

A REALIZATION OF THE AUTOMATIC ESTABLISHMENT SYSTEM OF SECONDARY SERVER BASED ON THE WEB SERVICE

RyongSu Kim and YongHak Sin

Institute of Information Science, Kim Il Sung University, D.P.R. of Korea

Abstract

In this paper we propose the method for preparing the preliminary server which can perform the role of the primary server when it may be attacked or other abnormal issues happen through the data synchronization system based on the web service. We propose and realize a new method to establish the replication system which can replace the primary server by changing the IP address of the secondary server in case the primary server has stopped working while serving the customers the information preparing the primary and secondary servers. The important point is that the changed data of the primary server should be sent to the secondary server's database and that the added files should also be sent. In order to solve this problem we suggest the immediate synchronization protocol based on the web service and propose the method to safely establishment the secondary server through the data synchronization system of streaming style. This method makes the services of the several servers identical, thus can also be applied to the cloud computing.

Keywords:

Primary Server, Secondary Server, Web Service, Replication, Synchronization

1. INTRODUCTION

Many studies on the data replication and access management in the server management has been performed. Typically the research has been performed to use the RMAN restoration manager in the Oracle server and the Auto Dump functions provided by the MySQL administrator tools [3]. The synchronization system using these functions has the following issues: Firstly, the newly added or updated data cannot be separated and synchronized alone. Secondly, the files cannot be synchronized. Thirdly, there is no system establishment method for the automatic establishment of the secondary server.

1.1 THE REPLICATION

The widespread use of web applications pushes for faster and more reliable web services. Several techniques have been developed to address web-service reliability, and to provide a better Quality of Service (QoS) for Internet users. Replication can be addressed both by a cluster of servers, and by servers geographically distributed in the Internet. Clustering strategies aim at maximizing the web servers throughput, rather than minimizing the document download time, and pay little or no attention to main network issues, e.g., the latency of data transfer through the Internet [1].

Applying replication for Web applications and content distribution has received a lot of attention in the Web and database communities. At the same time, many database and application server vendors are beginning to integrate Web acceleration through data caching in their software. Database replication technologies

and availability of data centers allow database copies to be distributed across the network. The goal of this approach is to offset the high cost of replica synchronization by moving data closer to the users (similar to caching in which data is moved closer to the users reducing network latency). Such data center architectures make it more flexible to deploy a distributed website. However, the system architecture of data center hosted Web applications has many drawbacks that prevent it from being considered for many e-commerce applications, in which content freshness, request response time, application result precision, and transaction throughput need to be maintained at a very high level and sometimes they need to be guaranteed to be within an acceptable threshold specified as a part of QoS agreement [2] [5] [8].

The general problem refers to how to decide on the number and placement of replicas across multiple servers, with the aim to improve the overall system performance and minimize the costs involved. Previous work on data replication has focused on networked systems, distributed database systems, and most recently, content delivery networks (CDNs) [3].

In recent years, web users (clients) have increased explosively with the rapid expansion of the Internet. Consequently, the accesses from clients have come to be concentrated on popular servers, causing a problem of increased loads of these servers and a significant degradation in the content obtaining time. As measures against this problem, mirroring or caching, CgR (Caching goes Replication), and others, and techniques for distributing the load to multiple servers have attracted attention and have been studied. The authors have previously proposed an algorithm for replicate-distributing content to dynamically determine and allocate the replicated numbers by content in accordance with the popularity.

However, even if such replication distribution algorithms are applied, the number of replicated servers used to accommodate the replicated content is limited. When the accesses from clients are excessive, each server goes into an overload state even with the distribution of content, possibly causing a significant degradation in the content obtaining time. In order to maintain the QoS enjoyed by the users in the replicated servers in such a state, accesses to the replicated servers must be limited. This problem is not considered in the various references mentioned earlier. In this paper, the server which is the information originating source, having the original content, is called the primary server (PS), and the replicated server accommodating the replicated content is called the replicated server (RS) [4].

Data replication maintains multiple copies of data, called replicas, on separate servers. So the requests to these servers are answered locally. Replication improves performance by the following:

- Reducing latency, since users can access replicated data, so it avoids remote network access and

- Increasing throughput, since multiple computers can serve data simultaneously.

