The Design and Implementation of a Mobile Distributed Web Server System

Timothy K. Shih and Yun-Chung Lu
Department of Computer Science and Information Engineering
Tamkang University
Tamsui, Taipei Hsien, Taiwan 251, ROC
E-mail: tshih@cs.tku.edu.tw

Jason C. Hung
Department of Information Management
Kuang Wu Institute of Technology
Peitou, Taipei, Taiwan, ROC
E-mail: jhung@cs.tku.edu.tw

Keywords: Mobile Distributed Web Server System, Web Content Requesting Imbalance Load Balancing , Network

Abstract

The Internet traffic is growing rapidly in recent year. How to distribute the network traffic effectively is a important issue. In this paper we propose a Mobile Distributed Web Server System. This system can distribute web traffic to the mobile servers around the world in cheap and effective way. It is fully compatible with existing systems. We also provide a possible implementation of a Mobile Distributed Web Server System. In this paper, we discuss the Web Content Requesting Imbalance (WCRI) problem and describe how this system is suitable for this problem.

1. Introduction

The requirement of quality of service (QoS) over the Internet is higher and higher.The growing rate of bandwidth is slower than the consuming rate relatively. However, the Internet multimedia service makes this imbalance situation more serious. Transferring multimedia content over Internet requires high and stable bandwidth to achieve good quality. This makes web sites only provide limited multimedia service despite of multimedia content is more attractive than static ones (i.e. HTML pages).

Another, like transferring large or popular files over the Internet, is similar to this situation. Besides these problems, there is another imbalance situation on the web server,

named Web Content Requesting Imbalance (WCRI). There are some popular web contents, including HTML pages, files, etc. For example, if web content is up-to-date, it will be very popular for a period of time. The web site that contains news web pages is another example, since the latest news will be request frequently by users. The problem describe previously is more serious for large (popular) web site.

In order to solve these problems, researchers had proposes a distributed web server system. Basically, this idea uses multiple web servers to share client requests. However, load balancing is another important issue. Many approaches have been proposed [5,6]. Generally speaking, there are four kinds of dynamic load balancing methods on web server system, including Client-based approach, DNS (Domain Name Service)-based approach, Dispatcher-based approach, and Server-based approach .[5] these approaches use hardware or software redirection method to share web client requests to web servers.

These approaches have the same problem about flexibility. All approaches are not compatible to current systems (i.e. DNS, web browser, web server, etc) due to the modification of software or hardware. This will lead to high constructing cost, low generality, low popularity, and most important of all, low flexibility. Because these approaches lack flexibility, they don't work well in some situations. For example, if we want to deal with Web Content Requesting Imbalance (WCRI), these approaches are not suitable. If we want to redirect client requests that request popular web contents instead of all client requests, but these approaches are designed for redirecting all client requests. To solve this problem, we propose a system call mobile distributed web server system. In our system, no software or hardware modification need and this system can provide flexibility since it is a pure software system. This will also reduce the cost since it can be constructed by existing systems. And this system can make use of on-line PCs around the world.

In this paper, we describe our system the as follows: Section 2 we describe our system architecture. In section 3, we discuss the redirecting algorithm in this system. We describe some important system issues in section 4. We discuss our implementation in section 5.and we conclude our system in Section 6.

2. System architecture

There are five components of our system: host server, mobile server, database server, communication server, and management tools.Figure 1 give a basic architecture of mobile distributed web server system. In figure 1,we can see there are two sides of servers in mobile distributed web server system.One is host server side; the other is mobile server side. There are three kinds of server in host server side, host server, communication server, and database server. On the other hand, there are only mobile servers in mobile server side. All servers in host server side are on the Local Area Network (LAN). Mobile servers are connected on Wide Area Network (WAN) with host server side. Mobile servers are connected each other on LAN or WAN. In the following section, we will discuss each server in detail. Then we will discuss the message model in this system.

