2023 3rd International Conference on Electronic Information Engineering and Computer Communication (EIECC) | 979-8-3503-5996-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/EIECC60864.2023.10456691

Research and Implementation of Complex Real-time Computing based on Data Middle Platform

Peng Wang
NARI Group Corporation (State Grid Electric Power·Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
wangpeng@sgepri.sgcc.com.cn

Jiajin Li*
NARI Group Corporation (State Grid Electric Power·Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
*Corresponding author: lijiajin@sgepri.sgcc.com.cn

Baolong Xuan
NARI Group Corporation (State Grid Electric Power Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
xuanbaolong@sgepri.sgcc.com.cn

Wei Lu
NARI Group Corporation (State Grid Electric Power·Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
lu-wei@sgepri.sgcc.com.cn

Xian Zhang
NARI Group Corporation (State Grid Electric Power Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
zhangxian@sgepri.sgcc.com.cn

Jian Hu
NARI Group Corporation (State Grid Electric Power Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
hujian@sgepri.sgcc.com.cn

Hailang Yan
NARI Group Corporation (State Grid Electric Power Research lnstitute); China Realtime Database Co., Ltd.
Nanjing, China
yanhailang@sgepri.sgcc.com.cn

***Abstract*—With the development of big data, the need for real-time data processing becomes more urgent. However, the complexity of the big data business causes the current data components to not be well supported. For this reason, a complex real-time calculation method based on data center is proposed. Build incremental dimension tables and incremental fact tables based on the original micro-batch scheduling, and analyze the impact of business incremental data changes on statistical results. And through the analysis of business operations, simplify the data processing process, reduce the data Merge operation, and reduce the data processing time. Finally, the feasibility, correctness and real-time nature of the method are verified through the application of real-time calculation of electricity bills.**

Keywords-data middle platform; real-time computing; stream batch integration

I. INTRODUCTION

The data middle platform is a unified data processing platform for constructing multiple systems, forming a consistent data structure, unified data modeling, and unified external data services. While providing consistent data externally, it avoids creating data chimneys and information islands [1][2].

In the data middle platform, offline batch processing technology can already meet a wide range of requirements. However, an increasing number of application scenarios are putting forward higher demands on the timeliness of data [3] [4]. Traditional technology has been unable to meet the data timeliness requirements of business and management decisions [5][6]. The value of data lies in its timeliness, if data is not processed and utilized promptly when generated, it cannot maintain the highest "freshness" and maximize its value.

Stream data in general has the following characteristics:

1. Stream data has high timeliness, it requires data responses in seconds or even milliseconds;

2. Stream data processing tasks are typically continuous tasks that run continuously once started;

3. Stream data requires that the system has high performance and has the ability to balance high throughput and low latency;

4. Stream data computation is costly and may not support complex business logic scenarios. The arrival time of data is uncertain and results may differ from offline computations. It is difficult to achieve for high accuracy data statistics.

Currently, real-time computing is primarily achieved through two methods:

1. The script of batch processing system is scheduled many times by merging the incremental table to the full table in real time, refreshing the data. The advantage of this method is that the data statistics are accurate. The code logic does not need to be modified, running the script repeatedly many times can achieve real-time calculation. However , the downside of this approach is that it takes a lot of time, offline systems are modeled hierarchically and the data generation time is further delayed;

2. Real-time statistics are accomplished through tools like Flink or Blink, with results being output in real-time. The advantage lies in fast computation and high timeliness. However, the disadvantages include complex logic in cases of multiple stream associations, substantial waiting for intermediate data, high computational costs, and difficulty in supporting demands for data retrospective recalculations [7-10].

Therefore, the current challenges that need to be addressed are as follows:

1. How to achieve real-time statistics for daily data under the middle platform's offline batch processing architecture while ensuring the accuracy and timeliness of data statistics.

2. How to reuse existing offline models of the middle platform to build real-time statistical models, enabling rapid iteration for the implementation of real-time metric statistics.

To address these challenges, The innovation of this article is to propose a quasi real-time method suitable for large-scale complex data statistics under the existing offline batch processing architecture of the data center, which solves the problems of slow response speed, complex calculation, and high calculation cost in complex real-time calculation of big data under traditional batch processing architecture, and ensures the accuracy of data statistics.