When data is replicated in more than one computer, it is necessary to keep it consistent. There are some protocols such as single lock, distributed lock, primary copy, majority protocol, biased protocol, and quorum consensus protocol which are responsible to keep data consistency.

The remote server must be kept synchronized with the primary server, as updates are performed at the primary. This synchronization is achieved by sending all log records from primary server to the remote backup server. The remote backup server must be physically separated from the primary, so a disaster at the primary does not damage the remote backup server. When a primary server fails, its remote backup server is responsible to answer the requests until the primary server come back to stable state [5].

1.2 THE WEB SERVICE

Recently, many researches on web services have been performed [9] [10]. Web services are autonomous software systems identified by URIs which can be advertised, located, and accessed through messages encoded according to XML based standards (e.g., SOAP, WSDL, and UDDI) and transmitted using Internet protocols. Web services encapsulate application functionality and information resources and make them available through programmatic interfaces, as opposed to the interfaces provided by traditional Web applications which are intended for manual interactions. In addition, since they are intended to be discovered and used by other applications across the Web, Web services need to be described and understood both in terms of functional capabilities and Quality of Service (QoS) properties. The emergence of Web services (e.g., for order procurement, finance, accounting, human resources, supply chain, and manufacturing) has created unprecedented opportunities for organizations to establish more agile and versatile collaborations with other organizations. Widely available and standardized Web services make it possible to realize Business-to-Business Interoperability (B2Bi) by inter-connecting web services provided by multiple business partners according to some business process: a practice known as Web Services Composition [6].

The technology of Web service is becoming mature and increasingly applying in the whole network. But it is vulnerable to vicious attacks too. Since it has been far from enough to solve the problems by technology only, security risk assessment, therefore, is demanded to find out the risks confronting web, consequently, for us to take effective measures to reduce and avoid them. Risk analysis is considered as an important process to identify the known and potential vulnerabilities and threats in the web service security.

Web service, based on software technologies such as WSDL (web security description language), SOAP (simple object access protocol), XML (extensible markup language), UDDI (universal description discovery and integration), provide an environment for dynamic discovery and integration of new and pre-existing software components which are distributed across the Web. Web service made the mutual communication of different platforms in various environments not be a problem, and supplied the software application system integration and the network calculation technology with an engine [7].

WSDL is an XML format used to describe network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. WSDL is often used in combination with SOAP and an XML Schema to provide web services over the Internet. A client program connecting to a web service can read the WSDL to determine what operations are available on the server. Any special data types used are embedded in the WSDL file in the form of XML Schema. The client can then use SOAP to actually call one of the operations listed in the WSDL.

Web Services (WSs) technology is emerging as a powerful vehicle for organizations that need to integrate their applications within and across organizational boundaries. It has already been established by the World Wide Web consortium and solves the problems of usability and functionality of the services. Today, all the businesses that provide services over the Internet are based on this technology of design, publication and exploitation of their as it simplifies the creation and provision of services over the Internet by defining a large number of specifications and rules that one can easily follow. This development has given services a new dimension and henceforth each business can relatively easily create and provide services over the Internet but it can also use existing services in a simpler way. In particular, the process-based composition of WSs is gaining a considerable momentum as an approach for the effective integration of distributed, heterogeneous, and autonomous applications. A WS is a self-described application that uses standard Internet technologies to interact with other WSs

In a web environment, multiple WSs may provide similar functionalities with different non-functional property values (e.g., different prices). Such WSs will typically be grouped together in a single community. To differentiate the members of a community during service selection, their non-functional properties need to be considered. These properties are characterised as quality of service (QoS) and it is highly recommended to be taken into account during the WS selection. As mentioned beforehand, WS technology is a promising solution for addressing platform interoperability and compatibility problems faced by system integrators [8], [9].

2. THE FORMATION OF DATA SYNCHRONIZATION SYSTEM AND ALGORITHM

This paper proposes and settles the data synchronization system based on the web service for the successful solution of the data replication issues.

2.1 THE MODE CHOICE OF THE DATA SYNCHRONIZATION SYSTEM

The synchronization system is the one which enables the change of one side database to be made simultaneously at the other side database. The synchronization system works at the main two directions. One is the method which the updated data of the primary server is changed in the secondary server at the same time identically and the other one is that the secondary server sends signals to identify the updated data at specific time intervals. The first one has the weakness of the large load on the

primary server and the second one is that the adaption of the updated data to the secondary server takes some time. This paper selects the second one thus giving little load to the primary server and expanding the servers as well as the secondary server.

