

Research on the Key Technology of Cement Enterprise Data Synchronization

Shuaihao Huang, Xiaohong Wang and Shaohong Jing
School of automation and electrical engineering, University of Jinan, Jinan, China.
1287018471@qq.com, cse_wxh@ujn.edu.cn, cse_jsh@163.com

Keywords: Cement, Cloud platform, Replication, Data synchronization.

Abstract: As the level of industrial information continues to improve, cement production enterprises have basically achieved continuous acquisition and storage of front-line production data. However, decentralized production sites and relatively independent management modes are not conducive to the improvement of overall production efficiency and the realization of energy conservation and emission reduction. We can build a cement enterprise cloud platform and transfer real-time production data which from different cement manufacturing enterprises to the cloud platform. After that the cloud platform can make summary and analysis and data mining of the data and offer optimization for cement production control algorithm, and also provides rationalization proposals for arrange production tasks, upgrades production processes and reduces pollutant emissions, and this function can be achieved through SQL Server replication and related key technologies.

1 INTRODUCTION

SQL Server replication technology requires that each enterprise database server in the cloud platform is connected to the same logical local area network. SQL Server can heterogeneous connect with multiple data sources to provide powerful support for complex distributed database management systems. In order to introduce the synchronization solution put forward by this article as detailed as possible, all the production enterprises adopt the SQL Server database for storage. This article will introduce the choice of synchronization mode, and the creation of synchronization and a series of personal settings and how to reduce the number of database access and insert frequency solution, and the implementation of the corresponding strategy to ensure data consistency and real-time synchronization of both data when the data congestion occurs in the transmission process.

collected by the DCS process and meter data measured by the ammeter, and their update frequency is very fast, respectively, once a second and once ten minutes, the resulting amount of data with real-time requirements is very large so the choice of synchronization is particularly important. There are four kinds of synchronization schemes provided by SQL Server: snapshot replication, transactional replication, peer-to-peer replication, merge replication. Snapshot replication is in accordance with the data at a particular moment in time to distribute the data, do not monitor the data update, and transactional replication just make up for this shortcoming; In peer-to-peer replication topology, the relationship between the server is a peer relationship rather than a hierarchical relationship; Merge replication allows each site to make independent changes to the copied data, and merge the changes or resolve the data conflicts when needed later, and the data storage topology and data synchronization are shown in Figure 1 and Figure 2:

2 SYNCHRONIZATION OPTIONS

Replication technology is to transfer database objects from one database to another and then maintain the final consistency of the two sides, cement production process involves the data

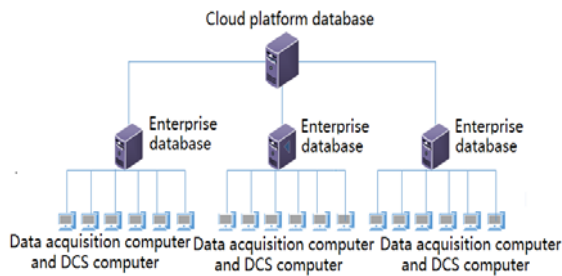


Figure 1: Topology of data storage

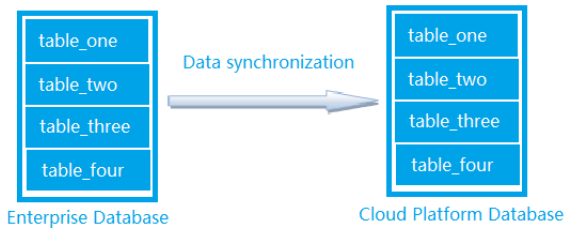


Figure 2: Data synchronization specific configuration

The use of snapshot replication on cement production data is obviously not applicable because on the one hand, as the amount of data increases, the space occupied by the snapshot file itself is very large, and on the other hand, a large amount of bandwidth will be taken up when the snapshot replication start work . In the same time, the problem is also reflected in peer-to-peer replication and merge replication, so incremental update of the newest live data is clearly the best option given the same initial data set and architecture for both enterprise and cloud platform databases, so this article proposes the use of transactional replication method for data synchronization, the principle shown in Figure 3:

