Design of Enterprise Economic Dynamic Management System Based on Spark Technology

Lu Zhang and Yipin Yan(✉)

Faculty of Management, Chongqing College of Architecture and Technology, Chongqing 400000, China
z113452873019@163.com

Abstract. Aiming at the problem that the currently used dynamic management system based on Hadoop and B/S architecture is affected by the slow data mining rate, resulting in low efficiency of data dynamic management, a design of enterprise economic dynamic management system based on Spark technology is proposed. Deploy the physical architecture of the Spark-based economic dynamic management system, and build a data warehouse in this architecture to facilitate users to quickly view data in real time. The B/S (browser/server) model is adopted to design data collection modules, business service modules and performance modules to meet the needs of big data analysis and decision-making. When using Spark technology to dynamically adjust the difference data in the database, the rule base needs to be updated in time to convert the automatic conversion system to a detection system. Use the optimization algorithm of Spark Join operator to optimize the entire connection operation, filter out the project data without specific categories in the bank flow data, reduce the data entering the shuffle stage, and design a dynamic management process. It can be seen from the test results that the system has a maximum management efficiency of 92% in a safe environment. In a non-interference environment, the highest management efficiency is 0.95, which has an efficient management effect.

Keywords: Spark technology · Enterprise economy · Dynamic management · Data warehouse · Spark Join operator

1 Introduction

At present, the enterprise finance department does not have a data management and analysis platform. It mainly relies on general office software such as ERP software and UFIDA financial management system to process data. However, general office software such as ERP has problems such as huge system, multiple functions, information redundancy and poor information sharing. In terms of function, such software is mainly used to input data and store data, Instead of collecting and analyzing data, the company's leaders cannot view the enterprise's financial data and understand the current company's

S.-H. Wang and Y.-D. Zhang (Eds.): ICMTEL 2022, LNICST 446, pp. 274–286, 2022.
https://doi.org/10.1007/978-3-031-18123-8_21

operation through the visual information management platform in a timely, accurate and real-time manner. In addition, such software usually has high cost, complex system upgrade, difficult system maintenance and slow data migration. These problems greatly affect the enterprise's office efficiency. Therefore, the dynamic management of enterprise economy is necessary. The previously proposed enterprise economic dynamic management system based on Hadoop uses clusters to complete high-speed operation and storage of data. At the same time, it is transparent to developers and supports agile development. It is mainly composed of HDFS and MapReduce. HDFS not only has high fault tolerance, but also has low hardware requirements and high throughput, You can also access the data in the file system as a stream. MapReduce implements task fragment processing, distributes fragments to multiple nodes through map, and then synthesizes data sets through reduce and loads them into the data warehouse, which truly realizes parallel computing [1]; The economic dynamic management system based on B/S architecture and data warehouse technology are used to realize data query and analysis, so as to ensure that the management can view the analysis results in real time and accurately understand the overall operation of the company [2]. However, the above two systems read the disk and file system more frequently, which makes the data mining speed slower. Therefore, the design of enterprise economic dynamic management system based on spark technology is proposed.

2 System Hardware Structure Design

According to the demand analysis of the economic dynamic system and the characteristics of the big data platform, the physical architecture of the Spark-based economic dynamic management system is shown in Fig. 1.

As can be seen from Fig. 1, in the whole process of designing the system, designers and developers do not need to care about the specific physical environment, but only need to know how to collect the data source. The economic dynamic management system is mainly composed of big data platform, data analysis management platform and MySQL database [3]. The big data platform runs on multiple servers in parallel, which is used to complete the standardized and structured processing of customer transaction behavior, details and other log information, conduct data analysis, and then store the results in the database, so that users can easily query and analyze the results. The data source of the big data platform comes from its own HDFS distributed file system. HDFS is used to store persistent log information, that is, all log information is stored in HDFS, while the corresponding analysis result database only stores the information of the day, which not only ensures the data mining of long-term accumulated big data, but also reduces I/O operations, So as to improve the overall efficiency of the system. The data analysis management platform provides external web services to facilitate users to query the analysis results more clearly and intuitively [4].

2.1 Data Warehouse Design

The traditional data warehouse is combined with the Spark big data platform to expand the real-time analysis function of big data to meet the original business needs. Figure 2 shows the architecture design of the system data warehouse.