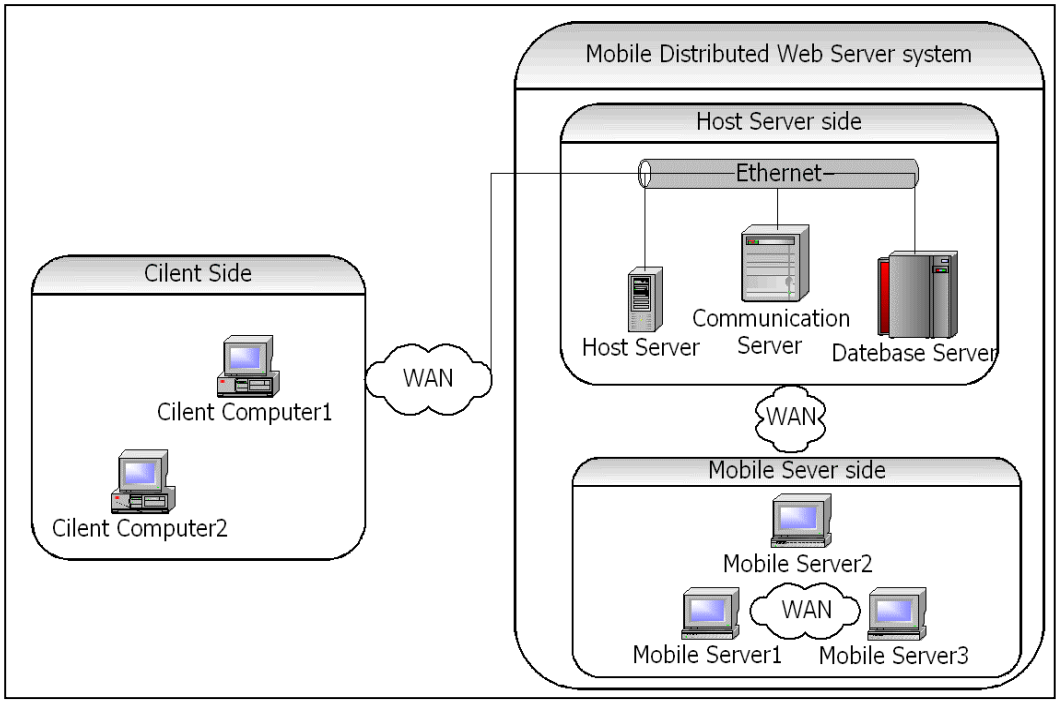

Figure1.System Architecture

2.1. Host server:

Host server is basically a web server with ability to access database server via server script pages, for architecture transparency [5], all client must connect to host server first before redirection to other mobile servers, and if the client send next request (i.e.: click a hyperlink on HTML pages), it will be link to host server for redirection. This process will continue until client exits this website. We monitor host server status and store this information in database server. We monitor host server's CPU, memory, and network utilization.Host server consists of the following components:

- **Web server with supporting server scripting language**

Web server is an ordinary web server, which can support server scripts (for example, ASP or JSP). We use HTTP redirection [13] to perform the redirection process.

- **Service program**

We use service program to perform the following tasks:

  - Monitor host server status
  - Access database server
  - Receive/send control messages
  - Receive/send specialized web contents

- **Specialized web content (i.e. files, HTML pages ,etc) distributed to all mobile servers**

Our distributed mobile web server system implementation uses some specialized web content. Because we need redirecting every client's request, all web pages in our system are server script pages. These specialized web pages can be modified from normal web pages easily. The main difference between specialized web pages and normal web pages is the internal hyperlinks in web pages. Internal hyperlinks are hyperlinks referred to the website which this

web pages belongs to. These internal links need to be modified so that they can link to a special server scripting pages. We call this server script page as Selector. All Internal hyperlinks must link to this Selector for next possible redirection. Next possible redirection occurs when client clicks hyperlink on the web pages. When users click these internal hyperlinks, they will be redirecting to one mobile server that has the web content base on redirecting algorithm in selector. The other web contents, for example: files, don't need modification since it contains no hyperlink.