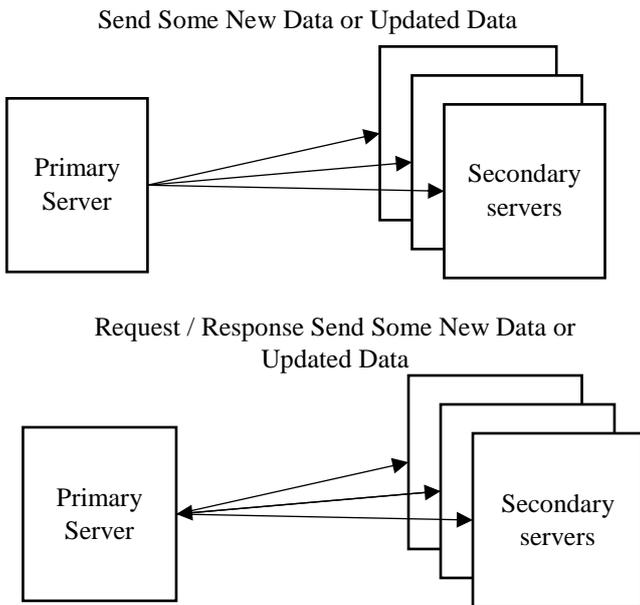


Fig.1. Mode of data synchronization

2.2 MODE SELECTION OF THE DATA SYNCHRONIZATION

Here it arises that what is the object of the data synchronization. Every server has two kinds of data. They are the business data stored in the database and the files saved in the hard disk. All applications of the user is either stored in the database as the data record through the business process of the server or saved in the hard disk in the form of file. Therefore, in the data synchronization only the database and the files are synchronized.

2.3 THE DATABASE ESTABLISHMENT FOR THE DATA SYNCHRONIZATION

The main issue in the data synchronization is to identify only the newly added or updated data to reduce the communication load. To this end this paper has designed a new database to identify the new or updated data proposed the identification algorithm. Above all let's have a look at the data synchronization system realization based on the database design and the web service. Every database table has the information of data creation and update.

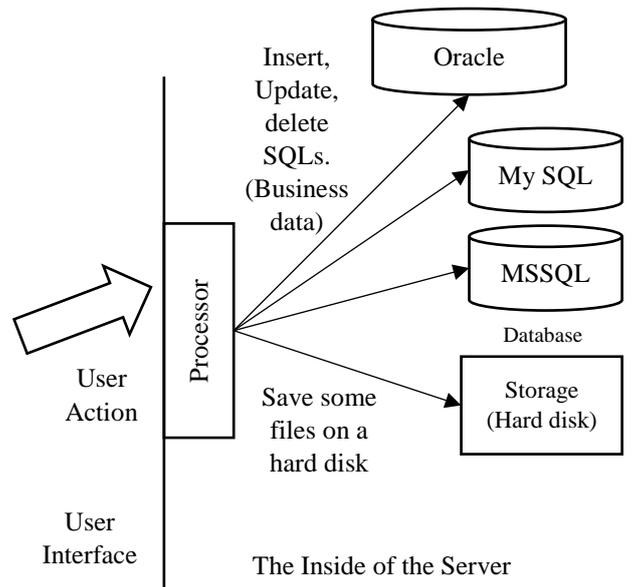


Fig.2. Kinds of data process (SQLs or Files)

Table.1. Design of the database

Field name	data type	Description
Ndx	bigint	Unique identifier (main key, automatic increase field)
...		fields...
Ctime	datetime	Data added time
Mtime	datetime	Data updated time
Rtime	datetime	Data received time (field only in the secondary server)
Iscopied	bool	The field which shows that the data is synchronized (used only in the secondary server)

Here "rtime" enables the secondary server list the time of receiving the record. The "iscopied" field is the one which is used in the secondary server in case of the record representing the added files. This is reduced to 1 when the added file for the record is synchronized. The deleted data is registered in table actions table of Table.2. The database is designed representing the field of Table.1 in the sub tables. However, when the program of the server is already made and the program cannot be updated because of the change of the database field like the Table.1 one table is made like Table.2 in the database and trigger is defined in every table thus presenting the time of creation, update and deletion.

