Research and Implementation of Data Synchronization for
Wireless Data Acquisition System

Tao Huang

School of Computer Science

Zhaoqing University

Zhaoqing, Guangdong Province, China

htmouse@qq.com

Abstract—**A wireless data acquisition system will continue to carry out data collection, which will face the problem of data redundancy. Therefore, how to manage the database is particularly important. This paper mainly studies and designs the database synchronization module of the wireless data acquisition system. Following the consideration of the requirements of the system and database synchronization, this paper makes researches of MSF (Microsoft Sync Framework) and designs a database synchronization module on the basis of it, which achieves distributed database synchronization module and peer-to-peer database synchronization.**

Keywords—*Wireless Data Acquisition System; Database Synchronization; Microsoft Sync Framework*

I. INTRODUCTION

With the development of society and economy, people's living habits have undergone great changes. In the past, manual meter reading not only requires additional manpower and time, but also brings inaccurate result and other problems. It is difficult to know the current entire electric quantity and the current operating status. The data synchronization module designed in this paper is based on MSF. In this module, the central database server and the client each have a database. They only communicate with the local database and the data synchronization between them is finished by MSF. Therefore, the database synchronization module establishes a communication bridge between in the client and the central database server. If the data synchronization module cannot run properly, it will directly lead to the collapse of the entire wireless data collection system. This paper makes researches of MSF[1] and designs a database synchronization module on the basis of it, which achieves distributed database synchronization module and peer-to-peer database synchronization.

II. WIRELESS DATA ACQUISITION SYSTEM

Wireless data acquisition system consists of collector, central database server and data collection station. The system obtains the data of the smart meter through the collector and transfers it to the data acquisition station by ZigBee, and then synchronizes the data to the central database server by the data collection station. ZigBee is a technology with two-way wireless communication. It has short communication distance, low complexity, low power consumption, low speed and also low cost, which is mainly used for data transmission between electronic equipment with short distance, low power consumption and low transmission rate. As a network platform of wireless data transmission, it can be composed of 1 to 65000 wireless data transmission modules. In the equivalent network, every ZigBee data transmission module can communicate with each other, and the distance between each network node can be infinitely extended from the standard 75m[2].

In the wireless data acquisition system, the data synchronization modules in the client are mainly the data collected by the device and the synchronization of the server. The synchronization of data and server is mainly responsible for synchronizing data, such as the electric quantity of the smart meter that collected in the client side, to the server. If there is a network, the client will first collect the data stored in the local database, and then synchronously upload the data to the server. However, in reality, we cannot always keep the network available. When the network is unreachable, it is likely that there will be a variety of problems, resulting in inaccurate data. There are two cases of data conflict between databases.

Data conflict caused by sudden network disconnection during data synchronization: When the network is unreachable, the data collected from the smart meter in the client side will be saved to the local database. And when the network is reachable again, they will be synchronously uploaded to the server. If there is a lot of data in the local database to synchronize to the server and the network is interrupted during the process of synchronization, which will result in the data cannot be all updated to the server, it may bring data error.

Data conflict caused by modifying the same data in the server and the client: The conflict here mainly refers to an error occurred during the data synchronization when the client and the server are creating, updating, and deleting data, such as the usual constraint conflict (repeat the primary key), or the same line is changed on multiple nodes (clients) during synchronization, or the server removes the row while the other node has updated the row. These conflicts will bring great trouble to data synchronization, which may lead to inaccurate data, so the necessary detection and resolution have to be made.

III. IMPLEMENTATION MECHANISM OF DATA SYNCHRONIZATION MODULE

A. Brief Introduction of MSF

MSF is Microsoft Synchronization Framework launched by Microsoft, which is a fully functional synchronization platform[3]. It achieves the collaboration between application, services and equipment as well as offline access. Sync Framework provides a number of technologies and tools that can support roaming, sharing data, and acquiring data offline. By using Sync Framework, developers can build a synchronization ecosystem that integrates any application with any data in any storage area by using any protocol on any network.

