BISYNC-BINARY XML BASED DATA SYNCHRONIZATION FOR MOBILE DEVICES

1KOTTILINGAM K, 2GUNASEKARAN R, 3PRAVEEN G

1,2Department of Computer Technology, MIT Campus, Anna University
E-mail: k.kottilingam@gmail.com, gunasekaran.mit@gmail.com, praviinysr@gmail.com

Abstract- The technology rapidly expanding for mobile devices will give a communication capability for mobile users. Every device has its own database so the users can synchronize the data from anywhere and anytime. Recent improvements in mobile device for data synchronization is based on the relational databases and also used an algorithm SAMD (Synchronization Algorithm based on Message Digest) to improve the synchronization. In current practice, mobile application can able to send several requests alternatively to synchronize itself. But relational database involves much more time consuming for data synchronization whenever an application send multiple requests alternatively. Similarly message digest algorithm consumes more time and it involves more calculations. To achieve efficient synchronization we propose xml based data synchronization for mobile devices in heterogeneous environment. Xml is adopted to describe databases. We can achieve this by replacing the existing relational data to xml data in a real time manner. This paper includes BiSync mechanism, a binary encoding method which is based on property of xml data to reduce the time consumption for processing several alternative requests from mobile applications.

Keywords- Data synchronization, Binary XML, Mobile devices, Xml based synchronization, Data integrator.

I. INTRODUCTION

Transmission technologies have steadily improved the user connectivity and available bandwidths. Many of the industry operators are looking for new ways to maximize their investment in communication networks while ensuring reliable, secure data transmission. There is a variety of communication solutions, the two most common is wired and wireless technology. Wired LANs use Ethernet cables and network adapters. Although two computers can be directly wired to each other using an Ethernet cable. The cable configuration for a wired LAN varies depending on the mix of devices, the type of connection and whether internal or external modems are used. Traditionally, an Ethernet connection offers only 10Mbps bandwidth and also 100 Mbps fast technology but it costs little more. Fast Ethernet should be sufficient for home file sharing, gaming and high speed internet access.

Wireless network uses high frequency radio waves rather than wires to communicate between nodes. Individuals and organizations can use this option to expand their existing wired network. Each device can communicate directly with all of the other wireless enabled devices. Mobile computers do not need to be tied to an Ethernet cable and can roam freely within the wireless network range. WiMAX is also a technology based on the IEEE 802.16 specifications to enable the delivery of wireless broadband access. It can be able to deploy in both licensed and unlicensed spectrum and also to extend the fixed access to mobility and broadband multimedia services delivery.

It can be used in both point to point and the typical WAN type configurations that are also used by 2G and 3G mobile network carriers. WiMAX signals can function over a distance of several miles with data rates reaching up to 75Mbps.

An increasing number of new autonomous, portable devices such as smartphones, Tablet PC, PDAs (Personal Digital Assistants), laptops, e-reader and netbooks has become a significant part of everyday life and work, leading to a decentralized, location independent, wireless computing environment.

Unlike traditional wired communications, the deployment of these new services faces several important restrictions due to the wireless mobile environment. The amount of expendable energy, both in total and per unit of time on the battery powered smartphones is orders of magnitude smaller than on conventional computers connected to electrical grid. Also by optimizing the smartphone CPUs for processing cycles per energy unit, the amount of processing and other energy consuming activities is still limited compared to conventional computers. As battery technology is improving slowly, highly energetic processing on a small device would introduce considerable heat dissipation issues.

The prevalent technology for persistent storage is flash memory. As with processing, the amount of persistent storage on smartphones has been growing rapidly. In contrast to processing and networking, flash memory requires energy only when in active use, so energy is less of a limiting factor in this case. On a
smartphone communications is typically quite expensive in terms of energy. Furthermore, the smartphone may experience periods of disconnection due to radio interference, hand off between cellular base stations, lack of coverage etc. Therefore, it is not easy to process a large size of stored data and maintain a continuous connection with the server side database. For these reasons, mobile devices have mobile databases in order to achieve stable data processing.

In order to avoid later communication overhead or unavailability, replication helps to creating local or nearly copies of useful data items in case of on demand retrieval from their original location. Disconnected or poorly disconnected users rely primarily on local copies of the required data in order to achieve availability and reliability. Full replication results in additional energy consumption and heavy traffic generation. Partial replication can be regarded as an alternative solution but it also hides some important issues that need deep consideration. It may be achieved by partitioning the data. In mobile context and especially in peer to peer systems this task could prove to be very difficult as the data placement problem in such systems is NP complete. Generally replication in a highly dynamic mobile environment is a difficult task.