2.2. Mobile servers

Mobile servers are web servers, which hold web contents transferred by communication server. They can set up through web browsers. Once setting up, they can response to client's requests, which redirect by host server. Furthermore, remote communication server can control these mobile servers and minimize the local maintain cost. Mobile server contain the following components:

- **Service program**

This Service program is similar to Service program of host server, except it cannot access database. There are three reasons why we don't let this Service program direct access database server. First reason is simplicity. We hope the structure of mobile server as simple as possible. This can reduce the transfer and maintenance cost. Second reason is transparency. In this approach, this Service program doesn't need to know information about database (i.e. location, database structure, password etc). We can modify database server without change mobile server's service program. We only need modify host server side software to reflect this modification, which is an easier and cheaper choice. The third reason is security. Since mobile server side doesn't know the information of database server, this can avoid malicious intention on database server. We use message-passing approach to communicate between host server side and mobile server side. Whenever mobile server need communicate with host server side , it can sent message to communication server and let communication server decide next step base on the message of mobile server. This indirect message-passing way can avoid drawbacks describe above (simplicity, transparency, and security issues).

- **Distributed web contents (i.e. files, HTML pages etc)**

Distributed web contents are the same to host server. Distributed web contents in mobile servers can be partial or full mirror of website. Information about these distributed web contents should be stored in organized form for efficiency querying. In our system, we store this information in database server.

- **Web server with supporting scripting language.**

Web server is a general web server. But this web server has to be the same to host server side. There are two main reasons: first reason is simplicity. If mobile server side's web server is different with host server side, we must maintain several versions of web contents since the server script languages between these web servers are different. It is an inefficient choice. Second reason is compatibility. Even we maintain several version of web contents, we still have to face the compatibility problem. Since not all web servers are compatible with each other, we have two choices: design a new web server or choose one web server. Clearly, the letter choice

is better in cost and stability and most important of all, compatibility.

2.3. Database server

In our system, we use relational database server to ménage information.We use database store the following information:

- Mobile server information
- Host server information
- Communication server information
- Web content information
- Redirection information
- User information
- Other information

Database server consists of the following components:

- **Database system**

This database has no special requirements and no modification needed. This means simplicity and generality, which is important to developers and administrators of this system. In our system, database has an important role since all redirection must query this database. This database's loading may be relative high. Database Optimization can ease this situation and this issue is beyond this paper. We take several actions in order to achieve database higher performance. One of these actions is keeping the database small

- **Service program**

Besides database optimization, we need a service program to monitor whole database server in order to take proper response when overloading or crashing occurs.This service program has the following functionalities:

  - Monitor database server status
  - Receive/send control messages

3.4 Communication server:

Communication server is responsible for communicating between host server and mobile server side. When it receives messages form mobile servers, and performs proper action to reflect this message (i.e. update database). It can send massage to particular mobile servers (i.e. check mobile server status). It has a service program to perform operations describing above. This service program should perform the following functionalities:

- Monitor communication server status
- Access database server
- Receive/send control messages
- Receive/send specialized web contents

3.5. Management tools:

In order to manage this system, we need develop some management tools. One is maintain tool that can synchronize the whole web contents in this system when update occurs. It work as follows: first web administrator can decide what web content need updating via our tool, then it will search database for every web content information (i.e. web content location). After gathering information, it sends every updated content to specified mobile server and completes this process. The other tool is a report and analysis tool. It can generate reports like usage report or statistics report to help web administrator make decisions (i.e. decide which content is heavily used and it need more mobile server to share loading).

3. Redirecting algorithm

Since we want to select proper mobile server and redirect user's request to this selected server. We propose a redirecting algorithm to perform this task. The goal of this redirecting algorithm is selecting a mobile server that

geographical location is close to web user and this server's status is available. The whole process can be divided to two parts: determining user location process and selecting mobile server process.