Table.2. Table actions structure

Field name	Data type	Description
Ndx	bigint	Unique identifier (main key, automatic increase field)
Tablename	varchar	Table name
Tablndx	bigint	Main key of table

sqltype	varchar	Kinds of process (insert/update/delete)
ctime	datetime	Data added time
mtime	datetime	Data updated time
rtime	datetime	Data received time (only used in the secondary server)
iscopied	bool	The field which shows that the data is synchronized (used only in the secondary server)

2.4 THE FORMATION OF THE DATA SYNCHRONIZATION SYSTEM

The secondary server requests the primary server to send the updated data at specific time intervals. At that time the secondary server sends the request including the information which shows how much data is received. Then the primary server identifies the newly updated data and sends the response. In the data synchronization system [the Export and Import are installed in the front of the database server, the source and target. It can avoid the direct access to the database server and downloads and saves the necessary data of the Import from the data provision server in the target database server. The Import sends the request S_t the time of t to the data server and the data server sends the response $R(S_t)$. The request data S_t is defined in Eq.(1).

$$S_t = \{TN, N, T, TC\} \tag{1}$$

where, TN is a table name, N is the maxim of the data identifier, T is the maxim of the updated time in the received data and TC is the number of the updated time T .

The response data $R(S_t)$ is defined in Eq.(2).

$$R(S_t) = H(S_t) + TF + M(S_t) + C(S_t) \tag{2}$$

where, $H(S_t)$ is the head of the response data and is defined in Eq.(3). TF is the number of the data order. In other words, they are the field names of database which is ordered. $M(S_t)$ is the updated data and $C(S_t)$ is the newly added data.

$$H(S_t) = RN + LN \tag{3}$$

where, RN is the number of the response data and LN is the number of the left data. $M(S_t)$ and $C(S_t)$ are plain data output in the field order defined in the TF .

$$M(S_t), C(S_t) = \{D_1, D_2, \dots, D_n\} \tag{4}$$

The Fig.3 is the structure of the data synchronization system, where

- “time counter”: module which operates the “reception” module at specific time intervals
- “reception”: module which receives the data from the “output” service and saves in the database.
- “output”: module which responds to the request of the “reception” module

The “output” module operates on the web service. The program of the data synchronization copies the data of the “source database” to the “target database identically.

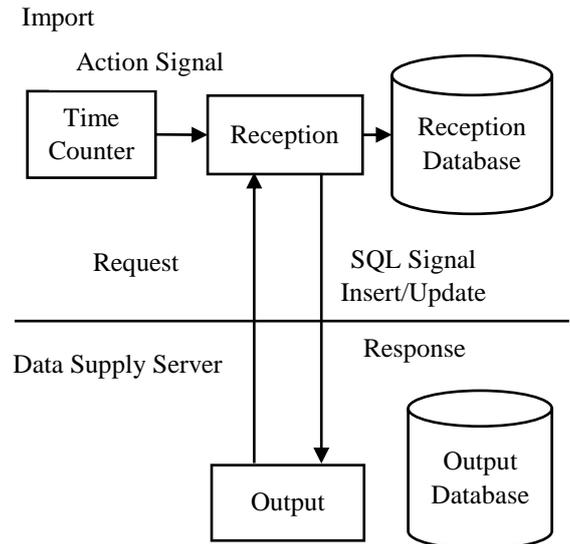


Fig.3. Structure of the data synchronization system

2.5 DATA REQUEST AND RESPONSE ALGORITHM

The Fig.4 is the action process of the “reception module”. The “reception” module forms the S_t , synchronization parameter to call the output module of the data supply server (primary server). The mode of calling is to use the WSDL call of SOAP object or RPC mode. Next the “output module” action process of the data supply server is defined.