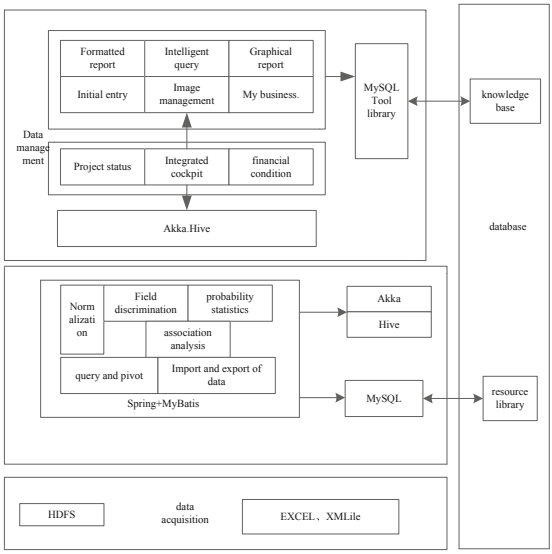

Fig. 1. Schematic diagram of the hardware structure of the Spark-based system

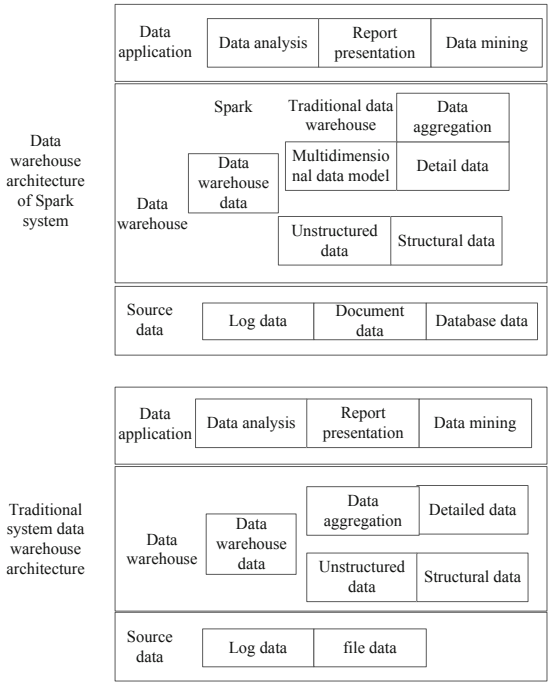

Fig. 2. System data warehouse architecture

As can be seen from Fig. 2, this architecture increases the number of types in the source data layer compared with the traditional data warehouse and expands the data source. In the data warehouse layer, spark is integrated with the traditional data warehouse layer, in which spark is responsible for processing real-time, batch and unstructured data; Other unstructured data is processed by traditional data warehouse. At the same time, the connection between the data source layer and the data warehouse layer has also been changed accordingly. If it is integrated into the big data platform, it needs to be processed by ETL (extraction, conversion and loading) [5]. In the data warehouse layer, identify the data information mode, fractal compress the image and video information [6], extract the initial additional error value, abstract the distribution of error value, save database memory and realize effective data compression [7]. In the design of data warehouse of the whole system, the design of database occupies a key position, and the conceptual structure design of database is the core of the whole database design. E-R diagram is the basic method of conceptual model design. E represents entity and R represents relational model. It is composed of entity, relationship and attribute. If the system function module design gives the specific function module and behavior implementation of the developer, the data structure of the system provides the data storage mode and the correlation between data [8]. Although the system uses HDFS to store unstructured data such as logs, real-time data and data analysis result files, the final normalized analysis results will still be stored in the relational database for users to view the data in real time and quickly.

2.2 Data Acquisition Module

The economic dynamic management system adopts the B/S (browser/server) model. The B end is responsible for generating various requests to the S end, and the S end responds to related requests to complete data collection, which is sorted and stored in a database or file [9]. The data collection layer specifically corresponds to the initial input and business function modules of the system. The initial input mainly enters some basic company information such as organizational framework, employee files, bank files, etc. and some data directly related to the project at the initial stage of the project, such as the owner Files, supplier files, etc., can be entered in the form of direct entry through the browser form input box, or by importing the corresponding existing Excel file. Manual entry ensures the real-time performance of system data, while file import provides convenience for initial batch entry of data [10]. The business module divides different business scenarios according to the different positions of the employees, and enters real-time dynamic business data. Because the business data entered is complex and diverse, there are also many types of entry forms provided [11]. The direct input and file import are the same as those described above. At the same time, in order to meet the needs of big data analysis, it also provides a way to read files from the distributed file system and import batch data from the database. The composition of data acquisition module is shown in Fig. 3.