Caching is a widely mechanism for improving data access performance and availability. The main difference between caching and replication is that the former occurs after the retrieval and use of the data while the latter in an a priori way. Especially in a wireless mobile environment, caching of frequently accessed data in a mobile nodes local storage can reduce energy and bandwidth consumption as well as query delays, while at the same time increasing the systems flexibility in cases of disconnection. A client must always ensure that data in its cache is up to date in order to be able to provide valid responses in submitted queries. The very nature of a mobile environment indicates the distribution and heterogeneity of data. Either symmetric peer to peer architecture or a client server one, mobile nodes must be able to reach the desired data in a cost efficient way. Data transfers must be minimized so that wireless environment’s limitations do not result in a degraded service and mechanisms must be deployed in order to confront the frequent disconnections and high data access performance, data availability and consistency.

II. ANALYSIS OF EXISTING SYSTEM

A. Message Digest Algorithm

Message digest algorithm is based on a server for computation and synchronization of data that works against a database server and is connected to a wireless network to which the clients also connect. As the mobile devices has limited battery, less computing power and less memory capacity, it is difficult to process a large set of data and without a stable network connection. It is preferable to do as much computations as possible on the server. Initially the mobile device downloads required amount of replicated data from the server to perform the work in an offline state. The advantage of this algorithm is purely based on relational database and it uses standard queries in its communication. This algorithm works under heterogeneous environment that it does not depend upon different databases or different platforms on the server and clients.

Message digest algorithm used a single hash function (H) applied on the message which gives a fixed length hash value h.

\[ h = H(M) \]

\( M \) is the message present in the database table with random length that is transformed into a message digest. If there is any modification in the message, then it will generate a different hash value. If there is no modification, the message digest will remain same. In the case either the client or the server has inserted a new row, that row is copied to the other side message digest table and the data is sent. If instead a row is deleted the message digest value is set to null and the other side will delete its row. Hence, the synchronizing policy is based on message digest value. The limitations of using this algorithm are the client always sends its entire data to the server in synchronization which can be troublesome with a bad network connection. There will also be a lot of extra tables on the server side. It takes more time to compare message digest value whenever a insertion of new data takes place.

B. Data Management and Synchronization

Data synchronization is not required for one-tier architecture, as one-tier architecture is not shared among multiple devices and the mobile device API (Application Programming Interface) directly receives the data from the database. In multitier architecture, the database is stored at a remote server and the copies of the database are cached at the client device. A mobile device can select data if data is broadcast, pushed by the server or it can ask the server for specific data and store that data in a local database. In such cases, data synchronization is required. The server provides and updates all local copies of the database stored at each mobile device. Cached copy of data is stored at tier 2 and is retrieved using some connectivity method and then transmitted to the requested device. The advantage of caching is that it takes less time to access a requested data from the server as each device has quick access to the cached data. The limitations is that the data stored at the...
server should be consistent with the data stored at other nodes and it requires some synchronization method to be applied. Before applying synchronization method, one should have knowledge about the structure of database. Data can be stored in a logical manner such as lookup tables where data is stored in tabular format.

III. PROPOSED WORK

This paper proposed an XML based data synchronization for mobile devices in heterogeneous environment. A major drawback in the use of relational database is that machine performance. If the number of tables between which relationships to be established are large and the tables themselves effect the performance in responding to the SQL queries. In case of manipulation, the server would take replica of data consumes more storage size. Complex queries require sophisticated processing power. Although most of mobile devices can manage the databases of the size and complexity often encountered in a small business setting, a database with external data sources or very complex data structures may require more powerful services to return results within an acceptable response time. Suppose a mobile client sends multiple alternative queries like insert update and delete to the server, it consumes more time to process in the relational database. For example, if the server database has 10,000 records with a part of attributes and the mobile client requests up to 300 queries alternatively like insert, update and delete, then server consumes more time to response for the mobile client. Insertion simply inserts the data to the database which has zero comparisons. But in the case of update and delete operation, it takes at least one or more comparisons in the database before manipulating the data. So for 200 queries of update and delete operation, the relational database consumes more time to process. This problem can be overcome by synchronizing XML data. The figure shows the processing of XML database inside the mobile clients through the network.

Figure 1: Processing of XML data

Comparatively large amount of structure exposed when using XML as well as its widespread deployment motivates to consider the problem of synchronizing data encoded as XML. Since the XML is completely text based and hierarchical structure, one can find the problem on a more general level is that of synchronizing tree structures. We propose XML based synchronization which includes two methods by checking two constraints. First is that request received from the mobile client is relational request and second is that whether the request received is XML based or not. If the request is based on the relational data, then the server executes Relational to XML mapping by using data integrator to achieve XML based synchronization.