The action of the output module

- Receive the parameter $S_t = \{TN, N, T, TC\}$ from the reception module.
- Get TF , the field list which will be output from the database.
- Compare the number of the data which the $mtime$ field value of TN table is T and the TC to see whether they are same or not.
- Yes: Combine the data which the $mtime$ field value is T into $M(S_t)$
- No: continue the action. (this means the data of the T vision is all synchronized)
- Identify whether there is data which $mtime$ field value of TN table is larger T .
- Yes: combine the data which the $mtime$ field value is larger than T into $M(S_t)$
- No : continue the action. (this means there is no newly updated).
- Identify whether ndx field value of TN table is larger than N .
- Yes: assembly the data record which is larger than N into $C(S_t)$
- No continue the action (this means that there is no newly added data).
- Make the $H(S_t)$ by getting of $M(S_t), C(S_t)$
- Example: “ H : 200:100”
- Description: H : head abbreviation, 200; updated records, 100: the number of added records

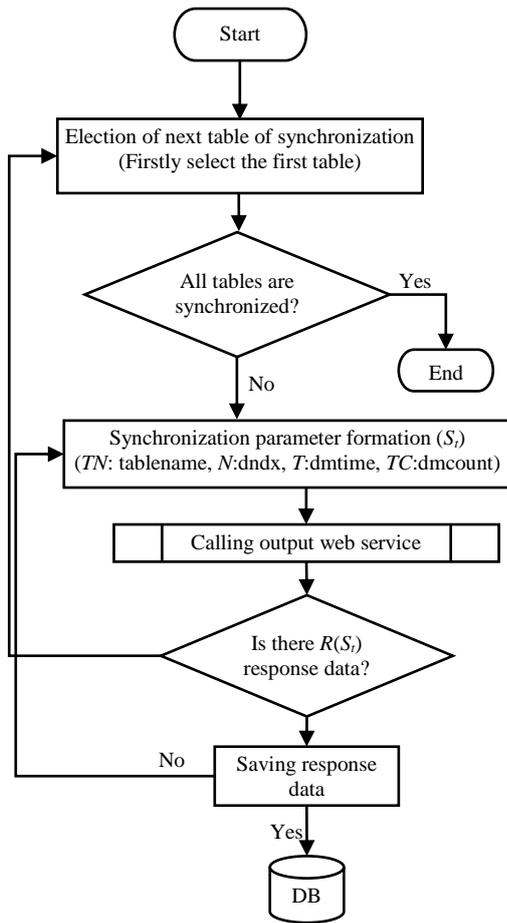


Fig.4. Action process of the reception module

- Output $R(S_i)$, response data (see Eq.(2))
- Output in the order of $H(S_i)$, TF , $M(S_i)$, $C(S_i)$
- End the process

3. THE REALIZATION OF THE DATA SYNCHRONIZATION BASED ON THE WEB SERVICE

The previous algorithm can be realized based on the web service. Web service is the most appropriate service system which supports the combination between the systems. Sometimes it is more appropriate to communicate with plain method than the one by the web service. In case the communication speed between the computer systems because of the cloud computing system realization or the other regional characteristics this paper realizes the synchronization of RPC method not using WSDL. In a word, the algorithm is mapped by the two methods: the WSDL function calling using the SOA-oriented and SOAP object, and RPC mode. Here in the RPC method it is streamed and the immediate synchronization protocol is proposed and realized. In particular, in the file synchronization the streaming data process is essential.

