ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Indian Journal of Science and Technology, *Vol 7(21), 10.17485/ijst/2016/v9i21/95295, June 2016*

Time Sensitive Business Intelligence - Big Data Processing Methodology for Frequently Changing Business Dimensions

Anusuya Kirubakaran[1]* and M. Aramudhan[2]

[1]Mother Teresa Women's University, Kodaikanal - 624101, Tamil Nadu, India; Anusuya.Kirubakaran@outlook.com
[2]PKIET, Karaikal - 609603, Puducherry, India; Aranagai@yahoo.co.in

Abstract

Background: In the competitive data driven business world, business Intelligence (BI) team converts the raw operational data to information for decision making. Operational system captures the day-to-day operations and BI database refreshes operational data periodically. **Methods:** A component to create the metadata repository which maintains the current BI database summary by logical data partitioning using range based partition for frequently changing parameter which are critical to business. During different time frequency, using metadata repository component identifies the latest data victim between BI vs operational data and refreshes the modified victim to BI database. **Findings:** In traditional data loading approach from Operational system to BI database, huge volume of data gets refreshed periodically irrespective of modifications, which leads to higher processing time and cost. To overcome this limitation, this proposed methodology helps to identify the latest data victims present in operational systems instead of bulk data replacement which can minimize the processing time and enables faster data transformation to achieve "Time to Decision" and "Quick to Market" implementation for business enhancements. Also component can be scheduled for data refresh with different time frequency for multiple critical to business as well as frequently changing parameters. **Applications**: In financial, traffic, weather, e-business, Logistics & stock management transactions, data changes frequently and process big data periodically to gain real-time knowledge discovery for time sensitive decision making.

Keywords: Frequently Changing Data,Time-sensitive Business Intelligence

1. Introduction

Modern days, Organizations rely on Decision Support System (DSS) for business strategies planning and execution as well as to survive &nurture the business opportunities focus on discovering knowledge, enabling quality and near real-time decisions. Unless the right data is available on time, it would impact the organization performance and good will up to a huge extent. For example, Real Time Gross Settlement fund transfer meant for large-value payments gets processed at the time of instructions received. To settle the funds transfer requests received from the customer, the financial services processes the requests with Centralized Payment System developed by National Monetary Control Authority (NMCA) to transfer the payment. A financial service gets membership from NMCA for on-line real time settlement of payments as through gross basis or Multilateral Settlement Batches. If the number of fund transfer request increases more than the approved limit by NMCA, to facilitate settlement of transactions, the financial services proactively predict the situation and get prior approval from NMCA to extend the real-time settlement of payments. This kind of time sensitive operational business intelligence processing impacts the business goals and force the organization

*Author for correspondence

to do complex groundwork involving high cost for infrastructure and man power.

Data is the real world object[1]which processed for knowledge discovery. As per the data representation[1] it's classified into

- Structured : Data represented in fixed structured
- Semi-structured: Data fields are not fixed at design time.
- Unstructured : Natural language

As per the data modification frequency[1], data will be classified into

- Stable
- Long-term changing
- Frequently-changing data

For Frequently-changing structured data[1], organization's business intelligence team generates the report on timely or requirement basis to operational team for decision making (E.g.: Supply chain/Inventory management team periodically reviews the inventory level for e-business). Business Intelligence team use the traditional analytics tools to process the raw data to meaningful information. The traditional analytical software loads the bulk data into user's local temporary space for processing. In recent years, to improve the overall processing speed of the business, in-memory business analytics methodology has been widely preferred by the architects in which data resides in user's Random Access Memory (RAM). To overcome the limitations with traditional data loading approaches like bulk appending and manual clean-up, we have proposed a big data methodology in this paper. We define, this methodology is to integrate the recent operational data with initial and subsequent data loaded in BI database by periodical data update using frequently changing business dimensions, instead of traditional reloading the bulk data every time and acquire business knowledge for time sensitive decision-making. Specifically, this methodology identifies the modified victims on the bulk and refreshes the victims for processing. To achieve it, we create the metadata repository which maintains the current data summary details for critical &frequently changing business parameters versus partitions where the partitions are smaller subgroup of the initial bulk which is logically separated. Using this frequently changing parameter vs partition summary details available on metadata repository, we refresh the latest data by comparing the current operational dataset in different partitions and this comparison of data would be managed by the component proposed. This component would be developed and implemented on BI tool which would be scheduled on different time frequency i.e. based on the frequency of data changing in different parameters we can setup the different timings for data refreshment. This approach avoids the manual intervention, enables the latest data for processing which optimize the time and processing cost and this methodology can be applied for frequently changing business dimension domains namely financial, traffic, weather, e-business, Logistics and stock management applications.