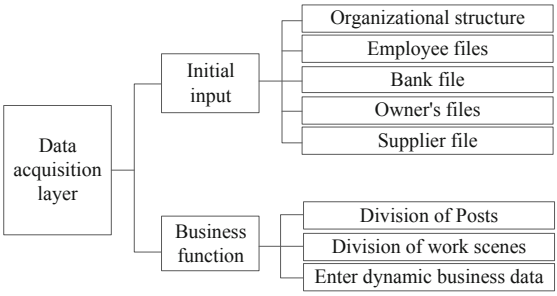

Fig. 3. Composition of data acquisition module

2.3 Business Service Module

The business service module is located on the S side, receives requests from the B side, and uses Java Bean, Servlet, SparkStreaming, and SparkSQL technologies to complete the corresponding processing of each request. The business service layer mainly corresponds to the three business analysis modules of the system's project status, integrated cockpit and capital status [12]. Among them, the project status and funding status belong to the analysis of the data in the traditional data warehouse to get the specific situation of a certain project, and the detailed analysis of the funding status of a certain project in the time dimension; while the integrated cockpit mainly combines large-scale analysis. The data platform analyzes the unstructured data from the file system or the batch structured data in the database to obtain the macro operation of the entire company, such as the geographical distribution of projects in the spatial dimension and the contract time of projects in the time dimension. The system business service module is shown in Fig. 4.

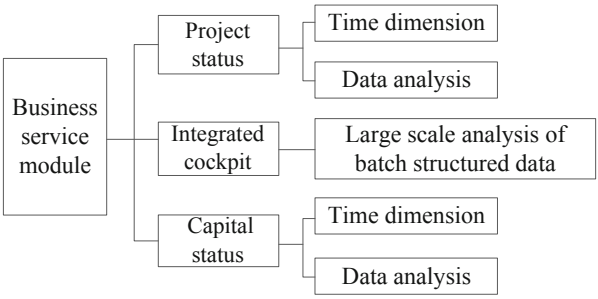

Fig. 4. Composition of business service module

2.4 Performance Module

The presentation module is mainly located at the B end. The data analysis results and data query results are dynamically refreshed and displayed on the page through the web page. It mainly applies HTML and CSS for page layout, jqueryeasyui for page

beautification, jQuery for page logic processing, and Ajax for asynchronous submission and refresh. The core of the presentation layer is the data query module, which mainly includes intelligent query, graphical report, formatted report and image query. Of course, it provides operation interfaces for project status, comprehensive cockpit, fund status, initial entry and my business. The system displays various data processing results through the beautiful and easy-to-use operation interface to meet the final analysis and decision-making requirements.

2.5 Real-Time Data Processing Module

Facing the new challenge of fast real-time streaming data calculation and fast batch calculation, the system needs to find a non-complicated implementation solution instead of combining frameworks for various computing scenarios to ensure concise data in the system Flow direction, so as to better ensure the high responsiveness of the system. Based on the distributed framework Spark and the distributed message queue Kafka as the main data transmission medium, a fast data processing module with two major functions of real-time computing and offline computing is constructed, as shown in Fig. 3.

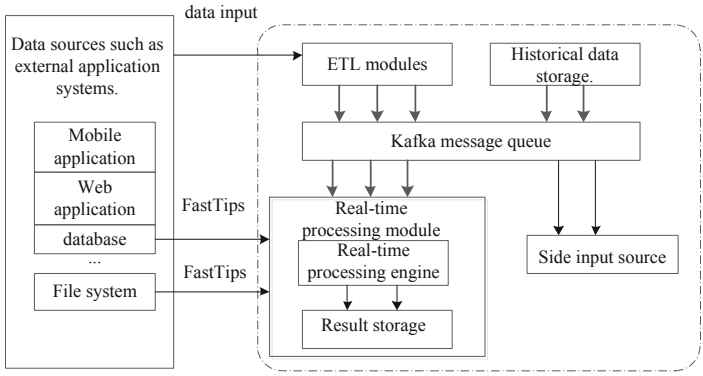

Fig. 5. Schematic diagram of the structure of the real-time data processing module

As can be seen from Fig. 5, the system mainly includes ETL module and real-time processing module. Kafka message queue can be regarded as an internal data transmission bus, which is connected between the two large modules to buffer data and distribute consumption data, so that the system will not go down due to excessive load when a large amount of data is generated by the data source, Kafka is also connected to historical data storage external disk devices and side input data sources.

