Design and Realization of Mobile Message Board System Based on Cloud Computing*

Qi Shen[1], Yan Tang[1], Meng Zhang[1] and Qingming Song[1]

[1]School of Software Engineering Beijing University of Technology Beijing, China
shenq@bjut.edu.cn, marie_t@sina.com

Abstract - This paper presents the solution to Google App Engine using in mobile cloud computing, based on the analysis of key issues in the mobile message board system which combined with related technology of mobile Internet and cloud computing. And on this basis, the design, development and deployment of a Mobile Web message board system using Google App Engine has been done. It achieves the integration of mobile Web applications and cloud computing**.**

Index Terms - mobile cloud computing; GAE; WAP2.0

1. Introduction

With the rapid development of smart phones, instead of using computer to access the Internet, people have been used to use mobile phone or other mobile terminal equipment to access the Internet and get more information. Message board system has been developed from the Internet to the mobile Internet, which further demonstrates its convenient features, meet users' demands for whenever and wherever possible to use message board system.

The combination of mobile Internet and cloud computing not only makes up for the deficiency of the mobile terminal in the computing and storage capacity, but also makes cloud computing deliver services by the huge number of mobile terminals. Therefore, the combination of mobile Internet and cloud computing will become the development trend of the message board system.

In addition, because the data of traditional Web application is usually stored centrally on a server, once the server goes wrong, the service will not be used even the data will be severely damaged and can not to be restored. But the cloud computing service can timely recover from failure due to its characteristics which store the application and data in server cluster

2. Key Problems of Mobile Message Board System

Although the combination of mobile Internet and cloud computing make up well for each other's deficiencies, viewed from the technology, the juvenility of the cloud computing and the limitation of mobile terminal hardware must cause a variety of troubles for the system design and development.

First of all, the mobile terminal browser usually only supports WAP protocol, can not access the HTML page, and can not display too much content due to the screen size limitations. So how to write a WAP page with strong performance and good visual effect will directly affect the user experience.

Secondly, it is complicated to store the data in a scalable network application[1,2]. Based on cloud computing, the message board system data will be distributed over multiple machines. So the user can communicate with any number of network servers at the same time. And it can not accurately predict which network server will processes the user's next request[3,4]. Therefore, all network servers are possible to interact with any machines stored needed data.

Thirdly, in a cloud computing environment, processes of different applications may be run on the same host simultaneously. In order to avoid interference and competition for resources between processes, all applications will be run in a restricted “sandboxed” environment. It is independent of network server hardware, operating system and physical location. Therefore, in the “sandboxed”, mobile message board system will not directly perform operations, such as to write to the file system, to open Socket or to access to other hosts.

Finally, how to deploy the mobile message board system in the cloud computing server cluster and how to manage and maintain the system is also different with traditional applications.

3. Realization of Mobile Message Board System

The main functions of the system include registration or login, message or reply, personal center, management center and information search. New users need to create your own account through the registration firstly. And after login, users can leave or reply message. In personal center, all users can maintain personal information and view own message. Administrator can manage all messages and users' authority through the management center. In addition, there is search message function.

A. Interface Design

WAP (Wireless Application Protocol) is the most popular wireless application protocol. It makes the mobile Internet has a common standard. And its goal is introducing the Internet's wealth of information and advanced services into mobile phones and other wireless terminals. The design of WAP protocol is using the idea of Internet protocol stack for reference with modification and simplification, which can be

* This work is supported by Scientific Research Project of Beijing Municipal Commission of Education Grant #KM201210005030.