2. Background

In financial, healthcare &e-business sectors, business Intelligence data processing on operational systems (like OLTP) has been discouraged by the organizations, since its challenges the performance and data security. By nature, data warehouse possess non real-time data with the considerable time delay leads to missing execution during time sensitive situations related to the compliance and operational failures. Since operational systems & data warehouse does not support real time analytical information needs, organizations has to setup the information platform in Operational Data store as interim solution for real time or near real-time data collected from multiple legacy systems. The real-time analytics on information platform's process the large and vast growing body of raw data and provides the intellectual, accurate and complete information on timely manner.

2.1 Big Data

Data is emerging and large volumes of data are collected by the each and every organizations and business since data becomes the new resource for competitive advantage, real-time operational intelligence[2] to enable better business decision-making. Big data challenges the processing speed since the data size, variety and frequency of data generation growing exponentially. Big Data analysis now drives nearly every aspect of our modern society, including mobile services, stock trading, retail, manufacturing, financial services, life sciences, and physical sciences.

2.2 Operational Data Store

An Operational Data store[3–6] contains current operational data pulled periodically from On-line Transaction Processing (OLTP) for real-time or near real-time reporting/analysis. ODS is built to have current operational data with few days of historical data. OLTP will have only the current transactional data and not preferred for reporting /analysis and Business not permitting to access the data directly from OLTP which may leads to data loss since the backup data loaded daily, hourly, or even immediately after transactions on operational data based on the business needs. It can optionally serve as a data source for the data warehouse.

2.3 In-Memory Database

An In-Memory database[7,8] is a database management system which stores the data in computer Random Access memory (RAM). Since the database stored in main-memory which minimize the query response time and makes faster reading, writing and processing. In-memory data storage designed to persistent media which differs from traditional database on-disk storage systems. In-memory database plays a vital role in analytics since the novel operational data can be pulled from operation database for business intelligence decision-making. In-memory databases are referred as Main Memory Database systems (MMDB), and have become more popular in recent years for handling High-Performance Computing (HPC).

2.4 Metadata Repository

A metadata repository[9–11] is a database which contains the metadata and structure about actual data also known as data dictionary. Metadata can have data access details, description about the specific data. Mainly Metadata has two types of details called business details and technical details. Business metadata[12,13] (data warehouse metadata, front room metadata, operational metadata) has high-level definition of the database schema. Technical metadata[12,13] (back room metadata, transformation metadata) has information about data flow from source system to destination system. Metadata Repository used for integrating and automatic synchronization with other metadata source systems across the organization.

2.5 Range based Partition

Range based partition[14–16] divides the bulk to smaller independent parts based on data range (e.g. employee number between 500 to 700). This enables the data access, maintenance and reachability easier. Each partition is provided with the key which can be referred by other tables for references.

2.6 Frequently vs Slowly Changing Parameter

Based on the time dimensions, the operational data base columns are divided to frequently changing and slowly changing[17,18] parameters. In Slow changing parameters, data gets modified once in a while but the frequently changing parameter gets updated with short term period. As in Figure1, account balance parameter gets modified frequently.

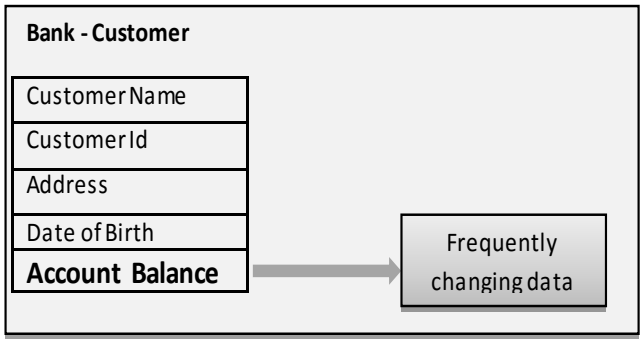

Figure 1. Frequently changing data.

