Big Data Research 2 (2015) 102–109

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

Parallel Techniques for Large Data Analysis in the New Version of a Futures Trading Evaluation Service ☆

Xiaoyun Zhou [a,*], Xiongpai Qin [b], Keqin Li [c]

[a] Computer Science Department, Jiangsu Normal University, Xuzhou, Jiangsu, 221116, China
[b] Information School, Renmin University of China, Beijing, 100872, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Article history:
Received 15 November 2014
Received in revised form 11 March 2015
Accepted 13 March 2015
Available online 22 April 2015

Keywords:
Large data analysis
Parallel processing
Futures trading evaluation

ABSTRACT

A futures trading evaluation system is used to help investors analyze their trading history and find out the root cause of profit and loss, so that investors can learn from their past and make better decisions in the future. To analyze trading history of investors, the system processes a large volume of transaction data to calculate key performance indicators (KPI) as well as time series behavior patterns, and concludes some recommendations with the help of an expert knowledge base. This work is based on our early work of parallel techniques for large data analysis for futures trading evaluation service. In our early work, we have used the query rewriting technique to avoid joining between fact table and dimension table for OLAP aggregation queries, and used a data driven shared scanning of data method to compute KPIs for one customer. However, the query rewriting technique cannot eliminate joining for queries which aggregate on an intermediate level of the hierarchy of a dimensional table, so we propose a segmented bit encoding of dimensional table method which can eliminate the joining operation when the query aggregates on any level of the hierarchy of any dimensional table. Furthermore, our previous method perform badly when concurrency is high, so we propose an inter customer data scan sharing scheme to improve system performance in highly concurrent situations. We present our new experimental results.

1. Introduction

With the fast development of economy, more and more investors put money into futures markets with the expectation of making profit. A futures market has the characteristics of high-risk and high-return due to its high leverage ratio. A large number of non-professional investors participate in futures trading. Some of them make profits, and most of them lose money because of their different capabilities. For these participants, what they are most concerned about is to receive some recommendations from experts according to their past trading transactions. We design a futures trading evaluation system to help them. The system is a profit making capability assessment software, which is developed by the authors and GT Futures Brokerage Company.

The system processes historical transaction data of a specific investor, and extracts key performance indicators and trading behavior patterns. Through the analysis of these indicators and patterns, with the help of the knowledge of investment experts, the system gives customers some suggestions on trading capabilities, trading habits, and trading psychology, etc. to help customers identify their shortcoming and improve themselves in the future. In short, according to a customer's historical transaction data, the system generates an assessment report, with the hope that it can help the customer improve his (her) trading capabilities, stop loss, and make profit. The futures trading evaluation system tries to discover some valuable information, i.e., the customer's behavior characteristics buried in the process of trading, from large volume of historical transaction data.

In the evaluation system that we built, the volume of the data is greater than 200 GB, and it is growing at a pace of 300 to 500 MB per day. The data should be analyzed as timely as possible, so that customers can adjust their trading strategies according to the analytic results.

Our contributions in this paper include: (1) we propose a segmented bit encoding of dimensional tables for star schema data, which can eliminate join operation during aggregating on any level of the hierarchy of any dimension; (2) we exploit inter customer data scan sharing, which can improve report generation throughput greatly; (3) we have conducted a series of experiments to evaluate the proposed techniques.

☆ This article belongs to BDA-HPC.

* Corresponding author.
E-mail addresses: wsp0516@163.com (X. Zhou), qxp1990@ruc.edu.cn (X. Qin), lik@newpaltz.edu (K. Li).

http://dx.doi.org/10.1016/j.bdr.2015.03.004
2214-5796/

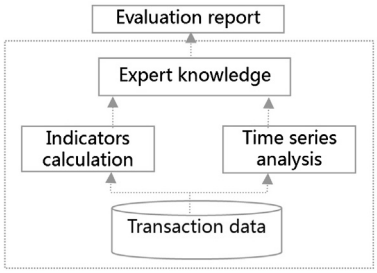

Fig. 1. The structure of the trading evaluation system.

2. Working logic of the futures trading evaluation system

GT Company has accumulated ten years' experience in futures brokerage. Its investment research department has gathered a group of futures market analysts. The core of the futures trading evaluation system is a knowledge base, which has solidified knowledge of these investment analysts.