(1) ETL module obtains the data to be processed by establishing a connection with external data sources. It can adapt to a variety of heterogeneous data sources and data in different formats, and support offline data and stream data, such as mysql, HDFS, Kafka, network socket, etc. one or more data sources can be selected for connection according to business scenarios. The received original data is cleaned, converted and sent to Kafka message queue.

(2) Kafka message queue is a distributed high-performance message queue system. Data receiving and distribution is to receive the converted data from ETL module through publish and subscribe (producer consumer) mode, classify these data according to business topics, facilitate applications to realize business functions, and also play the role of data buffer. In addition, it is also connected with a historical data storage device and a side input data source, which can be extended according to the actual needs of users.
(3) The purpose of storing historical data is to facilitate the tracing of historical data, to be able to analyze the complete and large-scale historical data in the past, and to output the specified range of historical data incrementally to the real-time processing module, thereby enabling data "replay" And the "recording and broadcasting" function, generally use a distributed storage system like HDFS.
(4) The side input data source is to save some metadata transferred from external application systems, because generally metadata changes and updates are not too frequent, these metadata are sometimes indispensable when processing data, so by establishing a side Save these metadata by inputting the data source, which saves the bandwidth of data transmission, and only updates when it needs to be updated. Redis or HDFS can be used as the storage device.
(5) The real-time processing module pulls data through the Kafka message queue for calculation in the manner of consumers, and provides calculation methods such as statistical analysis, complex aggregation, and machine learning. Users can implement their own calculation logic according to business needs. The calculated results are stored in the local file system or database, and provide an interface for external applications to quickly query and view the results in real time.

3 System Software Part Design

3.1 Enterprise Economic Data Mining Based on Spark

Spark is a fast and universal cluster computing platform for large-scale data processing. It provides services in the form of Web Service. Users can access the service by calling the relevant Web Service interface, or perform two operations on the WebService interface provided by the system. Development to provide richer services.

Emphasis is placed on the analysis of dynamically changing business economic data, and the detection of dynamic change data is regarded as a data analysis process. This process requires mining and analysis of massive data in order to obtain dynamic changing business economic data under the normal mode. When the mining process reaches a state of dynamic data changes, the automatic conversion system needs to be converted to a detection system. The rule base needs to be updated in time during this process, as shown below:

Let $R = \{r_1, r_2, \cdots r_m\}$ be a collection of projects, and $F = \{f_1, f_2, \cdots f_n\}$ represent a collection of database projects. Each transaction T has a separate project subset $F \subseteq R$, and has a unique identifier. The association rule is the logical implication formula of the formal term $A \Rightarrow B$, and $A \subset F$, $B \subset F$, $A \cap B = \emptyset$.

The support of the logical implication formula $A \Rightarrow B$ in the rule base includes the percentages of both A and B transactions. The condition concept is $P = (B/A)$, which can be expressed as:

$$\begin{aligned} &\sup\ part(A \Rightarrow B) = P(A \cup B) \\ &confidence(A \Rightarrow B) = P(B/A) \end{aligned} \tag{1}$$

Confidence is responsible for measuring the accuracy of relevant rules, and support is responsible for measuring the accuracy of relevant relationship matrix. Through confidence and support, we can evaluate how representative the rule is in the whole mining process. Obviously, the greater the support, the more tense the relationship rules.

When using Spark technology for dynamic data mining, it is necessary to combine the characteristics of dynamic data in large differential database. The specific implementation process is as follows:

Step 1: Obtain interest

Through the query of related data, the mining target and the reference set are collected into the related database. According to the above association rules, the percentage of all things is regarded as the expected confidence level, and comparative analysis is performed. The obtained interest level W is:

$$W = \frac{confidence(A \Rightarrow B)}{\sup part(A \Rightarrow B)} \tag{2}$$

The interest degree W in formula (2) can include the correlation degree of all logical implication formula $A \Rightarrow B$.

Step 2: Set the minimum limit matrix

At the level of rough calculation, the mining target is set as the minimum limit matrix, and the corresponding matrix values are all stored in the database, where the attribute value is a single value.

Step 3: Calculate the matrix support

There are different degrees of support between different matrices, the calculation method is:

$$confidence\{A, B\} = \mathrm{W} \times confidence(A) \times confidence(B) \tag{3}$$

The support degree of the minimum threshold is obtained according to formula (3), and the minimum database is formed through the expenditure degree.