In determine user location process, we use IP address mapping method. First we build two databases. One database is used for IP-to-country code mapping, and the other is used for storing an order country list for every region. We call this list as Candidate List (CL). We call first database as Location-Mapping Database (LMB) and the second database as Candidate-Selecting Database (CSB). The main functionality of LMB is mapping one IP address to one country code that this IP belongs to. We can use a IP-address as a key to query LMB and get the country code of this IP address. With properly design; the number of records in LMB is approximately several thousands. This size is small for modern database. Small number of database records can help this system more efficiency and ease the maintenance work. For example, if we query this database with a IP address 163.13.127.8, we will get the country code TW (Taiwan).

Beside LMB, we build another database, CSB. The main functionality of CSB is providing web clients and host server with the location and number of mobile servers in order. Host server can use this message for choosing a geographically closer mobile server. The network distance between two points is proportion to the geographical distance. We can predefine the Candidate List (CL) for each region around the world and dynamic adjust it according to network condition in client and server side. The CL consists of a preferring country list and a number of total mobile servers in this region. For example, the CL for Taiwan region can be defined as TW (2), US (5), JP (6), HK (1)… this means that there two available mobile servers in Taiwan, five available mobile servers in United States, and so on. And the order of CL means that we prefer the mobile servers in Taiwan to serve the web users in Taiwan, and we prefer the mobile servers in United States to serve the web users in Taiwan, and so on. The number of records in CSB is approximately hundreds of records. This size is so small that we even do not need to use database server to store these data.

After building LMB and CSB, we can perform selecting mobile server process. The redirecting algorithm runs as follows. First, when client access homepage of our website for the first time, we will perform the IP-to-country mapping by querying LMB using the IP address of the client as a key. After querying, we can get the country code of this client. Then query the CSB for the Candidate List(CL) of this region.

We store this information in client side cookies. The reason we use cookies is that we can do this process only when cookie expired and saving computing power of host server. There is another advantage for using cookie: we don't need perform this process for every client's request. When client send request, we can choose a set of closer mobile servers (for example, 10% of all mobile servers) according to the order of CL. We randomly select one mobile server these selected mobile server. The reason we make a random choice is that we want the distribution of this algorithm can uniform enough to prevent the least-load problem [5,6]. If the number of selected mobile