The concept architecture of the system is shown in Fig. 1. The data for the evaluation system comes from production systems (they can also be downloaded from the China Margin Monitoring Centre Co., Ltd.). The data include daily status of *Holding* contracts, *Open* transactions, *Close* transactions, status of funds, and a number of ancillary information such as customers, varieties, and exchanges.

The indicator computing module calculates 87 key indicators including degree of risk, profit and loss, etc., from the data set of the reporting period. At the same time, the system analyzes 45 time-series patterns from the customer's trading transactions. These indicators and patterns are sent to the expert knowledge base for further processing. Finally an evaluation (assessment) report is generated, and it is sent to the users in the PDF file format.

We have adopted widely used indicators, but how to put indicators into groups and how to use them is the key of the futures trading evaluation system, and depends on experts' knowledge. Since the focus of this paper is to demonstrate some parallel techniques for large data analysis, and due to the requirements of confidentiality, we do not present further details of the knowledge based evaluation algorithm.

3. Parallel techniques for large data analysis

In this section, the scalable data processing architecture is firstly presented, followed by the data encoding scheme for star schema and corresponding query processing algorithms.

3.1. The scalable parallel data processing architecture

In the whole cluster architecture, different nodes are responsible for different jobs as follows.

3.1.1. Different nodes for different jobs

In order to support the increasing volume of data, we use cluster computing to do the job of data analysis. Cluster nodes are divided into three categories, including front-end processing nodes, data processing and analysis nodes, and data loading nodes (see Fig. 2).

Front-end processing nodes are responsible for pre-processing parameterized queries, and handing them over to the data processing and analysis nodes for further processing. We adopt a *Scatter-Gather* style of data processing method. When one of the front-end processing nodes receives an evaluation request, it decomposes the request into parts, and distributes the parts to data analysis and process nodes for real computing. Partially aggregated results are merged in the front-end processing node later, and sent to the expert knowledge base to generate the evaluation report. The report is returned to the client using a URL, which can be used to download the generated PDF file.

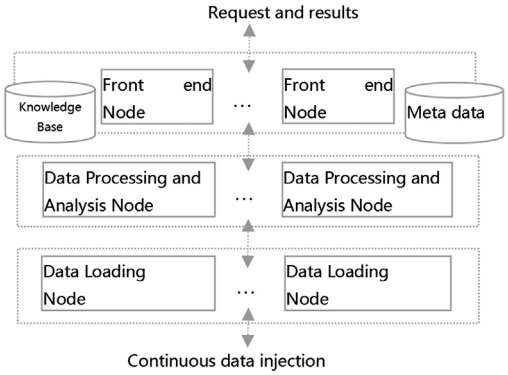

Fig. 2. The parallel data processing architecture.

Table 1
Nodes and functions.

Node	Functions
Front-End Node	Query Pre-processing Task Assignment Result Merging Meta Data Management
Data Processing and Analysis Node	Local Calculation of Indicators Local Calculation of Time Series Patterns
Data Loading Node	Data Extraction Data Transformation Data Splitting and Loading

Meta data about data distribution are stored in front-end nodes, which are used by the query dispatcher to select appropriate data processing and analysis nodes for parallel query processing. All front-end nodes could access a shared database engine which stores the meta data. To support highly reliability of the system, the hot standby technology can be used to protect the database.

Calculation of indicators can be expressed using SQL aggregation queries. Most SQL aggregation queries could be executed in a *Scatter-Gather* manner.

Each data processing and analysis node manages a subset of data, and it is responsible for local calculation of indicators and local time-series analysis, and returning partial results to the front-end processing nodes to merge. For example, the holding profit of a customer can be calculated on data nodes, and the partial results are merged on front end nodes to get the final result.

Since data analysis touches large volume of historical data, and the data are seldom updated (a data *append* is not an *update*, and new data are imported into the system by the data loading nodes), we decided to use non-transactional database storage engines to manage the data on processing and analysis nodes. MySQL is used as the underlying storage engine. By modifying the code base, the transaction management burden is avoided. Compared to database systems with fully transaction management support, the storage engine becomes much lighter, and data accessing speed is increased.

Data loading nodes are responsible for splitting, transforming, and loading new data. We adopt a data distribution scheme that is similar to GFS (Google File System) [1], where each data partition is replicated to at least three nodes for high fault-tolerance. Table 1 summarizes functions of the three types of nodes.

The system architecture is inspired by MapReduce (the Google's large-scale data processing platform). The difference is that we use database engines for data management, rather than a file sys-