A Data Synchronization Method Oriented to Custom Hierarchical Multi-node System

Jing Zhang

Changchun Vocational Institute of Technology
Changchun, China
e-mail: zjzqy2@foxmail.com

Abstract—**In the modern enterprise system, data synchronization programs need to support data cloud storage to ensure data consistency, integrity, and security between two devices. The paper addresses to some observation network system as a hierarchical multi-node system. In its application development process, the synchronization functions in existing database management system cannot meet its system requirements. A kind of data synchronization method oriented to custom hierarchical multi-node system was brought out. The program floe and synchronization task queue structure to implement the method were also given.**

Keywords-multi-node system; data synchronization; synchronization task; task queue

I. INTRODUCTION

After observation network of some monitoring center been completed, many monitoring stations, regional monitoring centers cover in wide range area are established. These stations and monitoring centers connect to each other in accordance with their level, thus form a custom hierarchical multi-node system. The paper addressed to such hierarchical multi-node network system for application development. The system requires subordinate node to duplicate and synchronize data to superior nodes so that data in superior nodes accurately reflect status of subordinate nodes. The superior nodes can also statistic, index and query all information of subordinate nodes. There are several problems accounted by data synchronization module in the task implementation process as following.

The traditional database management system has higher requirements on operation users to complete data synchronization [1, 2]. Limited to computer levels of the work staff of monitoring stations and monitoring centers, the replication function provided by operation database management system may have problems. Therefore, the data synchronization task from monitoring stations to superior nodes should be simplified as possible to reduce engagement and actions from users. Station nodes and regional center nodes will adapt different database management system in practical applications according to different tasks and throughput. It needs to implement data automatic synchronization and replication among different database management systems. When subordinate nodes sync data to superior nodes, it may exist multiple nodes ask for synchronization to many nodes at the same time, resulting in slow response or communication block from superior nodes. After system restart, it can automatically join last unfinished task to ensure data synchronization process timely and automatically. Aiming at above data synchronization problems, the paper gives a method for data synchronization in multi-node system. The paper is organized as follows: section 2 analyzes the solutions and processes; section 3 gives specific synchronization method; section 4 concludes our work.

II. SOLUTIONS AND IMPLEMENT PROCESS

A. Solutions

It is difficult for data replication function provided by database management system to meet requirements of above problems. To address such problem, we developed a program to implement synchronization after repeated modifications and adjustments in the implement process.

The data synchronization process can be divided into two parts:

Firstly, synchronization task notification. The initiator is subordinate nodes, which operates according to following steps.

Step1: Subordinate nodes data conduct update (add/delete) operations.

Step2: Add needed tasks for synchronization to sync queue.

Step3: Subordinate nodes notify synchronization tasks to superior nodes in the task list.

Step4: Subordinate nodes receive synchronization tasks and save it locally.

The second is data synchronization. The initiator is subordinate node, which initiate operations according to following steps.

Step1: Superior nodes send data request to subordinate nodes in order based on local synchronization task queue.

Step2: Subordinate nodes response request and return data.

Step3: Superior nodes complete data synchronization.

The data synchronization task from subordinate node to superior node can be achieved by data synchronization module deployed on subordinate nodes and superior nodes according to above idea. The synchronization process needs not intervention from users. The subordinate node and superior node can use different database systems. The requesting data from the upper node initiates the process to

avoid multiple nodes simultaneously send data may cause network congestion to fully meet the system requirements for data synchronization.

B. Implementation Process

The method does not require complex algorithm and it is relatively easy to implement. It needs both sides work together. It should consider some details in order to avoid lower node and the superior node synchronization conflict occurs during data [3-5]. The implementation processes of two parts in data synchronization process are provided with flow diagram.