3. Methodology

3.1 Problem Statement

This problem aims to optimize the cost & time since the traditional data processing technique loads the bulk data from transactional database for real time analytics even though the amount of data variation from the initial bulk data available in the BI database is minimal and this redundant data pulling involves high cost in terms of time, effort and rework. To overcome this limitation, the proposed methodology aims to avail the latest data for processing in BI database by periodically automated update and ignore the download of huge volume of

unchanged data by identifying the modification victims and refresh the victims to BI database. Assume that operational dataset (L)has been loaded to the destination system earlier and needs to be reloaded with latest modified operations data from the source systems (I). Need to identify the modified dataset from the last pull and load the victims into L which expressed as $L = \Lambda Y (\sim (I \cap \Lambda) \subseteq I$

3.2 Proposed Architecture

In traditional big data processing, BI team pull the operational data from the online transactional database / operational data store based on the enterprise environmental setup using business intelligence tools. The business analysts process the data and transform the data to information required for decision making. The real-time analytics are majorly focused by the enterprise for time sensitive scenarios. In this time sensitive situation, the reliability, consistency, accuracy and correctness of the reporting should be higher. Due to some limitations in the traditional business intelligence tool, the reporting accuracy, correctness is impacted. The identified limitations are:

- As stated before, though the variation of initial data fetched for data processing and subsequent data are minimal, most of the BI tools download the huge volume of data from operational database to temporary space on different frequencies by complete replacement.
- In some instances, cached data would not get replaced with the updated data due to huge data handling in temporary memory, which affects the reporting quality and reliability.
- Sometimes subsequent data appended with initial data makes the database size growing cumulatively, which requires more database size and manual intervention for cleanup activities.

To overcome these limitations, we added the metadata repository which maintains the history of the data gets