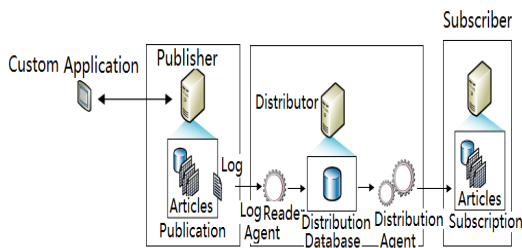


Figure 3: Transactional replication schematic

3 BUFFER SETTINGS

A good network status is a good guarantee when the data is synchronized between the enterprise and the cloud platform, but the cement industry is often far away from the urban,the network often has

congestion or instability. A healthy data synchronization system must be able to handle these unexpected situations perfectly. Therefore, we propose that when these conditions occur, a special area will be set up to store the temporarily forwarded transactions. After the network regains its smooth flow, the deferred transactions will be resumed and distributed, as shown in Figure 4 :

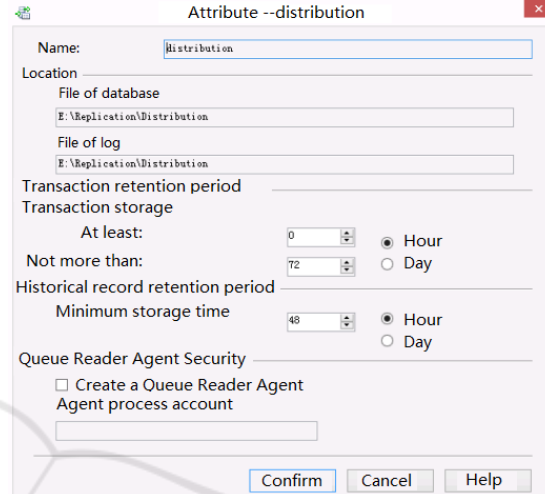


Figure 4: Data synchronization buffer settings

The specific realization is to create a special folder in the distribution database to read through the log reader agent to read over the transaction at the same time as the distribution server belongs to the logical server its physical storage location needs to be based on the actual requirements of business and consideration of the hardware situation of cloud platform , although the cloud platform server also equipped with excellent hardware and peripheral equipment generally , but taking into account the possible network storm and data security and storage pressure considerations, decided to set the distribution server in each enterprise,and the length of the retention period needs to be based on the type of synchronization data and update frequency, combined with the actual production of cement enterprises in terms of 72 hours of cache time will not bring pressure on the storage space and a long time transaction stack will not affect the final data consistency, of course, the premise is that the relevant agents did not stop working.

4 CREATE SYNCHRONIZATION JOB

Earlier we have already mentioned that transactional replication is mainly achieved by the snapshot agent, the log reader agent and the distribution agent, and in fact is the working result of the log read job, snapshot job and distribution job corresponding to each publication. To a certain database, in terms of all the tables it contains, is divided into different groups called nodes. After all the creation and subscription of a node are completed, the corresponding three jobs are created accordingly. It just have only one log reader job in a certain database ; for each identified job in terms of mainly including steps, plans, alerts, notifications, goals five steps in the process you can create a specific operation of the agent and specify each steps to run the next step after the success of the operation step, you must specify the database to be operated on and the specific command to be executed, the specific commands through a series of SQL statements to implement procedures, such as the steps on the log reader to run the agent statement is as follows:

- Step 1: `sp_MSadd_logreader_history @perfmon_increment = 0, @agent_id = 34, @runstatus = 1, @comments = N 'Start the agent.'`
- Step 2: `-Publisher [QTX] -PublisherDB [Db_02_01_Cement01] -Distributor [QTX] -DistributorSecurityMode 1 -Continuous`
- Step 3: `sp_MSdetect_nonlogged_shutdown @subsystem = 'LogReader', @agent_id = 34`

The above steps can be achieved log reader agent startup, running, and close when no new data for a long time, although the above steps can basically meet the data synchronization requirements, but when the production process stop work due to mechanical failure or routine maintenance for a longer time the agent will automatically shut down when that happens, and the agent will not start automatically when the device needs to continue to synchronize data as it is restarted. Therefore, it will cause the synchronization failure. This is something we do not want to see. So what we need is that the agent can automatically started after a transaction is resumed or whether or not there is logs, the agent stays in working condition for a long time. Obviously, the latter is more satisfactory for the cement industry which will not stagnate for a long time, the basic flow Figure 5 as follows:

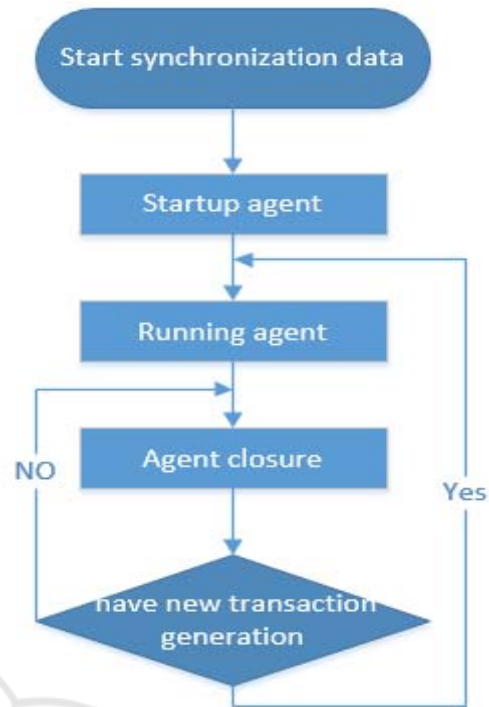


Figure 5: Sync agent job run steps

5 SYNCHRONIZATION EFFICIENCY OPTIMIZATION

The basic requirement of data synchronization is to realize the consistency and real-time performance of enterprise-side data and cloud platform data. Once data is collected from the data acquisition system, the corresponding program must be connected to the database and then inserted into the data to be finally break off the connection to the database. Although the cloud platform does not need to connect to and disconnect from the database, the frequency of once a second DCS data acquisition adds an additional burden to the normal operation of the enterprise data acquisition system and the storage of both databases. Because the continuity and stability inherent in industrial production data tend to make a significant correlation between current data and nearby data, so a brief lag in data compared to industrial sites does not affect the overall performance of the entire production system accurate reflection, this paper proposes that take the DCS data which would be stored once a second "gather parts into a whole", but not changing its acquisition frequency, that is, temporarily stored together the data which send out once a second to form a chronological queue

every ten seconds then centralized insert into a database to ensure the integrity of the data and reduce the additional burden on the database while reducing continuous to occupy on the network bandwidth ,to make the data acquisition and synchronization systemsthe work in high efficiency. the process as shown in Figure 6.

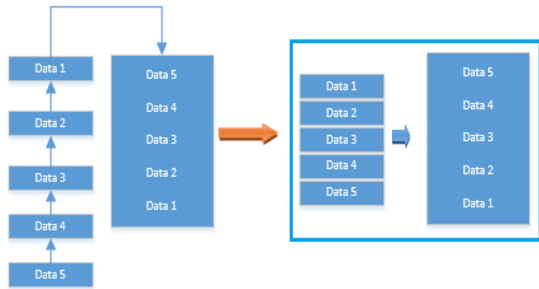


Figure 6: Storage optimization diagram

6 SYNCHRONOUS MAINTENANCE

With the help of SQL Server data synchronization technology and a series of strategies and measures we have adopted which to ensure synchronization stability as a whole, we can ensure the consistency of enterprise data and cloud platform data as well as the robustness of the whole system, though with the acquisition parameters increase or decrease the inevitably encounter need to add or reduce the synchronization of the table or change the table structure , in the past the above situation can only be solved by withdraw the corresponding subscription and release,that is to re-establish the corresponding synchronization after the completion of the change which will result in the data difference between publishing and subscribing. However, the missing data can be filled after the synchronous reconstruction,this process depends on the Visual Studio database comparison tool and the SSIS solution.The attendant drawbacks are that the larger the amount of data, the more time it takes to trap, and also more of the occupancy of hardware and bandwidth ,But it at least avoids the loss of data. In this process, we just fill in missing data during the publish subscription process, so do not use snapshots to reinitialize the subscription, which will delete the data that has been synchronized to the cloud platform,this is what we do not want to see.

Of course, the above approach is a viable way to ensure the smooth running of data synchronization;

however, we often hope that the addition, deletion or change of the structure of the table without deleting the existing subscription ,after deeper exploration and learning, that can be achieved by change its table structure directly in the publishing side when need to change for the table on synchronization,because the transactional replication is usually begin to work from issued the database object and the data snapshot, after the initial snapshot is created and then the changes on data or schema of released are usually (in near real time)passed to the subscriber.

Data changes will be applied to subscribers in the order in which they originate on the publisher and the boundaries of the transactions, so transactional consistency can be guaranteed within the publication; adding or deleting a table in the node of synchronization process requires execute SQL statements about procedure `sp_addarticle` and `sp_addmergearticle` ,and also can be set in the properties of the synchronization node, it only displays all the table information contained in the node when select the "project" option, then only need to remove the "only display list selected items ",it will displaying all tables that have primary key in the database that the node belongs to, choice the new table which need to be released to achieve the release of the new table, without having to stop the data synchronization when add a table (or another object) of a released node, the above method, empathy can also delete the release table.