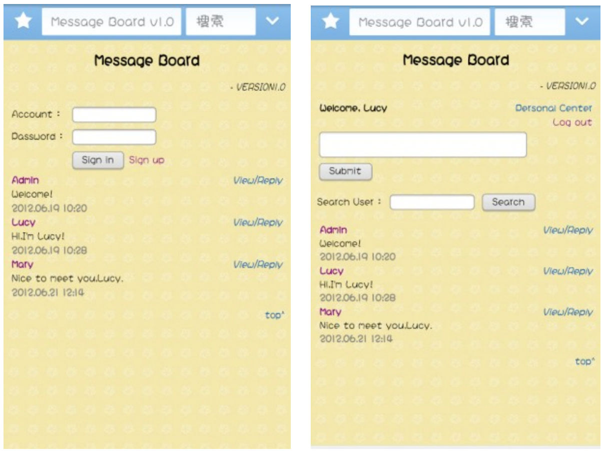
effectively applied to wireless application environment. Pages of the system use the XHTML MP (XHTML Mobile Profile), which is customized by WAP2.0 protocol. Compared with the traditional WAP site, it has been greatly improved in the demonstrated ability, and makes pages have more abundant presentation.

Pages of the system use XHTML MP, and combine with the JSP script language. Part of page declaration is shown as follows:

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.0//EN"
"http://www.wapforum.org/DTD/xhtml-mobile10.dtd">
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
```

In order to save the user input data, a web form is used in the page to receive user input firstly. And then, a Servlet Class is needed to handle the form data. Finally, it is necessary to write the mapping between URL and Servlet Class into the "web.xml" configuration file. When GAE receives a Web request, according to a configuration file, it will call the appropriate servlet and accept the corresponding data.

Fig. 1 respectively shows the interfaces of users with login and logout.



Admin
Welcome!
2012.06.19 10:20
Lucy
Hi.I'm Lucy!
2012.06.19 10:28
Mary
Nice to meet you.Lucy.
2012.06.21 12:14

(a) (b)

Fig. 1 Interface of User: (a)Before Login. (b)After Login.

B. Google Datastore

GAE uses the Google Datastore to store the application data. It solves problems of data distribution, replication and load balancing for developers, and it also provides powerful functions of query engine and transaction[5]. GAE Java SDK provides standardized API: JDO (Java Data Object) and JPA (Java Persistence API), which are provided by DataNucleus Access Platform and provide GAE Datastore adapter. In contrast, JPA is newer, but GAE support JDO more comprehensive. So the mobile message board system will use JDO to implement data access. In the directory "war/ WEB-INF/ classes/ META-INF/", the configuration file "jdoconfig.xml" is used to indicate using Datastore as JDO implementation.

1) JDO

By using JDO, POJO (Plain Old Java Objects) can be stored in any data store. It only need tell JDO how to store and reconstruction class instance[5].

In the mobile message board system, the class "Message" represents messages which users leave. It is annotated by @PersistenceCapable (detachable = "true") to mean that it can be persistent. And it uses the class name as the object name in the data store. The parameter (detachable = "true") means it can be evicted after read from the data store. Six member variables are defined in the class, as shown in Fig. 2.

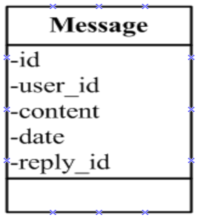


Fig. 2 The Message Class Structure.

- "id" is the primary key.
- "user_id" is user ID released message.
- "content" is the message content.
- "date" is release date.
- "reply_id" is the message ID which it replies.

By using "@Persistent" annotation, these variables as the object attributes are stored in the data store. In addition, the primary key "id" is also need to use the "@PrimaryKey" annotation, and by setting the parameter "(valueStrategy = IdGeneratorStrategy.IDENTITY)" declare that data store automatically assign the primary key value for entity when it be stored.

2) Operations of Google Datastore

Applications through the "PersistenceManager" object to perform operations of Datastore. And the object is created by the "PersistenceManagerFactory" object. The Factory Class instance needs some time to initialize. And the application requires only one Factory Class instance, so this instance can be stored in a static variable for multiple requests and multiple classes without initialized repeatedly.

The persistence operation can be performed with the configuration files and Java Data Objects. At first, call the constructor to create a Message instance, and create a PersistenceManager object using the Factory Class static instance. And then, passing the Message instance to the "makePersistent ()" method can store the instance in Datastore. And for the persistent instance, call the "deletePersistent ()" method can delete it from Datastore. Finally, regardless of success or failure of the operation, it is necessary to close the PersistenceManager object. In order to make query results can still be read after the PersistenceManager object is closed, it is necessary to call the “detachCopy ()” method to evict it from the entity in Datastore.

JDO standard defines a query mechanism called JDOQL, which is similar to SQL query language. It can query entities which meet the specified conditions set in the Datastore, and return results sorting by the specify order.

3) Mail Service

In the “sandboxed”, the application can not directly perform operations, such as to write to the file system, to open Socket or to access to other hosts. Therefore, GAE provides three ways to achieve applications communicate with outside world. They are mail service, URL Fetch and instant message.

When a user replies to a message, the application will send a mail to the user through the mail service. The mail content includes URL of the message, the reply user name, the reply date and so on. Mail service Java programming interface supports the use of JavaMail standard interface to send E-mail.

To send an email, a MimeMessage object must be created firstly, which is used to set the mail information, such as sender, recipient, subject, content, and any other related information. Then, call the “send ()” method of the “Transport” class to start the task of sending E-mail. It will be sent by asynchronous mode.

The “setFrom ()” method of MimeMessage object is used to set the sender of the message, and can provide any string as the sender's name; “setRecipient ()” method is used to set the recipient; “setSubject ()” method is used to set the subject; “setText ()” method is used to set the plain text message body. The sender and recipient need use InternetAddress class. Its constructor receives a string representation of E-mail address. If it is not a valid E-mail address, it will generate AddressException.

4) Application Deployment

Using the graphical management console provided by the GAE platform, it is easy to create and manage the application[6]. First, register ID for the application, and modify the configuration file “appengine-web.xml”. The value of the <application> element will be revised as the application ID. And then upload and deploy the application to GAE by using Eclipse plug-in or SDK command line. GAE will automatically run the application and provide hosting services. Google provides free appspot.com domain name to access the application. The URL is “http://application-id.appspot.com/”, and replaces the application-id with the registered application ID. It also can apply any other domain name for the application to bind.

4. Conclusion

By using standard Java network technology in Eclipse development environment, this paper realizes a mobile message board system based on cloud computing on GAE. Users use mobile terminals to connect to the Internet, and send Web requests to GAE through browsers. According to the requests, GAE processes data after receives them. Finally, results are returned to mobile terminals, and display to users by browsers. The system interface is programmed by using XHTML MP combining with JSP language. Data storage uses Google Datastore. Application, in the “sandboxed”, uses GAE mail service to communicate, and is deployed to GAE by using application hosting service.

The system, which achieves the integration of mobile Web application and cloud computing, has been designed, developed and deployed to GAE. By using the GAE services, this paper analyzes and solves the problems of page display, data storage, interaction with hosts and application deployment.

5. Acknowledgment

This work is supported by Scientific Research Project of Beijing Municipal Commission of Education (Grant No. KM201210005030), the support is gratefully acknowledged.

6. References

[1] Shu-Qing, ZENG, and Xu Jie-Bin, “The Improvement of Paas Platform,” In *The First International Conference on Networking and Distributed Computing,* China, 2010.

[2] Apache, “Welcome to Apache Hadoop,” http://hadoop.apache.org/, 2011

[3] R, LAMMEL, “Google's Mapreduce Programming Model-Revisited, ” 2010

[4] I, MUTTIK, and BARTON C, “Cloud Security Technologies,” Information Security Technical Report 14, no. 1, pp. 1-06, 2009.

[5] Yin, Huibin, Jun Han, Jing Liu, and Xu Hongyun, “Development and Research of Multimedia Courseware Sharing Platform Based On Gae,” In *2010 International Conference on E-Health Networking,Digital Ecosystems and Techonlogies*, China, 2010.

[6] Yin, Huibin, Jun Han, Jing Liu, and Jing Dong, “The Application Research of Gae On E-learning——Taking Google Cloudcourse for Example,” In *2011 IEEE International Conference on Information and Education Technology(ICIET 2011)*, China, 2011.