EAI Endorsed Transactions
on Collaborative Computing
Research Article EAI.EU

VTWM: An Incremental Data Extraction Model Based on Variable Time-Windows

Weixing Jia[1], Yang Xu[2], Jie Liu[3] and Guiling Wang[1,*]

[1]Beijing Key Laboratory on Integration and Analysis of Large-Scale Stream Data, School of Information Science and Technology, North China University of Technology, No. 5 Jinyuanzhuang Road, Shijingshan District, Beijing 100144, China
[2]Tianjin E-Hualu Information Technology Co., Ltd, No.1 Tianhua Road, Balitai Industrial Park, Jinnan District, Tianjin 300350, China
[3]Beijing Yidian Wangju Technology Co., Ltd, No. 30, Shixing Street, Shijingshan District, Beijing 100103, China

Abstract

Continuously extracting and integrating changing data from various heterogeneous systems based on an appropriate data extraction model is the key to data sharing and integration and also the key to building an incremental data warehouse for data analysis. The traditional data capture method based on timestamp changes is plagued with anomalies in the data extraction process, which leads to data extraction failure and affects the efficiency of data extraction. To address the above problems, this paper improves the traditional data capture model based on timestamp increments and proposes VTWM, an incremental data extraction model based on variable time-windows, based on the idea of extracting a small number of duplicate records before removing duplicate values. The model reduces the influence of abnormalities on data extraction, improves the reliability of the traditional data extraction ETL processes, and improves the data extraction efficiency.

Keywords: change data capture, incremental data extraction, timestamp, ETL

Received on 27 August 2020, accepted on 06 September 2020, published on 09 September 2020

doi: 10.4108/_____

*Corresponding author. Email: wangguiling@ncut.edu.cn

1. Introduction

In enterprises or government departments, due to the different development times and different development agencies, there are often multiple heterogeneous information systems running on different hardware and software platforms at the same time. These systems have some features such as independence from each other, the fact that it is difficult to share data and so on. These features raise the challenge to build data warehouse for data analysis. ETL (Extract-Transform-Load) is one of the main technical means to solve this problem [1]. Data integration through ETL solves the problem of difficult integration of heterogeneous data, and realizes the integration and sharing of the data between different departments and different system. At the same time, ETL technology also reduces the difficulty and cost of constructing data warehouse.

At present, the data extraction methods can be roughly divided into two categories: full data extraction and incremental data extraction. The full data extraction is simple and direct. It is a one-time extraction of the relevant data of the source data system, equivalent to data migration or data backup. This method is suitable for the first extraction [2], which is not the focus of this paper. Incremental data extraction is the extraction of data that has changed in the source system. The key to incremental data extraction is how to capture the data that has changed. Time Stamping Mode, Triggering Mode, Snapshot Mode, and Log Mode are common Changed Data Capture (CDC) methods [3-8]. For the Time-Stamping Mode or Incremental Timestamp-based Data Capture Mode, it requires the existence of a time attribute columns in the source database table and uses these attributes columns to determine which data is incremental data. It applies to occasions less demanding real-time [3]. For the Triggers Mode, it requires

The general improvement model de-duplication the data by comparing the new change data and the entire target table data one by one. This approach is inefficient and there are two main reasons:

Firstly, there is a large proportion of ineffective comparison operations. Secondly, the efficiency of data matching is too low. In general, the earlier the storage of data, the less likely to change in the future. Based on the above assumptions, we filter data of the target table according to the time of entry of data into the database. We only select the data from the target database for a period of time before the occurrence of abnormal data rather than the entire target table data to compare with the new data to narrow the search space. It can significantly reduce the number of comparisons between data.

4. Implementation

From the definition of VTWM, we can see that the table structure of time table described in 3.1 cannot meet the requirements of the new model. Therefore, this paper redesigns the middle table and the name is CDC_TIME. The table structure is shown in Table 1. *TS_S* and *TF* constitute a source time window *TW_S*. *TS_T* and *TF* constitute a target time window *TW_T*. And *pre_time* represents the data maintenance lead time Δt and will not change once determined.