Data integrator performs three steps to convert from relational model to XML data in a real time manner. Initially the relational data is allowed to knowledge modules. Secondly the server set the schema definition for XML data. Finally the mapping stage converts the relational to XML data based on the defined schema. If the request is based on the XML data, then the server creates a binary XML for the faster access. Because the text in the XML structure is highly redundant and bulky, it makes slow to transfer over the internet. So the server processes the BiSync mechanism (binary encoding method) for the XML representation. It reduces the verbose of XML documents, thereby also reducing the parsing cost. It stores the properties of actual XML file. Once the server saved the binary XML for mobile client, there is no need for checking or synchronizing the XML data directly to the mobile client. If the mobile client sends multiple alternative queries, the server refers the property of requested queries in the binary XML and also it saves the responses in the binary XML for frequent access. The BiSync mechanism can also directly represent raw binary data without any indirect textual encoding methods and a backward compatibility mechanism is provided to enable the translation raw binary blocks into an equivalent textual representation when necessary. The XQuery to access the entire information of XML database is shown below. ExtractValue extracts the data from the specified Xpath in the XML database. ‘/’ or ‘//’ represents the Xpath and ‘@’ represents the primary key of the XML data.

```
SELECT extractValue(OBJECT_VALUE, '/EMP/@EMPLOYEE_ID'),
extractValue(value(em), '/EMP/FIRST_NAME'),
extractValue(value(em), '/EMP/LAST_NAME'),
extractValue(value(em), '/EMP/EMAIL'),
extractValue(value(em), '/EMP/SALARY'),
extractValue(value(em), '/EMP/PHONE_NUMBER'),
extractValue(value(em), '/EMP/JOB_ID'),
extractValue(value(em), '/EMP/Salary'),
```

BiSync-Binary Xml based Data Synchronization for Mobile devices

106
BiSync - Binary XML based Data Synchronization for Mobile devices

I. INTRODUCTION

BiSync is a Binary XML based data synchronization for mobile devices. It allows for efficient and effective data exchange between mobile applications and databases.

II. TECHNICAL SPECIFICATIONS

BiSync utilizes Binary XML format (BXML) for efficient data transmission. It supports various mobile devices and platforms, ensuring compatibility across different systems.

III. IMPLEMENTATION DETAILS

The implementation of BiSync involves several components, including a client application, a server, and a synchronization protocol. The client application communicates with the server using BXML messages for data synchronization.

IV. DATA INTEGRATOR

Data Integrator delivers unique next generation Extract Load and Transform technology that improves the performance costs and also reduces the data integration costs even across heterogeneous environments. At runtime, the agent coordinates the execution of data integrator sessions. It retrieves the code stored in the repository, connects to various source and target devices and coordinates the overall data integration process. With the Extract-Load Transform architecture, the agent performs any transformation. It simply retrieves code from the repository and then requests database.

Table 1. Knowledge Modules

<table>
<thead>
<tr>
<th>Type</th>
<th>Knowledge Module</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load</td>
<td>LKM SQL to SQL</td>
<td>LKM &lt;source technology&gt; to &lt;target technology&gt; [loading method]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Loading process drops the temporary loading table and then creates the loading table in the staging area.</td>
</tr>
<tr>
<td>Integrate</td>
<td>IKM SQL Control Append</td>
<td>IKM [&lt;staging technology&gt;] &lt;target technology&gt; [&lt;integration mode&gt; [&lt;integration method&gt;]]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>IKM executes a single set oriented SQL based programming to perform staging area and target area declarative rules on all “CS” tables and source tables to generate result set. IKM then writes the result set directly into target table using defined integration mode.</td>
</tr>
</tbody>
</table>

V. PERFORMANCE ANALYSIS

Graphs depicting the experimental results are in Figure 4 and 5. In Figure 4, we show the response time for both homogeneous and heterogeneous solutions.
As for homogeneous environment, the databases are same in both of the server and client. But for heterogeneous environment, the databases should be different. Hence the synchronization time is less for homogeneous when compared to heterogeneous environment. For 300 queries of execution, the Oracle-Oracle communication consumes 22 seconds to complete the 300th query execution. But Oracle-MYSQL consumes 29 seconds to complete the 300th query execution.

![Figure 5(a), Time analysis for manipulating alternative requests using relational database.](image)

![Figure 5(b), Time analysis between RDBMS and XML database](image)

In Figure 5, we show the time analysis between the relational database and XML databases. In case of Oracle- MYSQL communication, synchronization time is large when compared to the homogeneous environment. Even in the best case, relational database takes 22 seconds to complete the execution of 300th query. But in the case of XML database, experimental result shows it takes less time consuming when compared to relational database. Binary encoding method completely reduces the time for data synchronization.

**CONCLUSION**

We have evaluated the impact of relational to XML data transformation and binary encoding method for XML data. Our results have shown that with the binary encoding method, the synchronization yields performance improvements in XML database. There is no need of care about homogeneous or heterogeneous environment. XML is completely a text based and it is also a platform independent. XML usually runs in main memory, hence we can achieve the synchronization at a faster rate. Its consumes less memory for data synchronization. Therefore even in the case of disconnection, we can ensure the data availability and consistency. Binary XML also saves the frequently accessed data so it consumes less time to synchronize the data.

**REFERENCES**


BiSync-Binary XML based Data Synchronization for Mobile devices