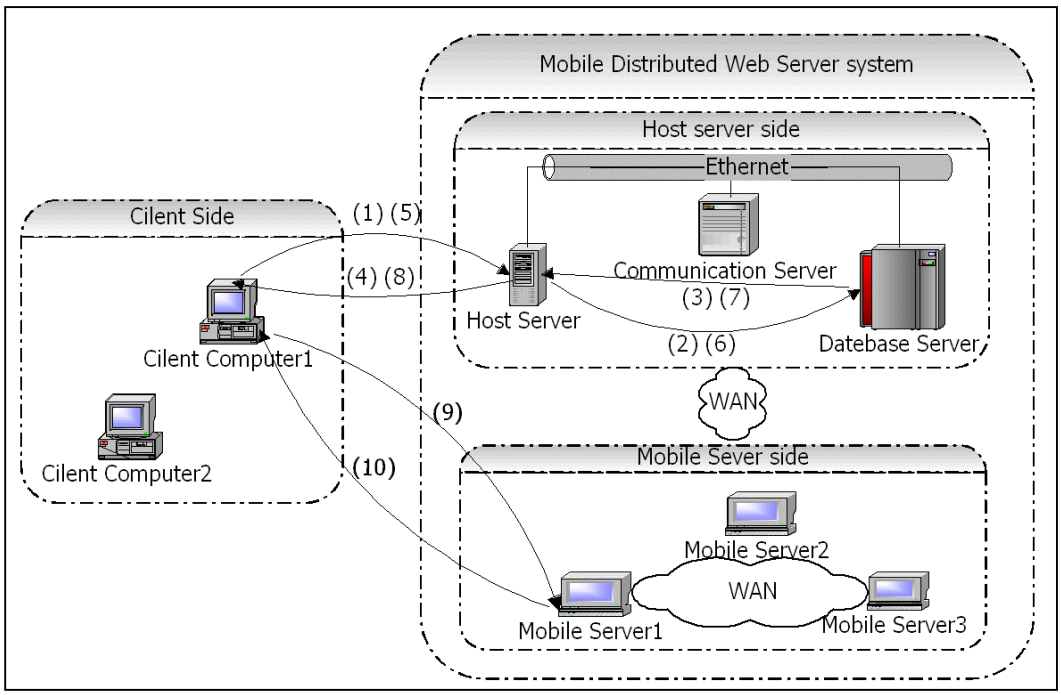

Figure2.The redirecting algorithm

server is not enough, we use host server to response this request.

Figure 2 shows the flow of this redirecting algorithm. Step (1) to (4) is the determining user location process and step (5) to (10) is selecting mobile server process. LMB and CSB are in database server.

In step (1), client computer1 accesses our website for the first time. When the host server receives this request, it queries LMB and CSB for the IP-to-country code mapping and CL (step (2) and step (3)). Then host server sends IP-to-country code mapping and CL back to client computer1 in step (4). The client computer 1 stores this information in its own cookies. In step (5), client computer1 sends another request and receive by host server. Host server then access the country code of client computer1 and query the CSB and get the CL of the region of client computer1 (step (6) and step (7)). The host server select a mobile server base on the redirecting algorithm describe above and perform a HTTP redirection (step (8)). After redirecting, client computer1 connect mobile server1 and mobile server1 response to client computer1 (step (9) and step (10)).

4. Important System Issues

In our system, there are some issues that need to deal with. These issues are reliability (availability) issue and security issue. We will discuss them in following sections. In reliability issue, we discuss that how to keep client's request will not redirected to an unavailable server. In security issue, we discuss that how to keep the web content of a mobile server form being suffered from malicious action (such as unauthorized deletion).

