to all kinds of failures (e.g. software sales, hardware, rack or even datacenter failures, etc.). A naive solution against load variations would be static over-provisioning of resources, which would result into resource underutilization for most of the time. Resource redundancy should be employed to increase service reliability and availability, yet in a cost effective way. Most importantly, as the size of the cloud increases its administrative overhead becomes unmanageable. The cloud resources for an application should be self managed and adaptive to load variations or failures.

II. DISTRIBUTED APPLICATIONS USING DIFFERENT COMPONENTS

Figure2.1: A Distributed application using different components Building an application that both provide robust guarantees against failures (hardware, network, etc.) and handles dynamically a load spike is a non-trivial task. We have developed a simple web application for selling e-tickets (print@home). Composed by 4 independent components (i.e.) web front-end, user manager, ticket manager, e-ticket generator.

Customer Registration

Customer has to enclose their details into the server. In this page several fields are mentioned name, e-mail id, phone number etc and also to provide card
details are available and visa card, master card , one more advantage is expiry date and cvc no standard for card verification number in this cvc no is checking their card details.

The main advantage login as customer they can select their ticket details. But each ticket details are identified by one secrete key. Based on that secret key it will process.

**User Manager**
Ticket manager  The user manager for managing the profile of the customers ,the profiles are stored in highly scalable ,distributed, structured key value User manager login they will monitor how many number of users are requested the ticket We Consider many independent components that interact together to provide you service to the end user as SOA.

In this page session is set for 60 Seconds. It beyond 60 seconds session will be Expired Distributed Optimization Algorithm are used. It maintains the all users’ profile, also gossiping.

**Ticket Manager**
The ticket manager for managing the amount of available tickets of an event. In this application to maintain how many member s are ticket booking .Store user profile and all profiles are maintain using one of the control is grid view control to visible in all booking details.

Grid view is store multiple records and retrieve thru database in this control one more advantage is paging and modification ,update, delete operation are applied.

**E-ticket Generator**
An e-ticket generator that produces e-tickets in PDF format, it will generate automatically all details in report format. And how many members ticket sanctioned to main all ticket details in this module using grid view control.

In this  Grid view is store multiple records and retrieve thru database in this control one more advantage is paging and modification ,update, delete operation are applied.

**III. ALGORITHM**

**Algorithm 1. Compute Dynamic Key**

1: ComputeDynamicKey($E_{sec}$, $ID_{txt}$)
2: begin
3: $j \leftarrow txID_{txt}$
4: if $j = 1$ then
5: $K_j \leftarrow F(E_{init}, IV)$
6: else
7: $K_j \leftarrow F(K_{j-1}, E_{sec})$
8: end if
9: return $K_j$
10: end

**IV. IMPLEMENTATION**

![Image](image_url)

**Description:** In this registration form it contains the fields like name, city, DOB, email-id etc..., after click on register button username, password generates to login into the website. And secret key is generated to maintain authentication. We can also get secret key into the mobile with the help of nokia pc suite.
Description: After reserving the ticket the user can check the details of ticket by using their secret transaction number, which is generated in the registration form.

Figure 3.2: e-Ticket Generator Secret Transaction Number Entry

Figure 3.3: e-Ticketing Service Electronic Reservation Slip

Description: In this form, after reserving the ticket one electronic reservation slip will be generated. User has to maintain this ticket along with ID proof original in the travelling.

V. CONCLUSION

A web front-end, which is the entry point of the application and serves the HTML pages to the end user. A user manager for managing the profiles of the customers. The profiles are stored in a highly scalable, eventually consistent, distributed, structured key-value store. A ticket manager for managing the amount of available tickets of an event. This component uses a relational database management system. An e-ticket generator that produces e-tickets in PDF format (print home).

The main advantage login as customer they can select their ticket details. But each ticket details are identified by one secrete key. Based on that secrete key it will process

This project is used to manage efficiently with less cost by using secrete keys. Every transaction is identified based on this key only.
The complete communication between customer, user manager, ticket manager and e-ticket generator with centralized server. Each component can be regarded as a stateless, standalone and self-contained web service. Figure 1 depicts the application architecture. A token (or a session ID) is assigned to each customer’s browser by the web front-end and is passed to each component along with the requests.

FUTURE ENHANCEMENT

We will continue to research on security mechanisms that support: to maintain Data securable and highly distributed data are the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

Cloud computing providing unlimited infrastructure to store and execute customer data and program. As customers you do not need to own the infrastructure, they are merely accessing or renting; they can forego capital expenditure and consume resources as a service, paying instead for what they use.

REFERENCES

[1]. User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
[2]. Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
[4]. Data Communications and Networking, by Behrouz A Forouzan.