In general, the basic issues needed to be resolved during synchronization are: source records changes and transfers the changed data to the destination, then destination detects and resolves conflicts, and applies the changes to the local. Among them, the most important problem solved by Sync Framework is how to record changes in data, how to detect the synchronization of data and how to deal with data conflicts. MSF uses metadata to document data changes and conflict detection, abstracting the issues that need to be addressed[4,5].

The synchronization process is as follows. First, when the source has the requirements of synchronization, it initiates a request to start a sync session and establish a connection to the destination[6]. Then, when the destination receives a request for a sync session, it finds its own metadata and sends the metadata to the source. After that, the source receives the metadata and compares it to determine the data that needs to be updated and sends it to the destination. When the destination knows the data that needs to be updated, it finds the corresponding metadata and performs conflict detection. After the detection, the destination requests the updated item data from the source and the source prepares and sends item data. After obtaining the updated data, destination updates the database and metadata, and finally ends the sync session[7,8].

B. Brief Introduction of MSF

Sync Framework provides an intuitive and flexible API which allows synchronization between databases and makes developer create an application program oriented towards the off-line collaborative plan. When using this API, you can use all or part of components offered by Sync Framework according to the practical requirement and architecture of your application program. Client-server, client-client and blending topologies are supported by Sync Framework. In client-server topology, all clients are synchronized with one central server while in client-client topology, all clients are synchronized with each other and altering among them need not be passed by the central server. Blending topology combines client-server topology with client-client topology. When using typical data synchronization plan, user uses the data by VPN connection or direct connection to the network access server. However, for most remote users, there are some disadvantages of this kind of solution as follows which cannot be neglected.

Network requirement: if users need to have access to the data, then remote equipment is required to keep accessing the server via network. Without network, users fail to access central database and cannot acquire valid data.

Data access speed: in a typical client-server network environment, users with high speed network can access information rapidly. However, in the context of poor network signal or slow network speed, every time users have to upload and download the data they need because the equipment cannot store data for a long period, which causes waste of time and incomplete data as well as data corruption.

A single fault point: users cannot acquire the data they need when server database goes wrong and is unusable, because the data is stored on the server.

Data acquisition requires communication with server. With the increasing number of users, load of server is becoming higher and higher. To reduce the load, extra expense may be needed.

After the analysis on the problems above, solution of occasional connection is adopted in the system[9] of this paper. Data is stored in the local database by client in accordance with the data on server by data synchronization, which helps users access the data without constantly accessing the server through network. With this kind of solution, users can access the data at any time, which not only reduces the load of server, but also accesses the data at the speed of local client, saving the time of data access. In Fig. 1. below, multiple clients connect to one central server and occasional connection is adopted in database synchronization in the client-server topology.
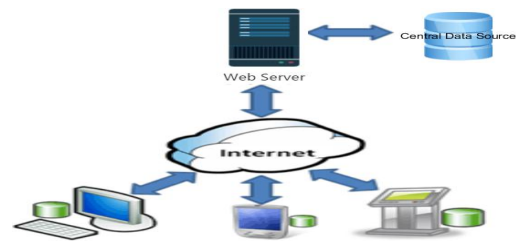

Fig. 1.   Occasional Connection Adopted in Data

Synchronization Sync Framework provides powerful design and flexible method for synchronization. The advantages of the design are as follows.

First, a synchronization ecosystem integrates multiple data source and is extensible. Second, a managed API applied in all components and a native API applied in selected components. Third, conflict resolution applied in the automatic and custom solution. Fourth, filter of subset applied in data synchronization. Fifth, the compact and efficient metadata model realizes the synchronization for all participants and meanwhile, does not lead to great alteration in data storage area.

C. Structure Selection of Database Synchronization in MSF

In the off-line and collaborative plan, Sync Framework is available for two-tier and n-tier architectures. In these two[10] architectures, activities take place in "synchronization session": "synchronization controller" communicates with two "synchronization providers" so as to search alteration and apply the alteration to all databases. These two providers apply

"synchronization adapter" in each table and the specific SQL orders which the tales requires. Some providers expose the adapters so that you can customize these orders. Other providers generate adapters automatically so as to reduce the complexity related to synchronization of specific storage area.