4.1. Reliability (availability) issues

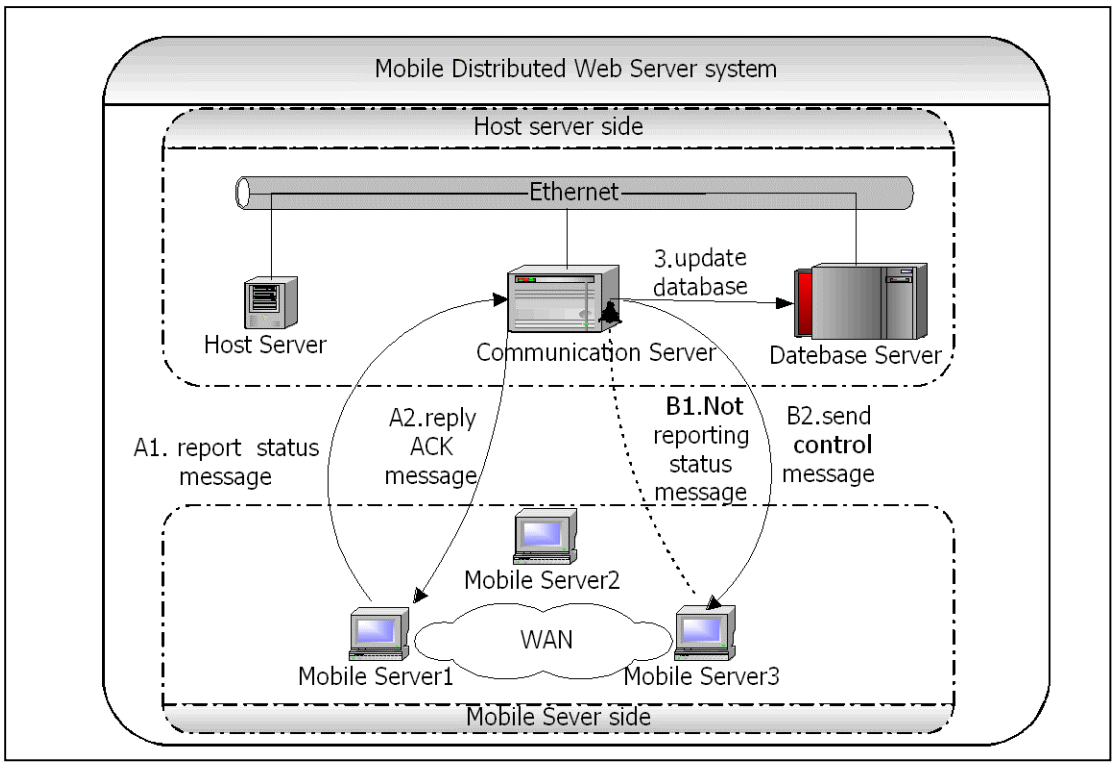

Figure 3. Passive Message-Probing Model

In our system, we use HTTP redirection to redirect client's request to one of mobile servers. In this process, we must make sure that this redirecting process will not redirect client's request to an unavailable server. We provide a solution called Message-Probing Model base on Message-passing model that we described in previous section and replication of web contents. We can distribute multiple copies of one web content to many mobile servers, each mobile server have one copy. In our system, host server has one copy of every web contents. This can ensure that at least one copy of every web contents is available. In other words, all clients' request will be responded since at least host server can response these requests.

On the other hand, System will keep mobile servers' status data in database server. This information will be kept in a database table. In this table, we keep mobile server's information such as mobile server's IP address, hardware information, utilization, and an availability flag. Mobile servers send this information periodically in status message. Communication server will gather the status message. Then communication server will update database base on these status message. We call this method as Passive Message-Probing Model. Figure 3 shows this model. In figure 3, mobile server1 send status message (A1) periodically, if this message received by communication server, it will acknowledge this message (A2). Then communication server will update the information of mobile server1 in database server. If mobile server do not send status message in time for some reasons, communication server will send control message to this mobile server to ask it send status message. If still no response, them communication server will update database