7 RESUME FROM BREAKPOINT

Through the series of measures above, most of the obstacles that may be encountered in the data synchronization process have been cleared up.However, as the field data is continuously inserted into the enterprise database, the corresponding data files and log files of the database will be larger and larger.We can take measures such as file partitioning to improve the database access rate,but we can not ignore the hidden dangers brought by the data insertion efficiency and success rate reduce,this paper proposes to create a cache database on the data acquisition computer to store the data that fail to insert into enterprises database,and then insert these temporary data into the enterprise database when have low presure.Which used to cache data table_SendBuffer table, the structure mainly includes two columns: one is the time column,and the second is SQL statement about the fail insertion data,and make the

time as the primary key also the column to establish an ascending clustered index in order to ensure that missing data can be filled out in an orderly manner and deleted in the table_SendBuffer table after padding, if the fault occurs in the network between the computer and the enterprise after the failure data is inserted in the enterprise database, the server can not determine whether the insertion is successful or not, then the above statement needs to be executed again. The corresponding row is deleted in the table_SendBuffer table after get the information about insertion of a repeated fault, as shown in Figure 7.

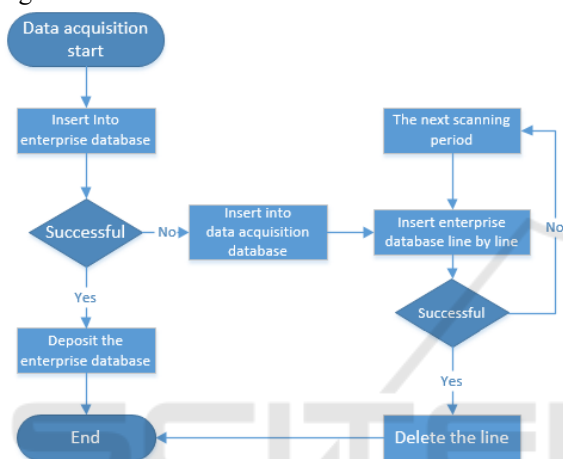


Figure 7: flow chart of resume from breakpoint

8 CONCLUSIONS

This article is devoted to establishing a cloud platform for cement enterprises to achieve the collection of data from multiple cement production enterprises ,and conducting a series of excavations and analyzes on the basis of this to provide a strong data support force for the optimization and upgrading of various aspects involved in cement production ,and the operation of the entire cloud platform is no need to participate in which specific manufacturing enterprises or specific produce processes and data acquisition and storage of , thus ensuring the independence and integrity of the cloud platform also avoids the additional occupancy of resources between each other ,this paper proposes to use SQL Server transactional replication technology as a bridge to establish real-time synchronization of data between the various specific production enterprises to the cloud platform to maintain the consistency of data on both sides. After establishing the synchronization, a series of measures was taken to

protect the data synchronization system stable and healthy operation such as buffer settings,create a synchronization job,synchronization efficiency optimization,synchronous maintenance ,resume from break point and so on . And it also avoids the preparation of specialized synchronization program to bring additional work, reduces the over-dependence on the network status, while reducing the probability of synchronization error and easy to maintain, at the same time, this set of synchronization system have good portability and versatility and also provides reference and inspiration for other similar forms of synchronization requirement.

ACKNOWLEDGEMENTS

This work is supposed by The Research Fund for the Taishan Scholar Project of Shandong Province of China and also supposed by Research and Application Demonstration of Key Technology of Intelligent Plant (2015ZDXX0101F01) and Research on Intelligent integrated optimization control technology of cement production process and industrialization (2014CGZH0601).

REFERENCES

- Jun Wang, Mu Hu, Qingqiang Meng, Xuehai Gu, 2017. mobile data synchronization conflict preprocessing and resolution strategy research. Information technology.
- Yongyong Chen, Zhenfei Wang, Jiajing Zhang, 2017. a method of data synchronization between air traffic automation systems. Wireless interconnection technology.
- Yingfan Xu, 2017.SQL database design related content. Network security technology and applications.
- Peifeng Yin, Jingyu Zhang, 2017. based on digital protection of intelligent substation sampling data synchronization technology application. Industrial instrument and automation device.
- Changping Wang, 2017. research on spatial data synchronization based on the "spatial information fingerprint of element objects". The geographic information world.
- Ning Liang, Xiaoqiang Shan, Die Sun, 2017. database synchronization technology research. Industrial control computer.