II. Overview of Related Technologies

The previous chapter provided a brief introduction to real-time computing related to big data, and this chapter explores the technical models related to real-time computing in this paper.

A. Data Middle Platform Layered Architecture

In order to improve data query performance, reduce data redundancy, achieve reuse of results and address inconsistencies in data statistics criteria, the data middle platform adopts a layered modeling design [11][12]. The layered structure consists of the ODS (Operational Data Store), DWD (Data Warehouse Detail), DWS (Data Warehouse Service), and ADS (Application Data Service) layers, as illustrated in Figure 1.

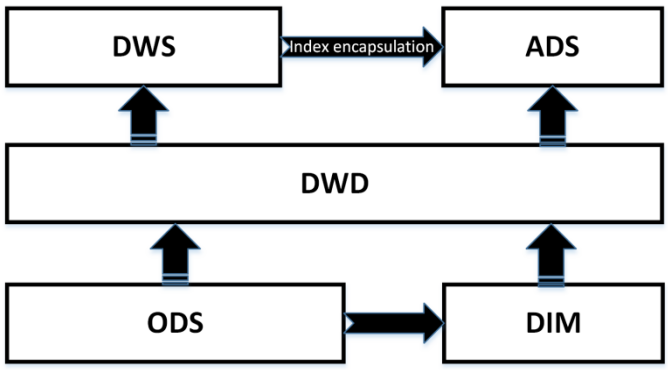

Figure 1. Middle platform layer architecture

ODS (Operation Data Store): The ODS layer Includes full tables and incremental tables. The data stored in the full table is consistent with the data from the previous day in the original business system. The incremental table records the running data of the operation log of the corresponding data table, such as OGG logs in Oracle database, BLINLOG logs in MySQL database, etc. It uses fields like U (Update), I (Insert), D (Delete) to identify the update status. Merge operations are performed on multiple records at scheduled intervals every day, combining the latest data with the maximum update time.

DWD (Data Warehouse Detail): Following dimensional modeling theory, the DWD layer constructs fact tables and dimension tables to form a multi-dimensional star model.

DWS (Data Warehouse Service): Based on the aggregation of the detail layer, the DWS layer performs dimensional degradation to form a multi-dimensional and multi-metric metric datasets.

ADS (Application Data Service): The ADS layer combines the results of the corresponding metrics based on the aggregation layer and encapsulates the data services externally.

B. Dimensional Modeling

Dimensional modeling divides tables into fact tables and dimension tables based on the metrics in the business process and the context describing the business process. There are various types of fact tables and dimension tables, such as single transaction fact tables and multi-transaction fact tables. In this paper, facts refer to the measurements of data, such as unit price and quantity, while dimensions represent perspectives for statistical analysis, such as filtering conditions and the scope of statistics. The discussion in this paper primarily focuses on the real-time statistics of additive metrics. For semi-additive metrics, they can be decomposed into additive metrics.

III. Specific Content of Real-time Solution

The previous chapter mainly introduced the layered architecture of the data middle platform and the dimensional modeling. This chapter focuses on detailing the specific principles of the real-time solution based on the aforementioned models.

A. Middle-tier Architecture

During the initialization of the data middle platform, business database data is fully synchronized. The ODS layer's full table stays consistent with the business data. The ODS layer's incremental table synchronizes business logs, recording data changes. At midnight every day, the incremental data is grouped by the primary key, and the corresponding full records are updated based on the operations (U, I, D). Subsequently, dimensional modeling is performed based on the full data, constructing corresponding fact and dimension tables, forming a wide table in the subject area. The degraded dimensions are used to build offline T-1 metrics, supporting report, dashboard, and data service products. Merging increments into a large table for association scanning with small tables takes a relatively long time, especially in multi-table associations, where one needs to wait for all tables to complete incremental and full merges, making the process too time-consuming to meet real-time statistical requirements.