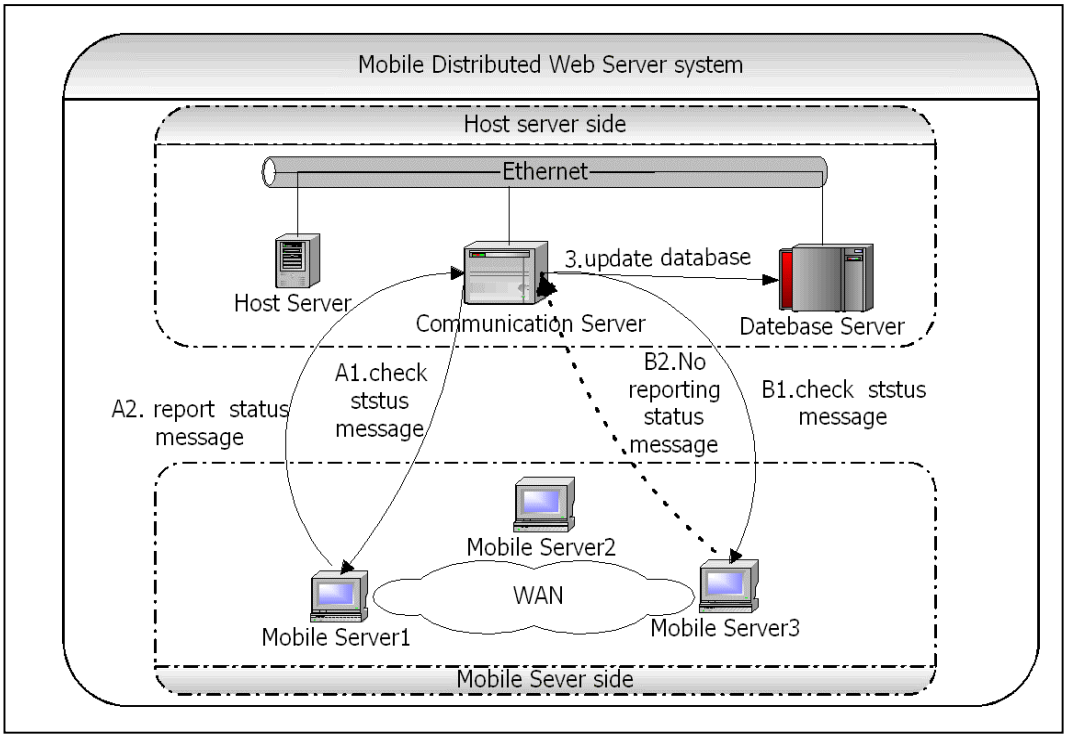

Figure 4. Active Message-Probing Model

server and mark this mobile server as unavailable. Message (B1) and (B2)in figure 3 shows this scenario.

If one mobile server sends its status message in time and its status is available, its status will be set available. If one mobile server sends its status message in time and its status is overloading, then its status will be set to unavailable. If one mobile server cannot send its status message to communication server in time for some reason (i.e. system crash or network failure), communication server will send control message actively to this mobile server. If this mobile server response, then communication server will acknowledge this message to the mobile server. If not, this mobile server will be determined as unavailable. There are two ways that one mobile server become available: one is this mobile server send status message to communication server. The other is communication server detect mobile server is unavailable by send control message periodically. When the service program of one mobile server detects some abnormal condition (for example, system reboot), it can send control message to communication server that indicates service unavailable.

Alternatively, communication server can send control message to some selected mobile servers actively to check their status. We call this method as Active Message-Probing Model. In this model, the communication server randomly selects some mobile server (for example, 5% of all mobile servers) and send control message to ask them reply their status. If they reply their status message, then communication server update information in database server and mark them as available. If not, the communication server will send control message for a few times then wait for status message of the mobile server. If the mobile server replies, then the communication server

will update information in database server and mark them as available. If the communication server does not receive this message, the communication server will update information in database server and mark them as unavailable. Figure 4 shows this scenario. In figure 4, the communication server send message (A1) and (B1) to mobile server1 and mobile server 3. Then mobile server1 replies its status message (A2) , then communication server update information in database server and mark them as available. Mobile server 3 does not send its status message to the communication server. As describing previously, the communication server will try for a few times and wait. If mobile server 3 replies, then communication server update information in database server and mark them as available. If not, then communication server update information in database server and mark them as unavailable

By combine Message-Probing Model and replication, this system can ensure that most requests of clients will be redirect to available server.

4.2. Security issues

In our system, mobile servers are not under control of host server completely. Since these host server side does not own these mobile servers and their location is distributed globally. It is not clever that we control these mobile servers by local personnel for each mobile server. Besides this, security is another issue. For example, if web content on one mobile server is modified unauthorized, it is dangerous for clients that are redirected by host server. To solve this problem, we need an automatic process to reduce manpower. We use a service program on each mobile server. As described previously, this service program on mobile server can monitor the status of mobile servers, including file status. Since web content is consisting of some related files, we can monitor these files to ensure security. For example, we can monitor files attributes, such as last modified time, to make sure that whether this file has be modified unauthorized. Another way is monitoring user's action. By monitor user action, we can detect user's unauthorized action such as deleting web content. We can encrypt web content if we need protect information inside the web content. By combining these methods, this system can provide security for web content on the mobile servers.