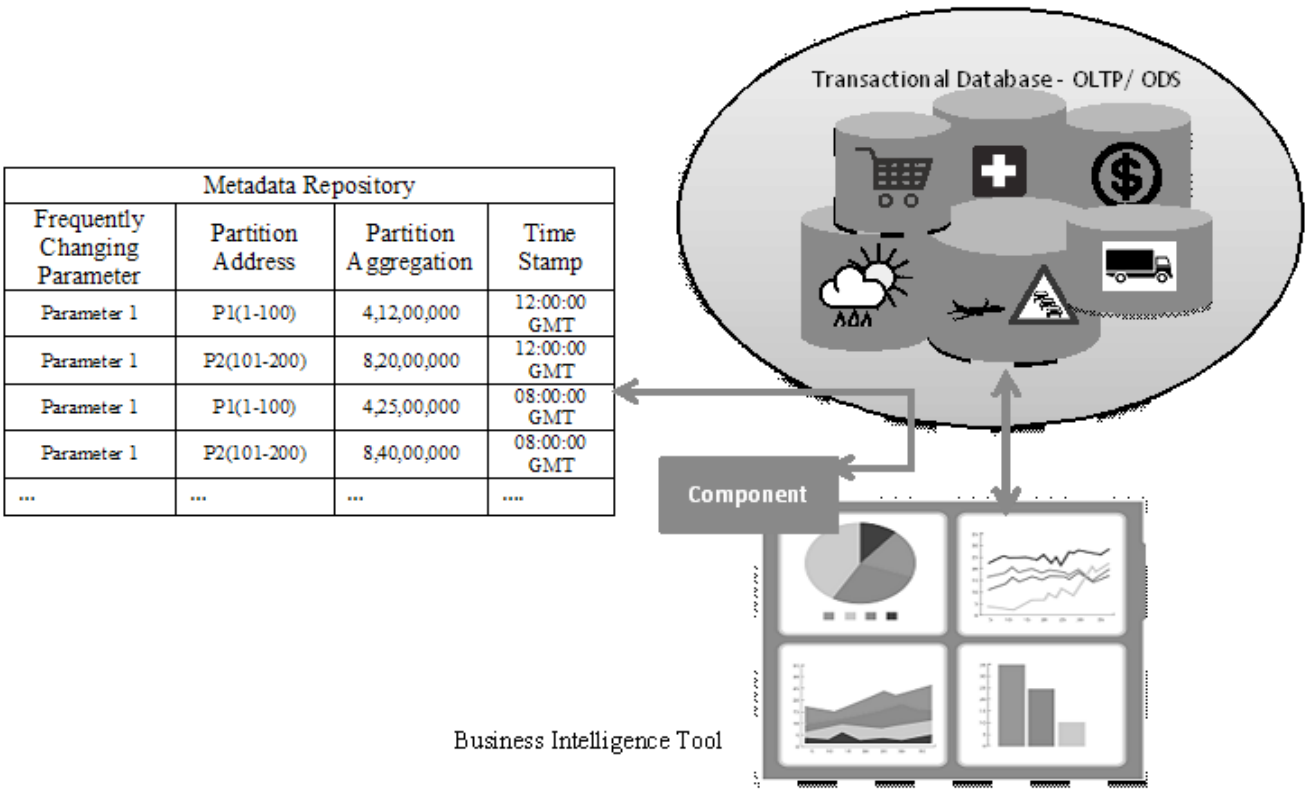

Figure 2. Proposed component architecture.

loaded to destination database for all frequently changing parameter with time stamp and the proposed component added on the business intelligence tool to identify and load the modified victim from source operational database on timely basis for processing as in Figure 2.

3.3 Proposed Methodology

Let I= {1,2,3,4…….n} be the current dataset in source database, L = {1,2,3,4…….n} be the current dataset in destination database where $L \subseteq I$. This proposed methodology aims to find the latest modified data (victims) in source database by comparing the metadata repositories history recorded for the frequently changing parameters for the last data pull versus current source database and ignore the blind replace of source database to destination database. The proposed methodology has work flow as below:

- Identify the frequently changing parameters which are critical to business.
- Create the Metadata repository which has frequently changing parameter, logical data partition details, partition aggregation details and last data pull time stamp for the Initial data loaded to the destination system by the business intelligence tool for business processing.
- Create a component, which compare the source data with metadata repository partition's aggregation and identify the victim and refresh the partition with different time frequency.

3.2.1 Identify the Critical to Business - frequently changing Parameters

According to data modification history in the source database, the columns which are critical to business and the frequency of data changes are identified as frequently changing parameters (r). Let R= $\{r_1 , r_2 \ldots\ldots r_x\}$ where $r \in R^x$ be the list of frequently changing parameters used to identify the data victims between source I and target database L. Identify and add the frequently changing parameter to the items set $r \in R^x$.

3.2.2 Metadata Repository

Metadata repository would be created with attributes namely frequently changing parameters($r \in R^x$), logical data partition key address ($p_t \in L'$), partition aggregation (A), and last data pull time stamp (T). Time Stamp T is used to maintain the history of data refresh, since the data pulled from source database periodically. $p_t \in L'$ is partition address used to store the logical partition key address i.e. $L' = \{p_1 , p_2 , p_3, p_4 \ldots\ldots p_k\}$,and Ais the frequently changing parameter's partition aggregation. This paper primarily focuses on reducing the time latency to load the bulk data with minor modification from source database to destination database. To achieve minimum time latency, we logically divide the bulk to smaller partitions using range based partition algorithm. Range based partition divides the data horizontally with specified range and group the records to capture the aggregation details of the bulk.

In this step, the destination dataset L = {1, 2, 3, 4…….n} logically partitioned to $L' = \{p_1,p_2, p_3, p_4 \ldots\ldots p_k\}$ where k is the number of partitions as in Figure 3. The number of partitions k is derived from

K= L /number_of_items_in_single_partition

Where

L is the destination dataset

number_of_items_in_single_partition is derived based on the maximum frequently changing parameter's aggregation value can be stored in metadata repository's partition aggregation attribute.

After logical data partition, metadata repository records the data aggregation for all the frequently chang-

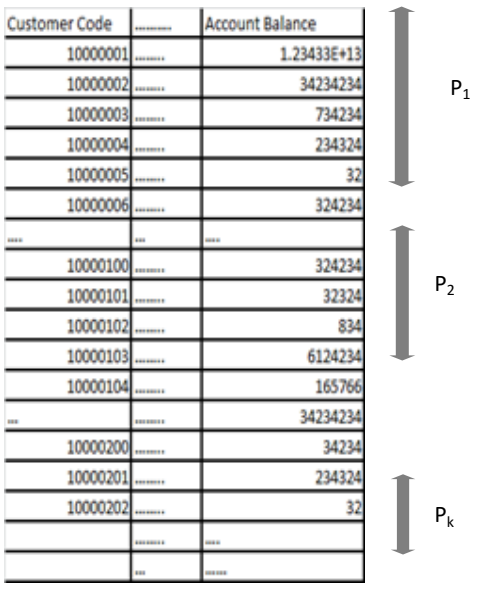

Figure 3. Bulk partition.

ing parameter with partition address and last data pull time stamp. The latest L' item set's aggregation compared with source dataset I's current aggregation for frequently changing parameters r∈ L and victims gets identified.