Table 1. Structure of CDC_TIME table

Field	Type	Explain
table_name	varchar(30)	Primary key, the target table name for the current record maintenance
TS_S	datetime	The source system extracts the start time of the record
TS_T	datetime	The start time of the target database extraction of records
TF	datetime	The current time of the system when the extraction process is started
pre_time	long	Data maintenance pre - time, unit: Second

The key for VTWM implementation is the maintenance of the time window. The maintenance algorithm for CDC_TIME table is shown in Algorithm 2.

Algorithm 2 Update algorithm for CDC_TIME table

Input: Record<TF, TS_S, TS_T, TABLE>, Timestamps information used to extract data from the source to the target table

Output: Record'<TF, TS_S, TS_T, TABLE>, Updated Record

```
1. begin
2.      if (Record equals null) then
3.              TS_S:=1970-01-01 12:00:00;
4.              TS_T:=sysDate;//system current time
5.      end if
6.      TF:=sysDate;
7.      start Job until finished ;//job: extraction of incremental data
8.      maxVal:=getMaxTime(TABLE);//the maximum value of the attribute column of the data update time from the target database table
9.      TS_S:=maxVal-pre_time;
10.     TS_T:=maxVal-pre_time;
11.     update Record;
12.     return Record;
13.end
```

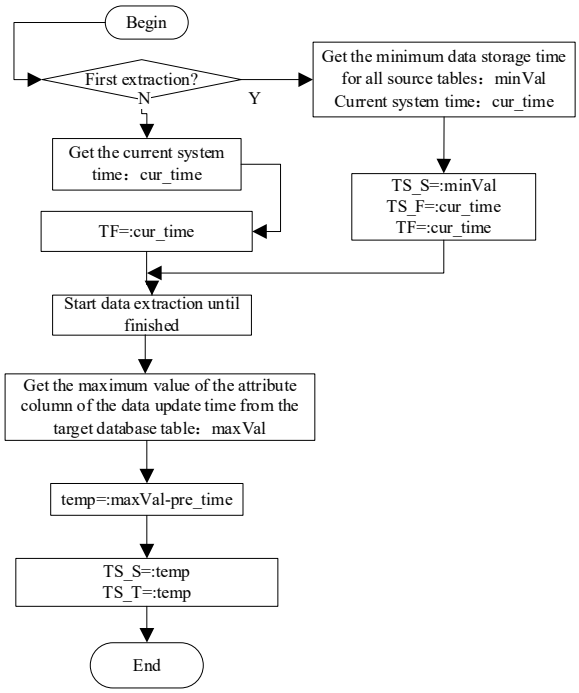

Figure 5. Processing flow of the update algorithm for CDC_Time table

For the first time extraction, all the data of all data sources needed to be extracted. Therefore, *TS_S* needed to be set as an earlier time for the first extraction. In this study, the initial value of *TS_T* was taken from 1970-01-01 12:00:00. In addition, the target database table was empty for first time extraction, so set the value of *TS_T* was the current system time. Except the first extraction, the value of *TS_S* and *TS_T* of each extracted data were the difference between the maximum value, *maxVal*, of the time of putting data into current target database table and the data maintenance lead time.

5. Experiment and Analysis

The traditional time-stamped incremental data capture model is deficient in reliability, which is improved accordingly in this paper. In order to verify the effectiveness of the design, this paper analyzes and compares the process shown in Figure 1 in terms of reliability and data extraction efficiency.

5.1 Experimental environment

This experiment was carried out in a 64-bit windows system. System version was windows 10 Professional Edition, and CPU was Intel i5-2400, quad-core, clocked at 3.10GHz. The memory was 4GB and the mechanical hard drive was 1TB.CPU, memory and hard disk free load were about 24%, 45% and 27% respectively. The source of the load was mainly generated by the operation of the system basic software. The database environment used in this experiment was Oracle11gRelease2 Standalone.

5.2 Comparison and analysis of reliability

In this experiment, we compared the data extraction of the Traditional Model, the General Improvement Model and VTWM. This experiment compared the three models from the exception of memory overflow and abnormal database connection. Table 2 is a comparison of the reliability of three models in the case of a memory overflow exception in the extraction process of 19880536 records in the A view. Table 3 is the reliability comparison of three models in the case of database connection exception in the data extraction process. The number of records to be extracted reduced to 3 million for avoiding the impact of memory overflow on the experimental results. Due to the amount of data to be extracted is small and there are very few anomalies in the extraction process. Therefore, the database connection abnormality is created by pulling out the network cable in the actual data extraction process.