1) Synchronization task notification implementation process

The implementation flow of synchronization task notification is shown in Fig. 1 and Fig. 2. It should be noted to achieve such function that data update task about same information only records once in the task queue. Any modification before data synchronization only record once, thus greatly reduces frequency to send information and request for synchronization. The information to send synchronization task is very simple. Even multiple nodes send information simultaneously, the network will not congest.
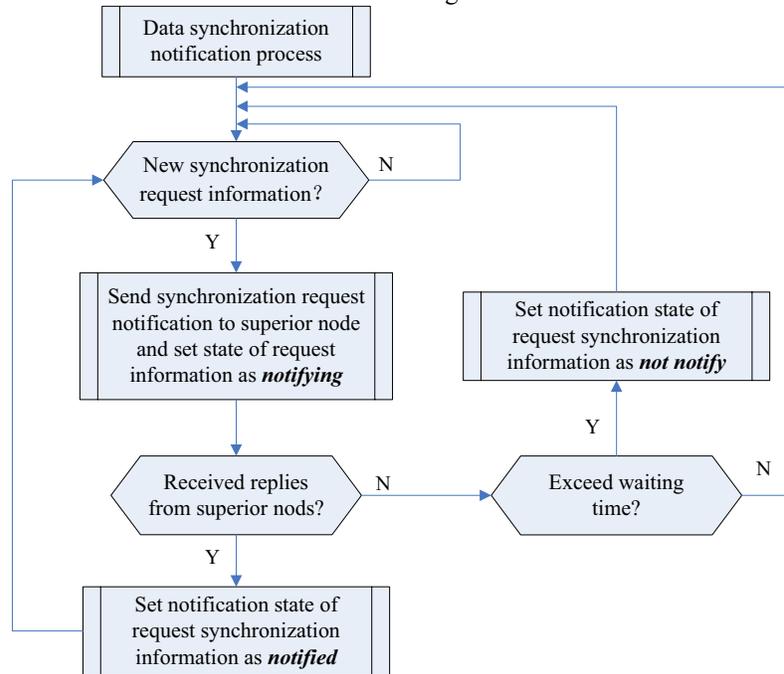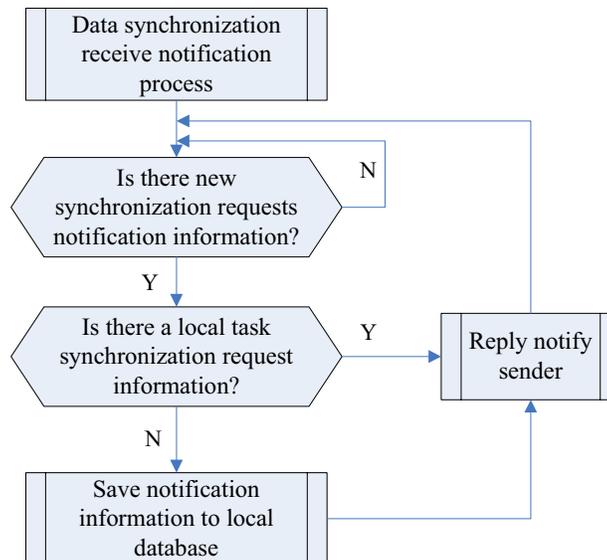

Figure 1.   Send task notification process.

Figure 2.   Receive task notification process.