3.3.3 Component Illustration

Once metadata repository is created, we create a component on destination (business intelligence tool) which finds the logical partition range from the metadata repository for every frequently changing parameter and map the same range with the destination database then compare the partition aggregation. This aggregation comparison helps to find the victim based on data aggregation discrepancies.

To refresh the single partition given below,

$L' = pi \sum_{i=1}^{i=k}, L'$ L we can refresh pi into L'

while

p_i= reload if $p_i \neq I_i$ { The identified victim partition gets reloaded to destination database

none if $p_i = I_i$ No data variation is found

where

p_i is the frequently changing parameters aggregation in metadata repository

Ii is the frequently changing parameters aggregation in source system

k is the number of logical partition in metadata repository

Based on the above, the modified victims are identified by the component for data refresh. Once the data refreshment happened, the component records the recent data details history in metadata repository for future reference with time stamp. Component would be scheduled with different time frequencies for each frequently changing parameter based on the business needs and component configured to refer latest timestamp history in metadata repository for data variation check

4. Conclusion and Future Work

The proposed methodology would be best suitable for the enterprises which focus on time sensitive in-memory real-time analytics on frequently changing business dimensions with big data to be processed. The proposed idea would be implemented using Hadoop, oracle database since the above applications are following Interval-Range partition which featured with automated partition management and parallel processing. In future, component can be implemented with parallel computation to speed up the real-time analytics.

5. References

1. Batini C, Scannapieca M. Types of data representation. Data Quality Concepts, Methodologies and Techniques. Springer Publications; 2006. p. 6–11.
2. Raj P, Raman A, Nagaraj D, Duggirala S. High performance big data analytics computing systems and approaches. Springer Publications; 2015.
3. Kimball R. Operational data store. The Data Warehouse Lifecycle Toolkit. John Wiley & Sons; 2008. p. 9.4–9.6.
4. Berson. Operational data store. Data Warehousing, Data Mining, & Olap. Tata McGraw Hill Education; 2004. p. 100–50.
5. Operational Data Store [Internet]. [Cited 2015 Feb 13]. Available from: Http://Www.Learn.Geekinterview.Com/Data-Warehouse/Dw-Basics/What-Is-Operational-Data-Store-Ods.Html.
6. Operational Data Store [Internet]. [Cited 2015 May 1]. Available from: Http://Randygrenier.Blogspot.In/2011/02/Operational-Data-Stores-Ods.Html.
7. Plattner H, Zeier A. Memory data management an inflection point for enterprise applications enabling analytics on transactional data. Springer Publications; 2011.
8. In- Memory Database [Internet]. [Cited 2015 May 1]. Available from: Http://Www.Mcobject.Com/In_Memory_Database.
9. Harman K. Learning objects standards, metadata, and repositories. Santa Rosa, California Informing Science Press; 2007.
10. Marco D. Building and managing the metadata repositorya full lifecycle guide. Wiley Publications; 2000.
11. Metadata [Internet]. [Cited 2015 May 1]. Available from: Http://Etl-Tools.Info/En/Metadata.Html.
12. Inmon WH, O'Neil B, Fryman L. Business metadata capturing enterprise knowledge. Morgan Kaufmann; 2010.
13. Bhansali N. Metadata management and data governance. Data Governance Creating Value from Information Asset. Auerbach Publications; 2014. p. 43–64.
14. Rao V. Data partitioning. Oracle Business Intelligence Solutions. Lulu Press. 2015; p. 200–350.

15. Alapati SR. Data partitioning. Expert Oracle Database 11g Administration. Dreamtech Press; 2009. p. 280–91.
16. Data Partition [Internet]. [Cited 2015 Apr 15]. Available from: Http://Docs.Oracle.Com/Cd/B28359_01/Server.111/B32024/Partition.Htm.
17. Kimball R. Data dimensions. The Data Warehouse Lifecycle Toolkit. John Wiley & Sons; 2008. p. 20–150
18. Kimball R, Ross M. Slowly changing parameter. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. John Wiley & Sons; 2013. p. 30–65.