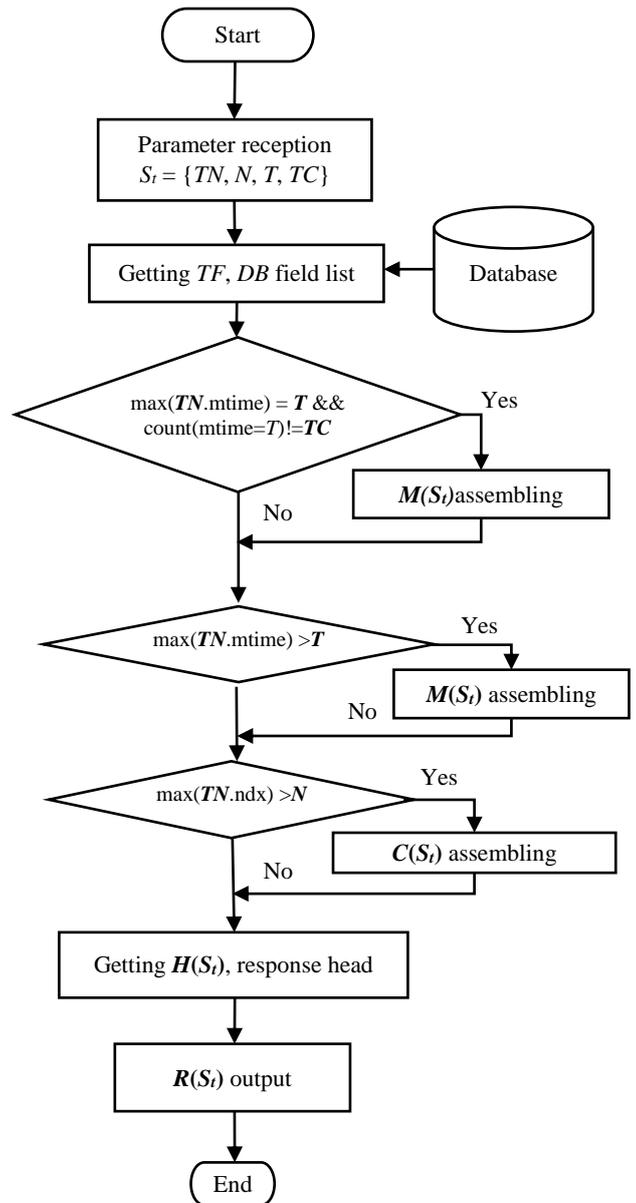


Fig.5. Action Process of the Output Module

3.1 THE REALIZATION BASED ON WSDL AND THE RPC COMMUNICATION METHOD OF AND STREAMING MODE

There is a lot of study on the WSDL, so only the WSDL area definition of the output model is mentioned. The WSDL can be defined by considering the Eq.(1)-Eq.(4). In order to process the response with the XML or JSON mode the whole data from the beginning to the end. That's how the parsing is available. At that time the large amount of the response data makes the process in the slow net impossible. Therefore in this paper it is settled by the streaming communication apart from the immediate synchronization protocol in consideration of the net speed. The Fig.6 is the output data example considering the Eq.(1)-Eq.(4). Here all data is plain text.

<p>Format:</p> <p>H num of response data num of left data </p> <p>T ndx field 1 field 2 ... field n ctime mtime </p> <p>M value 11 value 12 value 13 ... created time modified time </p> <p>M value 21 value 22 value 23 ... created time modified time </p> <p>...</p> <p>C value n1 value n1 value n3 ... created time modified time </p> <p>...</p>
<p>Example:</p> <p>H 10000 230000 </p> <p>T ndx firstname lastname gender birthday address ctime mtime </p> <p>M 1026 Benjamin Walter male 2/22/1987 Ros... 2017-08-2311:22:33 2017-09-2308:22:33 </p> <p>M 1231 Wiliam Tell male 6/3/1953 Ros... 2017-08-2311:22:33 2017-09-2308:22:33 </p> <p>C 2788 Samuel Johnson female 6/17/1955 Ros... 2017-08-2311:22:33 2017-09-2308:22:33 </p> <p>C 2789 Paul Robeson 2/5/1979 Ros... 2017-08-2311:22:33 2017-09-2308:22:33 </p> <p>C 2790 John Hopkins female 10/3/1981 Ros... 2017-08-2311:22:33 2017-09-2308:22:33 </p> <p>...</p>

Fig.6. Example of using the immediate data communication protocol

In the Fig.4, the answer to the question of “Is there $R(S_i)$, response data?” in the action algorithm of the reception module is that when there is response data to the question the process of saving response data is conducted. However, the large amount of response data takes fairly long time. To this end, this paper proposes the streaming communication between the reception module and the output module. The process of the streaming communication module between the two modules can be defined as following,

$$R(S_i) = \{R_1, R_2, \dots, R_n\} \tag{5}$$

The output module divides the $R(S_i)$, the response data of Eq.(4) into n . (the length of each part can be different according to the net communication capacity).

$$R_1 = P\{H, T, L^*, X?\} \tag{6}$$

where, H is the response data head to Eq.(3) and is presented with the letter “ H ” in front of $H(S_i)$. P is the function to convert the objects into plain text. This function inserts field division symbol between the fields and the data division symbol between the data. In Fig.7, the field division symbol is “|” and the data division symbol “\n”.

$$H = \{“H”, H(S_i)\} \tag{7}$$

where, T defines the field name in the response data, and it is presented with the letter T in front of the TF .

$$T = \{“T”, TF\} \tag{8}$$

$$L = \{M(S_i)^*, C(S_i)^*\} \tag{9}$$

$$R_n = \{Y?, L^*, X?\} \tag{10}$$

where, L is formula presenting one data. There may be several $M(S_i)$ s of the updated data and $C(S_i)$ s of the added data. X and Y are not the whole one, but divided data.