2) Data synchronization implementation process

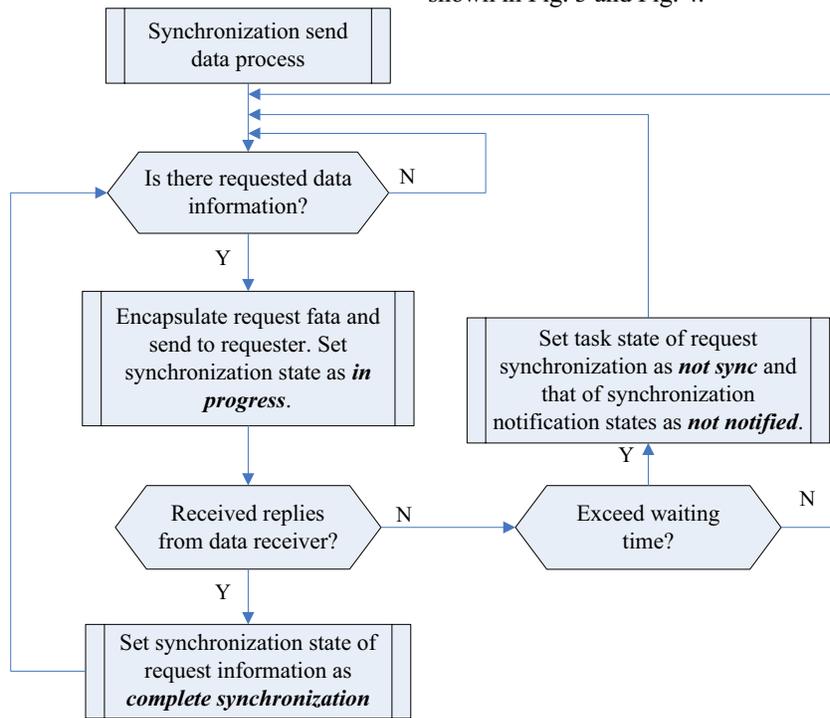The implementation process of data synchronization is shown in Fig. 3 and Fig. 4.

```
Synchronization send data process
  ↓
Is there requested data information?  —N→
  │Y
  ↓
Encapsulate request fata and send to requester. Set synchronization state as **in progress**.
  ↓
Received replies from data receiver?  —N→  Exceed waiting time?  —N→
  │Y                                          │Y
  ↓                                           ↓
Set synchronization state of request information as **complete synchronization**

Set task state of request synchronization as **not sync** and that of synchronization notification states as **not notified**.
```

Figure 3.   Send synchronization data process.

```
Synchronization request data process
  ↓
Is there data synchronization task?  —N→
  │Y
  ↓
Send request data and set local request synchronization task into **in progress**.
  ↓
Whether received returned data?  —N→  Exceed waiting time?  —N→
  │Y                                     │Y
  ↓                                      ↓
Reply data sender and set local synchronization task state as **complete synchronization**

Set synchronization state of request synchronization task as **unsynchronized**
```
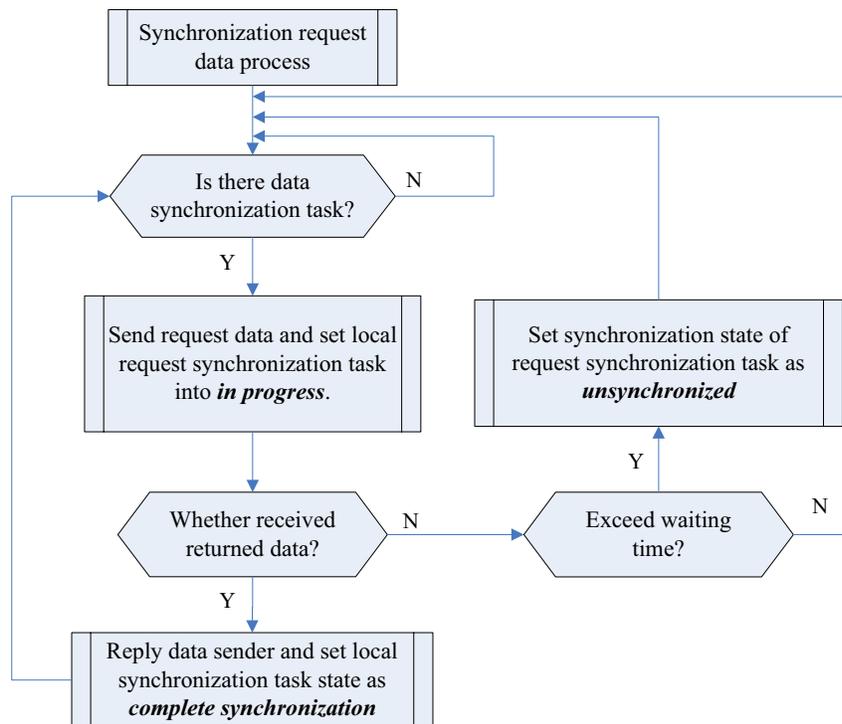
Figure 4.   Request synchronization data process.