Step 4: check the matrix relationship

Check the matrix relationship between common databases, eliminate the data inconsistent with the actual relationship, and form a topology.

Step 5: form a new topology

The topological relations obtained in the above contents are generalized to form a new topological structure, so as to realize the mining of enterprise economic dynamic management data.

3.2 Join Operation Based on Spark Join Operator

Data connection is a key step, so the optimization algorithm of Spark Join operator is used to optimize the entire connection operation. The specific analysis process is as follows:

Step 1: Use Kafka to receive real-time messages from external systems in real time; Copy real-time bank log data to HDFS file system;

Step 2: Use the hdfsutii toolkit to read the data in the HDFS file system; Use the data generator to generate simulation data to verify the reliability of the program;

Step 3: Convert data to RDD; Call rddoperatorutils class to group RDD and aggregation operations; Call bfjoin to perform RDD connection operation;

Step 4: Store the result set in MySQL; Read the data in MySQL and display the analysis results.

Bank flow data has the characteristics of complex payment types, which makes it impossible to accurately count many subtle types of payments. At this time, they can be classified into the same type of data. Generally, users pay more attention to certain data, such as a certain item in a certain period of time. The total income, total expenditure, the proportion of certain types of funds in the total expenditure/income of the project and other macro information, etc., therefore, six types of data are divided.

In the specific processing, when the RddBFJoin is processed, the BF Join algorithm designed in the previous section is used. In this business, before the connection of rddl and rdd2 is executed, the item data that does not have a specific category in the bank flow data can be filtered out. Reduce the data entering the Shuffle stage.

3.3 Dynamic Management Process Design

Spark provides a very direct way for users to submit compiled Spark applications, spark-submit scripts. Users only need to package the application and its dependencies together, and then use the script to submit to the cluster, and set the application deployment mode, program entry, parameters and other information to complete the submission, and start the application task on the cluster. Enter the cluster management interface, the first is the list of data types to be created, including data type name, brief description, jar package name, jar package location, creation time, etc. The list is sorted by the data type creation time. Buttons for creating, updating, and deleting data types are provided at the top of the list, and pagination components are provided at the bottom of the list. When creating a data type, click the Create button to pop up a window, enter the name of the data type (check the name), click the upload button, select the corresponding jar package, and upload, or you can briefly describe the data type.

Dynamic management process design, as shown in Fig. 6.

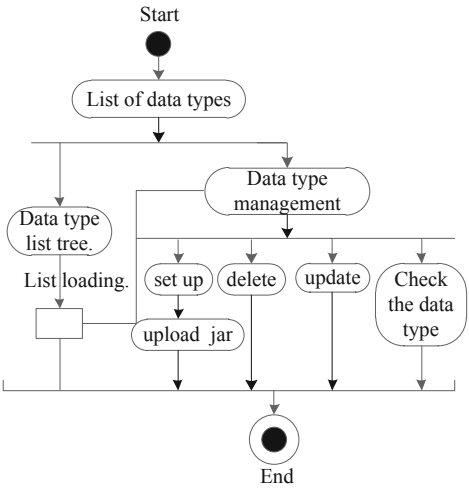

Fig. 6. Dynamic management process

When updating the data type, select the data type to be updated, and click the update button to pop up a window. The window type is the same as the pop-up window for creating the data type, but the data type information already exists in the window. You can modify the information to be updated. If you need to modify the corresponding jar package, re upload the jar package, and the original jar package will be deleted automatically. When deleting a data type, select the data type to delete and click the delete button to delete it. After deleting the data type, the corresponding data type information in the data type information table in the database will be deleted, and the corresponding data type jar file in the file system will be deleted automatically.

4 System Test

This test is derived from the ATL data processing analysis system project test report. The purpose is to summarize the test phase of the enterprise economic dynamic management system based on Spark technology and analyze the test results, evaluate whether the system meets the requirements, and at the same time, discover the system as much as possible Bugs and defects to ensure product quality.

4.1 Test Data

The data used in this experiment are nearly 300W bank journal data of a company. The following information is extracted. The specific data set format is shown in Table 1.

Table 1. Bank flow data field information

Field description	Type
Abstract	String
Borrow	Float
Loan	Float
Balance	Float
Bank account	Int
Transaction type	String
Transaction hour	String

Based on the data in Table 1, the system test and analysis are carried out.