5. Implementation

In order to examine our system design, we implement this system. As describing in previous sections, no software or hardware modification needed. We implement this system on Microsoft windows 2000 operating system. We integrate many existing systems to implement our system. These systems include Microsoft Internet Information Server (IIS) and Microsoft SQL Server. Besides of these systems, we integrate several software technologies such as ASP (Active Server Pages), COM (Common Object Model)[15,16], and Windows NT service. In our implementation, host server uses IIS on windows2000 server. Communication server is an NT service, database server use SQL server on windows2000 server. Mobile server uses IIS on windows2000 professional or server. All service programs are implemented with NT service. Our redirecting algorithm is implemented with ASP. All messages are passed via Windows Socket (WINSOCK) .we transfer

updated web content via Trivial File Transfer Protocol (TFTP)[14]. We choose TFTP instead of FTP because we only need to transfer files between two servers. TFTP is a simple protocol that meets our requirement. In database server, we maintain several tables in database: the IP-mapping table, the web content table, the server table, and the user table. IP-mapping table handles with IP to country-code mapping. Web content table deals with web content information. Server table keeps server information and user table keeps user information. We also use COM (ACTIVEX) technology to facilitate our mobile server constructing process. We package all software that mobile server need and deliver them via ACTIVEX. Since ACTIVEX can be embedding in web pages like JAVA, we can deliver our mobile server software via web pages. When user clicks a hyperlink on web pages, then this mobile server package will download to user's computer and install. After installing process, this computer is become a mobile server and can response client's request. We also use log analysis tools to analysis web loading, especially some popular web content. Log analysis tool can help us finding Web Content Requesting Imbalance (WCRI) in a website. An update tool is developed for updating and synchronizing web content in all servers.

6. Conclusion

In this paper, we propose our concept of mobile distributed web server system and design a possible implementation. We convince that this system can solve requesting imbalance problem with low cost and efficiency. This system can be used to redirect high resource-consuming web contents like multimedia files. This system's flexibility is another advantage. However, some experimental results are needed in different redirecting algorithms.

Reference

[1]. Redirection algorithms for load sharing in distributed Web-server systems *Cardellini, V.; Colajanni, M.; Yu, P.S.* Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on , 1999 Page(s): 528 –535

[2]. Dynamic load balancing on Web-server systems *Cardellini, V.; Colajanni, M.; Yu, P.S.* IEEE Internet Computing , Volume: 3 Issue: 3 , May-June 1999 Page(s): 28 -39

[3]. Huan-Chao Keh, Timothy K. Shih, and Jason C. Hung, "Electronic Mobile Notebook -- an Application for Distance Learning on the WWW," in International Journal of Computer Processing of Oriental Languages (IJCPOL), Special Issue on Virtual University, Vol. 13, No. 3, USA, 2000.

[4]. Timothy K. Shih, Jiung-Yao Huang, and Jason C. Hung, "An Efficient Approach to Holding a Virtual Conference," Proceeding of the National Science Council, R.O.C. Part A: Physical Science and Engineering, Vol. 25, No. 4, July, 2001.

[5]. Timothy K. Shih, Jiung-Yao Huang, and Jason C. Hung, "EVCS -- A Complete Electronic Virtual Conference System," Accepted for publication in International Journal of Software Engineering and Knowledge Engineering (IJSEKE), 2001.

[6]. Asia Pacific Network Information Centre(APNIC) web site http://www.apnic.net

[7]. American Registry for Internet Numbers (ARIN) web site http://www.arin.net/

[8]. RIPE (Réseaux IP Européens) web site http://www.ripe.net/

[9]. RFC2068: Hypertext Transfer Protocol -- HTTP/1.1. R. Fielding, J. Gettys,J. Mogul, H. Frystyk, T. Berners-Lee. January 1997.

[10]. RFC 1350: THE Trivial File Transfer Protocol (revision 2) K. Sollins, MIT, July 1992

[11]. MSDN online website: http:// msdn.microsoft.com

[12]. Microsoft website: http://www.microsoft.com