Based on the above architecture, this real-time solution is built upon the full table and incremental table in the source layer of the offline system. By analyzing the impact of the day's change values in the incremental data on both the full table and the incremental table itself, we derive the change in overall

metrics for the day, constructing the Incremental DWD layer. The table structure and statistical logic of the Incremental DWS and Incremental ADS layers are consistent with the full DWS and ADS layers, reducing development effort. The incremental real-time metrics are added to the full offline metrics in data services, achieving real-time statistical metrics. The overall architecture is illustrated in Figure 2.

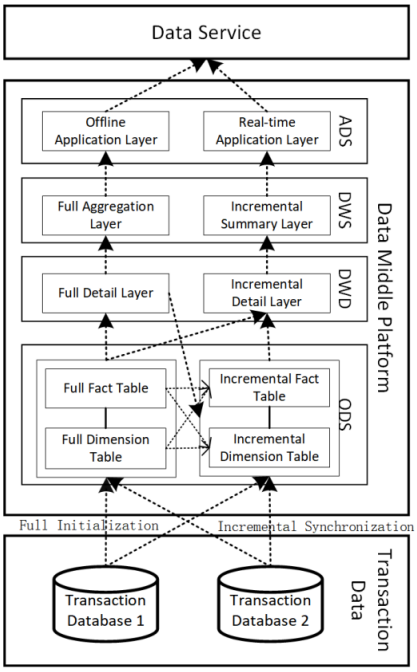

Figure 2. Real-time data model

B. *Overall Real-time Solution*

The previous section discussed the middle platform architecture and real-time program ideas based on this architecture. This section primarily delves into the specific real-time solution. By analyzing the changes in dimensions and facts in the incremental data for the current day, the solution assess the impact of these changes in the incremental table on the dimensions and facts in the full table, calculating the final impact value for real-time metric calculations. The overall approach is illustrated by equation (1):

$$Y_t = Y_{t-1} + \alpha Y_{t-1} + X_t \quad (1)$$

Explanation for equation (1) is provided as follows:

Y_t: Real-time data result values for the day.

Y_{t-1} : Results of the previous day's statistical metrics. This result is obtained by the offline batch processi ng system.

αY_{t-1}: Amount of impact on the previous day's statistics caused by some of the day's dimension table data. α refers to the influence of data in the incremental table for the current day on the data from the previous day such as updates or deletions and changes in the statistical logic that impact metric result in other tables.

X_t: Incremental data for the day. This data specifically refers to pure incremental data for the current day, and its processing logic is entirely consistent with the full data processing logic of the batch processing system.

From the above formulas, it can be seen that αX_{t-1} and X_t are the focal points of research in real-time computing processing. The main processing methods are shown in TABLE I.

TABLE I. REAL-TIME COMPUTING METHOD PROCESS

Input: Incremental data from the source layer (quasi-real-time data stream); Full data from the source layer;
Output: Real-time calculation metric values

Calculate Y_{t-1};
Identify the core primary fact table Fmain;
Determine the associated fact table Fother;
Identify the dimension table Dmain;
While(True):
1. Take the latest data from each incremental table;
2. Determine the change status of the latest data;
3. Calculate αY_{t-1}:
If(today's incremental data has an impact on yesterday's metric statistics):
If(flag = 'U' && change content to add or subtract metrics): At this point, two approaches can be adopted.
Approach One:
3.1 The incremental DWD layer is constructed by the primary key of the core primary fact table. The data in this record are the difference between the changed data and the original metrics, while the dimensions remain unchanged. Metrics that have not changed are set to 0;
Approach Two:
3.1. Add a new record with all metrics as negative values based on the primary key of the core primary fact table, while keeping the dimensions unchanged;
3.2. Associate today's incremental data with the full data from day T-1, constructing a new wide table of statistical results.
Else If(flag = 'U' && change content is a dimension change):
Refer to Approach Two;
If(flag = 'D'):
Refer to Approach Two 3.1;
4. Calculate X_t:
If(Today's incremental data has not been included in yesterday's statistics):
4.1. Interconnect today's incremental data with the full data using the logic of statistical calculations to build a new wide table of statistical results;
5. Update the incremental data in the DWS and ADS layers;
6. Accumulate incremental data and full data through data services to provide real-time metrics;

The above process is explained as follows:

1. Due to the consideration of various redundant scenarios during data warehouse modeling, it is necessary to filter fields for each table according to the statistical logic, reduce judgment logic, and differentiate each table based on dimensions and facts, forming TABLE II as follows.