In the data as the above equations, the reception modules define the processing data after receiving the R_n segment as following:

$$D_n = X_n - 1 + R_n \tag{11}$$

In other words, the new data is added to the left data which is previously processed. In this way it processes the response data by segmentation thus enabling the full and reasonable data synchronization even in the slow network.

3.2 THE REALIZATION OF THE FILE SYNCHRONIZATION AND DELETION

The file synchronization usually processes the large capacity, so data sending through the streaming mode is chosen. While synchronizing the table of the added file, the request for a file is sent to the output module in case of the iscopied field being 0. And receive the file by the streaming mode and make it 1. The processing of the deleted records can be done in the same way like the Table.2. When synchronizing the “table-actions” of table, the “tablendx” data of table “tablename” can be deleted if the Field “sqltype” is shown as “delete”. At that time if the to-be-deleted record indicates the file, it can be deleted simultaneously.

4. CONCLUSION

In this paper we studied the data synchronization algorithm for the automatic establishment of the secondary server and realized it by web service. In addition, we proposed the data synchronization of streaming mode considering the network communication load between the primary server and the secondary server. This research can be applied to load distribution system and cloud computing system as well as the establishment of the secondary server. We plan to introduce this research into the load distribution system, and establishing the secondary server we can bring only the list, not the whole data of the primary server to serve the clients when they are requested, the proxy mode and the data synchronization security.

REFERENCES

- [1] Marco Conti, Enrico Gregori and Willy Lapenna, “Content Delivery Policies in Replicated Web Services: Client Side Vs Server Side”, *Cluster Computing*, Vol. 8, No. 1, pp. 47-60, 2005.
- [2] Wen-Syan Li, Kemal Altintas and Murat Kantarcioglu, “On Demand Synchronization and Load Distribution for Database Grid-based Web Applications”, *Data and Knowledge Engineering*, Vol. 51, pp. 295-323, 2004.
- [3] Ling Zhuo, Cho-Li Wang and Francis C.M. Lau, “Document Replication and Distribution in Extensible Geographically Distributed Web Servers”, *Journal of Parallel and Distributed Computing*, Vol. 63, pp. 927-944, 2003.

- [4] P. Alireza, K. Mohammadreza and H. Jafar, "Using Replicated Data to Reduce Backup Cost in Distributed Databases", *Journal of Theoretical and Applied Information Technology*, Vol. 11, No. 2, pp. 195-203, 2010.
- [5] Marco Conti and Enrico Gregori, "QoS-based Architectures for Geographically Replicated Web Servers", *Cluster Computing*, Vol. 4, No. 2, pp. 109-120, 2001.
- [6] Liangzhao Zeng, Boualem Benatallah, Anne H.H. Ngu, Marlon Dumas, Jayant Kalagnanam and Henry Chang, "QoS-Aware Middleware for Web Services Composition", *IEEE Transactions on Software Engineering*, Vol. 30, No. 5, pp. 311-327, 2004.
- [7] Pang Jianjing and Peng Xinguang, "Trustworthy Web Service Security Risk Assessment Research", *Proceedings of International Forum on Information Technology and Applications*, pp. 1-4, 2009.
- [8] Vassiliki Diamadopoulou, Christos Makris, Yannis Panagis and Evangelos Sakkopoulos, "Techniques to Support Web Service Selection and Consumption with QoS Characteristics", *Journal of Network and Computer Applications*, Vol. 31, pp. 108-130, 2008.
- [9] Chandrashekar Jatoth, G.R. Gangadharan and Rajkumar Buyya, "Computational Intelligence based QoS-Aware Web Service Composition: A Systematic Literature Review", *IEEE Transactions on Services Computing*, Vol. 10, No. 3, pp. 475-492, 2017.
- [10] Peishi Jiang et.al., "A Service-Oriented Architecture for Coupling Web Service Models using the Basic Model Interface (BMI)", *Environmental Modelling and Software*, Vol. 92, pp. 107-118, 2017.