Table 2. Reliability comparison under exception of out of memory

Model	First Time Abnormal	Number of Exceptions	Target table data volume	Extraction results
traditional model	7517000	∞	7517000	fail
General model	7528000	∞	7528000	fail
Model of this paper	7539000	2	19880536	success

Table 3. Reliability comparison under the exception of database connection

Model	First Time Abnormal	Number of Exceptions	Target table data	Extraction results
traditional model	1506000	∞	1506000	fail
General model	1428091	1	3000000	success
Model of this paper	1490000	1	3000000	success

In Table 2, the traditional model and the general improvement model cannot continue to extract data when a memory overflow exception occurs in the data extraction process. However, after the improved memory overflow exception in the data extraction process of the VTWM, the data extraction workflow can be started again to extract the data. In Table 3, the traditional model cannot continue to extract data after database connection anomaly occurs in the data extraction process. In the general improvement model and the VTWM, the data extraction process can be restarted after the database connection exception occurs. But the general improvement model takes a long time. Combining the above two points, the VTWM in this study is better than other two models in the case of anomaly.

5.3 Comparison and analysis of time performance

In order to further verify the performance of incremental timestamp-based data extraction model of variable window, this study compared the model in the data extraction time performance with the Traditional Model and the general improvement model. This model was based on reliability, so this experiment mainly compared the extraction efficiency of the three models in the process of data extraction in the case of abnormal circumstances. The experiment was carried out in two times. The first fixed target table data was 1 million and followed by an increase of the amount of source data. The second fixed source data was 1 million and followed by increasing the target table data volume. Due to the probability of abnormalities in the data extraction process was relatively small in reality, so this experiment used manual interference to create abnormalities. Two comparative experiments were performed abnormalities when the data amount of the target table and the source table were extracted to 450 thousand. The model that extracted data unsuccessfully due to abnormalities, the statistical time were calculated with the maximum. The comparison results were shown in Figure 6 and Figure 7.

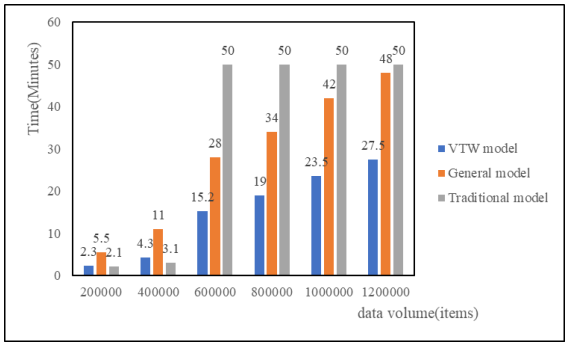

Figure 6. Performance comparison of three models in the case of fixed target table data

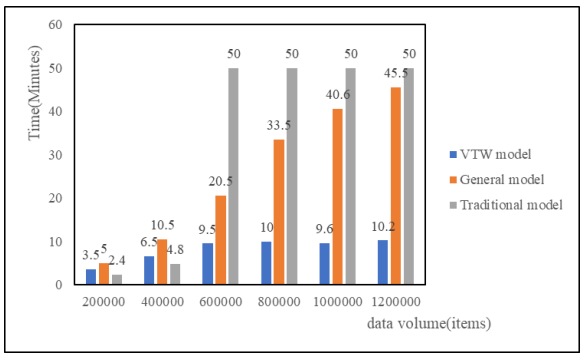

Figure 7. Performance comparison of three models in the case of fixed source table data

It can be seen from the comparative data in Figure 6 and Figure 7 that, although the extraction efficiency of the VTWM is lower than the traditional model if no abnormality occurs in the extraction process, the traditional model cannot continue to extract data when an abnormality occurs, while the general improvement model and the VTWM can continue to extract data and the efficiency of the VTWM is nearly doubled compared with the general improvement model. Comparing the extraction efficiency of the traditional model and the general improvement model, it is found that the main reason for the reduction of data extraction efficiency is the de-duplication operation. The VTWM is optimized on the basis of the general improvement model, and the amount of data comparison is reduced, thus improving the data extraction efficiency to a certain extent; meanwhile, as can be seen from the data in Figure 7, the VTWM avoids the problem that the extraction efficiency of the general improvement model gradually decreases as the amount of data in the target table increases.