TABLE II. DIMENSION AND FACT TABLE STRUCTURE

Column	Comment
Id	Primary key
Fact2	Fact 2
Dim1	Dimension 1
Dim2	Dimension 2
……	……

2. Determine the core tables. The middle platform model is mostly constructed through a star schema, where a fact table is associated with multiple dimension tables. The fact table is the core table, it's primary key serves as the foreign key for other tables. When other tables are associated with this core table, data duplication issues do not arise. This ensures that the data volume of the generated model tables after association is consistent with the original core table data volume.

3. Group the data by the primary key and obtain the latest data by sorting based on the update date. Following the logic in TABLE I, construct and rewrite the data in the Incremental DWD layer.

4. Aggregate data in the DWS and ADS layers, and accumulate incremental ADS data and full ADS data in data services to construct real-time data.

5. Loop starts from step 3.

C. Further Optimization

The data merge operations mentioned earlier consume a significant amount of time. Therefore, reducing these data merge operations is also a crucial path to enhance this real-time solution. Through modeling and analysis of business processes, the following optimizations can be made to the above solution, as illustrated in Figure 3.

For metric-type data, to ensure accuracy without allowing business modifications, statistical calculations can directly group data by the primary key based on that numerical value, resulting in a unique record.

For certain dimension-type data, partial optimizations can still be applied. The changes in conditional log data depend on the process variations in various business processes. For irrevocable processes and the conditional data involved in the final process, there is inevitably only one record. Therefore, incremental records identifying this process can be directly selected based on the primary key.

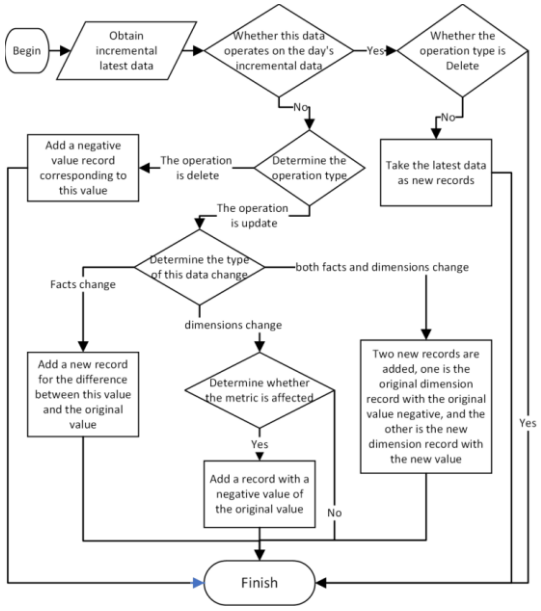

Figure 3. Flowchart for optimizing real-time computing methods

IV. Related Applications in Real-Time Reporting

The previous sections mainly discussed the specific methods of the real-time solution. This chapter focuses on describing the modeling and application of real-time electricity charge calculations in JiangSu Energy Internet Marketing Service System Unified Software 2.0, including data report and other related products.

A. Model Construction

Electricity charge calculation involves 12 tables, including installation point billing card, segmented quantity and charges, additional charges, etc. The maximum data volume for incremental changes is at the level of one billion. Report statistics involve various dimensions such as voltage level, electricity usage category, tariff codes. Multiple metrics involve sharp-peak-flat-valley and time-of-use electricity quantity and charges, profit and loss, basic electricity charges, additional charges, etc. As shown in Figure 4, modeling is performed based on business processes to construct metrics related to electricity charge issuance.

The electricity charge issuance follows a strict business process, including plan creation, electricity quantity calculation, electricity charge calculation, and issuance, as shown in Figure 5. After the electricity charge issuance, relevant data cannot be changed arbitrarily, and numerical values cannot be modified during updates. In the processing of incremental tables, the following method is employed: For numerical tables such as billing cards for installation points and segmented quantity charges, values to be counted on the day are obtained by grouping according to the statistics to reduce merge operations.

Whether the electricity bill is issued is estimated by the issue date. After the electricity charge is issued, the issuance cannot be rolled back. Therefore, to determine whether the electricity charge has been issued on a given day needs to look for records directly in the incremental table where the issuance date field is not empty.