IV. DESIGN AND IMPLEMENTATION OF DATA SYNCHRONIZATION MODULE

A. Data Synchronization to Server

Sync Framework is available for both two-tier architecture and N-tier architecture. In these two kinds of architectures, activity takes place during the sync session: "sync orchestrator" will communicate with two "sync providers" in order to retrieve changes and apply them to all databases. These two providers will apply "sync adapters" to each table as well as the specific SQL commands needed. Some providers expose the adapters so that you can customize those commands. Other providers automatically generate adapters so as to reduce the complexity of correlation with the synchronization of specific storage area.

In Fig. 2, two-tier architecture is shown: all components of one sync session reside on local computer and there is no direct connection of databases between local and remote computers. During the sync session, the local computer is the computer that initiates the synchronization. If synchronization from multiple computers was wanted to be initiated, each computer must contain all the components shown in the diagram.

Fig. 3 shows an N-tier architecture that requires additional components, including components on the remote computer; and the connection to the remote database is now handled by proxies on the local computer and services on the remote computer. The proxies and the services here must be implemented by the application developer. An example is provided to reduce the complexity associated with this task. For two-tier architecture, if synchronization from multiple computers was wanted to be initiated, each computer must contain all the components shown in the diagram.
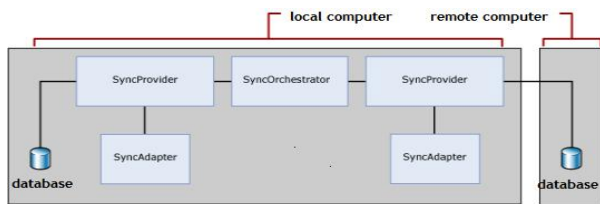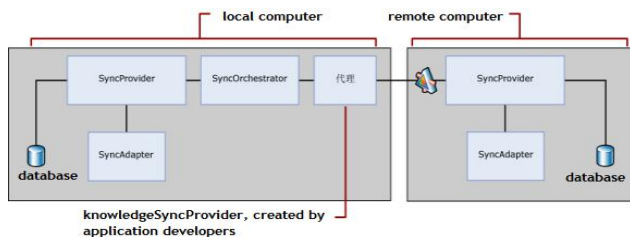

Fig. 2.   Two-tier Architecture

Fig. 3.   N-tier Architecture

In this paper, the wireless data acquisition system uses two-tier architecture to achieve data synchronization. Main implementation codes are as follows:

LocalDataCacheSyncAgent SyncAgent = new LocalDataCacheSyncAgent();
    SyncAgent.CommunicationList.SyncDirection = SyncDirection.UploadOnly;
    SyncStatistics syncStats = SyncAgent.Synchronize();
    Msg.Items.Add("Total Changes Downloaded from Server [including changes failed to be applied on client]" + syncStats.TotalChangesDownloaded);
    Msg.Items.Add("Total Changes Uploaded to Server [changes failed to be applied on server]" + syncStats.TotalChangesUploaded);
    Msg.Items.Add("Changes Uploaded from Client and Applied on Server" + syncStats.UploadChangesApplied);
    Msg.Items.Add("Changes Downloaded from Server and Failed to be Applied on Client" + syncStats.DownloadChangesFailed);
    Msg.Items.Add(" Sync Start Time " + syncStats.SyncStartTime.ToLongTimeString());
    Msg.Items.Add("Sync Complete Time" + syncStats.SyncCompleteTime.ToLongTimeString());
    TimeSpan time = syncStats.SyncCompleteTime.Subtract(syncStats.SyncStartTime);
    Msg.Items.Add("Total Sync Time" + time.TotalSeconds.ToString() + "Second");
    SyncAgent.Dispose();

B. Synchronization Module Testing

First of all, data synchronization between client and server should be tested in the context of network unimpeded. Data synchronization on client is shown in Fig. 4. and data synchronization on server is shown in Fig. 5. The sync time of a piece of data is shown in Fig. 6.