The data synchronization process firstly send data request from superior node to subordinate nodes with data update request. The superior node can handle the situation of the local tasks at the appropriate time to the respective

intermediate node sends a data request command in order to avoid data congestion situation.

To ensure data synchronization information can be correctly performed at superior nodes, the data sender must receive the data requester's response to confirm that the data synchronization is successful. Otherwise, it need to re-set synchronization task status, making the task of re-notification and data synchronization process. Although this may result in duplication data synchronization process, but does not cause two sides appear contradictory information. It can also ensure that data is properly synchronized.

III. SYNCHRONIZATION METHOD DESIGN

It is critical of synchronization task list to implement data synchronization process. Its structure directly affect whether there is conflict or contradiction in each stage of synchronization process or in the result.

A. Subordinate Node Synchronization Task List

The synchronization task list data of subordinate node should contain basic information as following:

(a) Flag code to show uniqueness of information. The superior node obtains request data information in synchronization process from this flag code.

(b) Data type flag. The data information can be obtained from this flag in different list.

(c) Primary key in data list. Through such item, it can determine some data record in the corresponding data list.

(d) Task generation time. With it, we can sort task and notify superior node according to time order of tasks.

(e) Notification status information. From this item, we can identify whether the synchronization task notify superior nodes. Its status can be divided into three types, namely not notify, notifying and notified.

(f) Send notification time. After sent notification message to superior nodes, the item should be set as current time. With this item, the notification status should be re-set as not notified when the waiting time is exceeds while not received reply.

(g) Synchronization task completion status information. It can identify whether synchronization task been completed. Its state can be divided into three types, namely not synchronized, in progress and completed.

(h) Synchronization data start time. When superior node requests for synchronization data, it should set this item. If the time exceeds regulated time without receiving node replies, set this item as not synchronized without notification.

B. Superior Node Synchronization Task List

The superior node synchronization task list data should contain basic information as following:

(a) Flag to identify synchronization task.

(b) Synchronization task information flag. It corresponds to (a) in synchronization task list data.

(c) Task generation time. Using it, request for synchronization data in time order.

(d) Subordinate node flag. With this item, it can differentiate synchronization data task come from which subordinate node.

(e) Synchronization task status information. With this item, it can identify whether synchronization task been completed. Its status can be divided into three types, namely not synchronized, in progress or completed synchronization.

(f) Synchronization data start time. When the data synchronization process exceeds regulated time but not completed, discard this synchronization process and set synchronization task status information as not synchronized.

IV. CONCLUSION

Because both data replication sides are self-developed software, they can operate different database systems. Both sides to conduct data replication need not equipped with same database management system, thus easily achieve data duplication synchronization among heterogeneous database. The manner that subordinate node notify superior nodes with synchronization task and then waiting for synchronization data request from superior node can avoid communication block or system crash caused by multiple subordinate nodes sync data to superior node at the same moment. In addition, the implementation method considers communication intervention in the synchronization process, which can avoid contradiction or conflict in both task lists basically.

REFERENCES

[1] Min Jiang, Ya-bo Dong and Dong-ming Lu, "A Method of Data Synchronization Based on Agent," Journal of Jiangnan University (Natural Science Edition), vol. 4, no. 6, pp. 598-605, 2005.

[2] Kun Ding, Hao Yan, Xing-chun Dao, "Research of data synchronization technology in distributed database," Journal of Naval University of Engineering, vol. 16, no. 5, pp. 100-104, 2005.

[3] Shu-guo Fa, "The Design and Implementation of Data Synchronization," Journal of Liaoning Provincial College of Communications, vol. 6, no. 2, pp. 18-19, 2004.

[4] Chun-hong Gong, "Research on the Synchronous of Mobile Database on Replication," Computer and Digital Engineering, vol. 35, no. 5, pp. 66-68, 2007.

[5] Hu Xie, Jing-fei Zhu, "Design and Application of Data Services Model in Configuration Software," Science Technology and Engineering, vol. 9, no. 6, pp. 1600-1603, 2009.