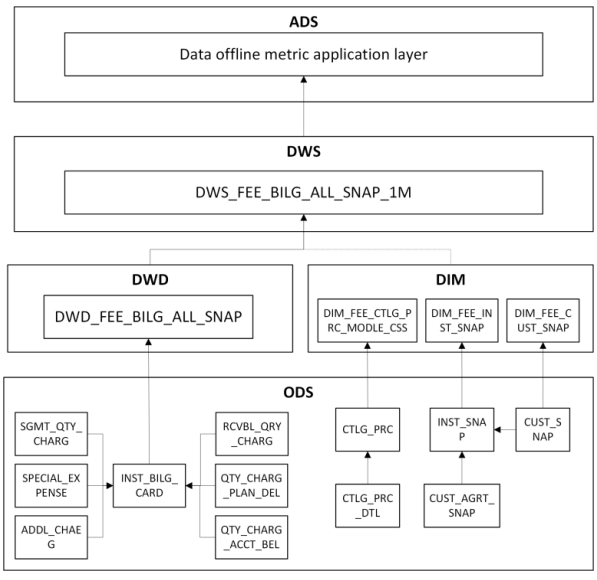

Figure 4. Electricity charge related table dimension modeling

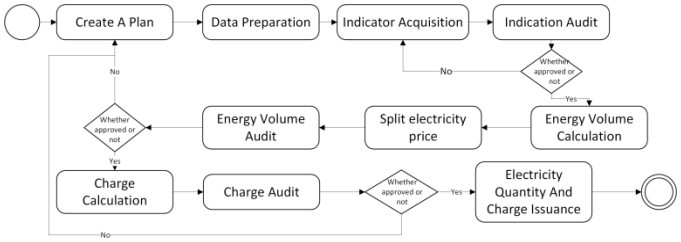

Figure 5. Electricity charge issuance process

B. *Result Analysis*

Through multiple scheduling in the production environment, the experimental data obtained is as follows. Due to data resource limitations, there is a significant difference in the time required for multiple runs. The overall situation is as follows.

As shown in Figure 6, quasi real-time calculations were carried out through a batch processing system, with a maximum runtime of 113 minutes, a minimum of 97 minutes, and an average of 106 minutes. As shown in Figure 7, the maximum calculation time for this scheme is 22 minutes, the minimum is 7 minutes, and the average is 16 minutes. The real-time computing efficiency of this scheme is much higher than that of batch processing systems.

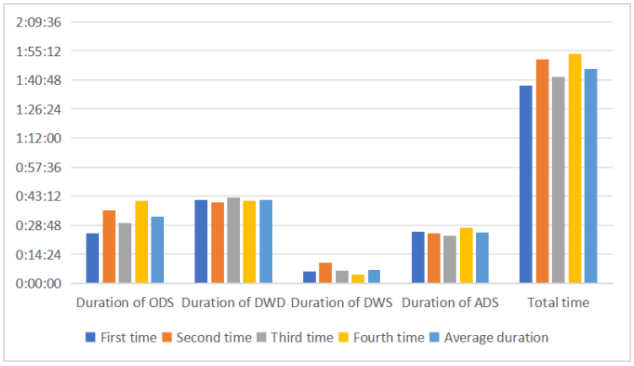

Figure 6. Batch processing system runtime

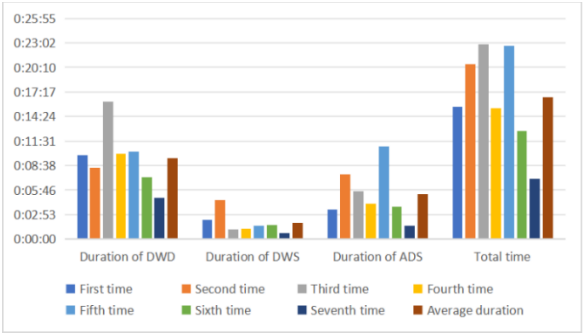

Figure 7. The system running time of this plan

V. Analysis and Prospects

Regarding the proposed complex real-time computing method, it has been validated through the electricity charge calculations in JiangSu Energy Internet Marketing Service System Unified Software 2.0. It meets the requirements of the correctness and accuracy of report statistics and basically meets the real-time requirements of complex data statistics.