Fig. 4.   Data Synchronization on Client

Fig. 5. Data Synchronization on Server

Second, data synchronization should be tested in the context of sudden network interruption in the process. Data synchronization adopts the method of synchronizing tables in single affair. Therefore, if all data cannot be synchronized to the target server and the network is down suddenly, data synchronization will stop implementing and all data will be reserved in client database, waiting for next data synchronization.

Total Changes Downloaded from Server [including changes failed to be applied on client]:0
Total Changes Uploaded to Server [including changes failed to be applied on server]:1
Changes Uploaded from Client and Applied on Server:1
Changes Downloaded from Server and Failed to be Applied on Client:0
Sync Start Time:15:25:37
Sync Complete Time:15:25:37
Total Sync Time:0.1890231Second

Fig. 6. Sync Time of a Piece of Data

V. SUMMARY

Data acquisition system is necessarily confronted with redundancy data volume and frequent communication. Therefore, management and communication of database play an important role in the acquisition system. According to the system demand, a kind of database synchronization module is designed which is put forward based on MSF framework in this paper. This module realizes data synchronization by setting MSF framework.

However, there are many shortcomings in data synchronization module, which needs improvement and perfection for future use.

Firstly, in the context of network unimpeded, acquired data can be synchronized at once after being stored in local database, which may cost many network resources and put pressure on the server. So, a solution should be figured out to keep a balance between data synchronization and network resources.

Owing to limited resources, simulation test of data synchronization module is carried out only on two computers while stability of MSF framework cannot be tested. Therefore, more computers should be used in simulation test of data synchronization in future experiments so as to further optimize the program.

REFERENCES

[1] Life a Poem, Microsoft Sync Framework Basic 1: Introduction of Micro soft Sync Framework, http://www.cnblogs.com/lifepoem/archive/2012/0 3/06/2382158.html, 2012(In Chinese).

[2] jady3356, ZigBee Technology, http://baike.baidu.com/view/3085090.ht m?fr=iml, 2014.

[3] MSDN, Benefits of Using Sync Framework, http://msdn.microsoft.com/ zh-cn/library/dd918617.aspx, 2008(In Chinese).

[4] FLYabroad, Microsoft Sync Framework Series 2: Downloading, Docum ent and Example of Microsoft Sync Framework, http://www.cnblogs.co m/flyabroad/archive/2008/05/17/1201332.html, 2008(In Chinese).

[5] FLYabroad, Microsoft Sync Framework Series 4: Basic Conception of Microsoft Sync Framework, http://www.cnblogs.com/flyabroad/archive/ 2008/05/20/1203546.html, 2008(In Chinese).

[6] FLYabroad, Microsoft Sync Framework Series 5: Metadata of Microsoft Sync Framework, http://www.cnblogs.com/flyabroad/archive/2008/05/2 2/1205130.html, 2008(In Chinese).

[7] FLYabroad, Microsoft Sync Framework Series 6: Knowledge of Micros oft Sync Framework, http://www.cnblogs.com/flyabroad/archive/2008/0 5/23/1206172.html, 2008(In Chinese).

[8] Life a Poem, Microsoft Sync Framework Basic 3: Metadata and Sync Pr ocess of Sync Framework, http://www.cnblogs.com/lifepoem/archive/20 12/03/08/2385290.html, 2012(In Chinese).

[9] Life a Poem, Microsoft Sync Framework Database Synchronization 1: Brief Introduction of Database Synchronization, http://www.cnblogs.co m/lifepoem/archive/2012/03/09/2387151.html, 2012(In Chinese).

[10] MSDN, Architecture and Classes for Database Synchronization, http://m sdn.microsoft.com/zh-cn/library/bb902829.aspx, 2008(In Chinese).