6. Conclusion

In this paper, we propose VTWM, an incremental data extraction model based on variable time-windows. We study the problems occurred in the incremental data capture method based on timestamp change, and optimize the incremental timestamp-based data extraction method. Although, without anomalies, there are still gaps in the improvements of VTWM proposed in this paper when compared to the traditional extraction model, VTWM reduces the impact of anomalies on data extraction and improves the reliability of the incremental timestamp-based data capture method, and also takes into account the efficiency of data extraction under the premise of ensuring reliability.

The data maintenance variable time in VTWM proposed in this paper needs to be manually set currently, and there is no uniform standard for setting the value size. The size of the value has a certain impact on the efficiency of data extraction. This study hopes to make a personalized setting based on the frequency of change of the source table data in the future studies, and minimize the effect of this value on data extraction efficiency.

References

[1] Mis Minakshi.HC Sharma. Near Real-Time Data Warehousing Using State-of-the-Art ETL Tools [J]. International Journal of Research (IJR) Vol-1, Issue-10 November 2014.

[2] SHU Qi. The Research on Optimization of ETL Process and Incremental Data Extraction [D]. Hunan University,2011.

[3] Jorg T, Desloch S. Towards generating ETL processes for incremental loading[C]. International Database Engineering and Applications Symposium, 2008: 101-110.

[4] Mekterovic I, Brkic L. Delta view generation for incremental loading of large dimensions in a data warehouse[C]. international convention on information and communication technology electronics and microelectronics, 2015: 1417-1422.

[5] Jia Yankai. Research and Design on Data Extraction in Multiple Data Sources [D]. Harbin Engineering University,2013.

[6] Wen Lu. Design and Implementation of Incremental Data Extractor based on Sector Inquiry[D]. Hebei University of Science and Technology,2015.

[7] XU Fuliang , ZHOU Zude.Research on Change-Data-Capture Technology[J]. Journal of WUT(Information & Management Engineering),2009, 05:740-743.

[8] DAI Hao,YANG Bo.Researches on mechanics of incremental data extraction in ETL[J].Computer Engineering and Design, 2009,(23):5552-5555.

[9] WANG Yu-biao, RAO Xi-ru, HE Pan.Incremental database synchronization update mechanism under heterogeneous environment [J].Computer Engineering and Design,2011,03: 948-951.

[10] YANG Le. Design and Implementation of Real-time data extraction Mechanism in data warehousing [D]. Beijing University of Posts and Telecommuni- cations,2007.

[11] [TAN Guang-wei,WU Tong. Study on Method of Data Warehouse Real-time Data Updating Based on Mechanism of CDC [J].Computer Science, 2015, 42(s1).

[12] ZOU Xian-xia , JIA Wei-jia,PAN Jiu-hui. Research of Log-based Change Data Capture [J]. Journal of Chinese Computer Systems, 2012, 33(3):531-536.

[13] MattCasters, RolandBouman, JosvanDongen,etl. Pentaho Kettle Solutions: Building Open Source ETL Solutions with Pentaho Data Integration [M]. Publishing House of Electronics Industry, 2014.

[14] CUI You-wen,ZHOU jin-hai. Research on Data Integration Based on KETTLE [J]. Computer Technology and Development,2015,(04):153-157.

[15] LIU Xueqiong,WU Gang,DENG Houping.Data deduplication in Web information integration[J]. Journal of Computer Applications,2013, (09):2493-2496.

[16] Ari Wibisono,Wisnu Jatmiko,Hanief Arief Wisesa,Benny Hardjono,Petrus Mursanto. Traffic big data prediction and visualization using Fast Incremental Model Trees-Drift Detection (FIMT-DD)[J]. Knowledge-Based Systems,2016,93.

[17] Annie Anak Joseph,Takaomi Tokumoto,Seiichi Ozawa. Online feature extraction based on accelerated kernel principal component analysis for data stream[J]. Evolving Systems,2016,7(1).

[18] Haixiang Li,Zhanhao Zhao,Yijian Cheng,Wei Lu,Xiaoyong Du,Anqun Pan. Efficient time-interval data extraction in MVCC-based RDBMS[J]. World Wide Web,2019,22(6).
[19] Chao Tan,Genlin Ji. Semi-supervised incremental feature extraction algorithm for large-scale data stream[J]. Concurrency and Computation: Practice and Experience,2017,29(6).