Currently, the complex real-time computing method proposed in this paper based on the data middle platform has been applied effectively. To achieve rapid data output, calculation units are merged at the hierarchical level, reducing the topological hierarchy, decreasing the levels of node scheduling, and minimizing data storage and retrieval, thereby shortening the metric generation time. However, the current system still takes too long to generate metrics and does not achieve the capability for sub-second responses. Moreover, resource contention can lead to node blocking, causing unpredictable delays in scheduling. Therefore, the next focus of research could be on leveraging real-time computing components to apply the current incremental table processing methods to incremental data, addressing issues related to retrospective calculations in real-time components and further enhancing the real-time nature of relevant metric calculations. On the other hand, restructuring the existing architecture by splitting and scheduling the logic processing of lower-level tables across nodes can reduce the scheduling time for nodes with longer processing times.

VI. Conclusion

This paper analyzes traditional batch processing systems with multiple script schedules and current mainstream real-time computing component frameworks to achieve complex real-time computations. It explores the method of constructing quasi-real-time complex calculations based on incremental log tables. Subsequently, by analyzing business change logs, the processing method for incremental tables has been optimized, reducing the Merge operations on incremental tables. Through this method, the modeling and implementation of calculations for relevant reports, dashboards, and other metrics in the data middle platform have been achieved.

Acknowledgment

This work was supported by Research and Application of Key Technology of Management Digitization based on Data Middle Platform (No.524623210007).

References

[1] Zhi Li, Xiaolu Fei, Zhen Guo. "Research on Data Asset Management Method of Power Enterprise Based on Data Middle Platform." Electric Power Information and Communication Technology, 2020,18(07):76-81.DOI:10.16543/j.2095-641x.electric.power.ict.2020.07.013.

[2] Qiming Wan, Sining Wang, Xin He. "SG-CIM model application method in data middle platform." Telecommunications Science. 2020,36(03):136-143.

[3] Xinpeng Li, Wei Liu, Zhiping Yang, Pengfei Sheng. "Research on Data Mid-Platform Scheme of Power Grid Enterprises." Electric Power Information and Communication Technology, 2020,18(02):1-8.DOI:10.16543/j.2095-641x.electric.power.ict.2020.02.001.

[4] Bingsen Li, Quangui Hu, Xiaofeng Chen, Bingqiang Gao. "Research and Design of Data Platform for Power Grid Enterprise." Electric Power Information and Communication Technology. 2019,17(07):29-34.DOI:10.16543/j.2095-641x.electric.power.ict.2019.07.006.

[5] Pratt, Dexter , et al. "NDEx, the Network Data Exchange." Cell Systems 1.4(2015):302-305.

[6] Mishina, S. V. and Dmitriy Kornienko. "Setting up data exchange between information systems that automate accounting at the enterprise." Journal of Physics: Conference Series 2094 (2021): n. pag.

[7] Bing Yan, ZhongLei Wang. "Real-Time Calculation Method of Big Data Index Based on Storm." Computer Systems & Applications, 2019,28(04):90-95.DOI:10.15888/j.cnki.csa.006840.

[8] Wenjie Zhang, Liehui Jiang. "Parallel computation algorithm for big data clustering based on Map Reduce." Application Research of Computers,2020,37(01):53-56.DOI:10.19734/j.issn.1001-3695.2018.05.0496.

[9] Siping Xu. Research on the Key Technology of Real-time Computing of Big Data Supporting the Back Calculation. Diss. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences.

[10] Sheng Zhao, Jinlei Jiang. Typical Big Data Computing Frameworks[J]. ZTE TECHNOLOGY JOURNAL, 2016,22(02):14-18.

[11] Sureddy, Madhusudhan Reddy , and P. Yallamula . "DATA QUALITY ARCHITECTURE FOR DATA WAREHOUSES." (2020).

[12] Ali, Taghrid Z. , et al. "A Framework for Improving Data Quality in Data Warehouse: A Case Study." 2020 21st International Arab Conference on Information Technology (ACIT) 2020.