4.2 Test Results and Analysis

Use the Hadoop-based enterprise economic dynamic management system, the B/S-based economic dynamic management system, and the Spark technology-based enterprise economic dynamic management system to compare and analyze the management efficiency in a safe environment and a disturbed environment. The result is shown below.

Safety Environment

In a safe environment, the management efficiency of the three systems is compared and analyzed, and the results are shown in Fig. 7.

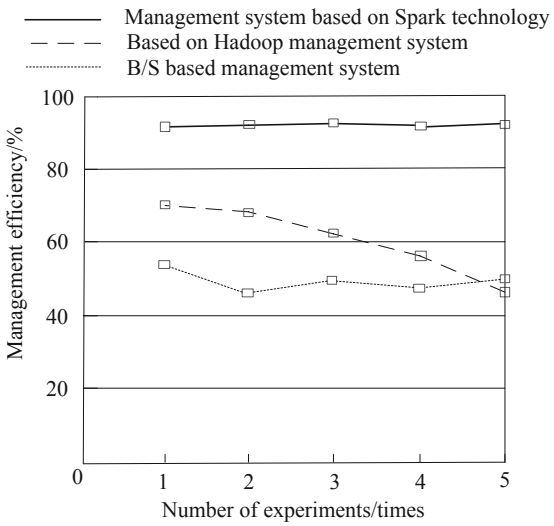

Fig. 7. Comparative analysis of the management efficiency of the three systems in a safe environment

As can be seen from Fig. 7, the management efficiency of the management system based on Spark Technology under the security environment is always maintained at more than 90%; The management efficiency of Hadoop based management system is 46%–70%; The management efficiency of the management system based on B/S is 46%–56%. It can be seen that the management efficiency of the management system based on spark technology proposed in this paper is higher than that of the other two systems.

Interference Environment

In order to verify the actual operation performance of the system designed in this paper, the management efficiency of the three systems under different concurrency is analyzed, and the results are shown in Fig. 8.

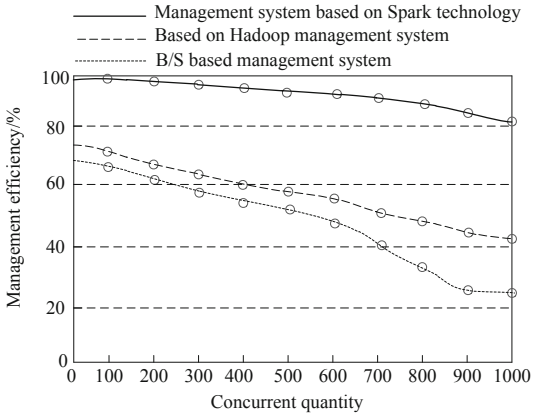

Fig. 8. Comparison and analysis of management efficiency of three systems under interference environment

As can be seen from Fig. 8, when the concurrency of the three systems increases, the management efficiency gradually decreases. The management efficiency of the management system based on Hadoop fluctuates between 42% and 74%, and the management efficiency is low; The management efficiency of the management system based on B/s fluctuates between 25% and 68%, and decreases greatly when the concurrency is 500–900, so the system is relatively unstable; Although the management efficiency of the management system based on spark technology proposed in this paper has been declining, it can always be maintained at more than 80%; It shows that the management system proposed in this paper can maintain 80% management efficiency under the condition of 1000 concurrency.

5 Conclusion

Based on the new management and analysis requirements of enterprises for financial data, an economic dynamic management system integrating multi-source data collection, statistics, analysis and management is constructed. The hierarchical design idea is used in the construction of the whole system, the mature spark architecture is used to build

the server side of the system, and the spark big data technology is used to complete the basic development of the system.

This paper mainly designs and develops the economic dynamic management system based on spark technology, studies the impact of data connection operation in spark on data processing and analysis in the economic dynamic management system, and optimizes the large table equivalent connection algorithm in spark. However, at present, the system has only completed independent development, and cannot interact with external systems. At the same time, some functions have not been migrated from traditional platform to large number platform. In addition, there are still many deficiencies in the optimization of data connection algorithm. Here, we put forward our prospects for the future:

Although the proposed optimization algorithm BF join improves the connection performance by effectively filtering the data that does not meet the connection conditions, the pre filtering of RDD partitions may lead to data skewing of data with roughly the same size due to different matching degrees of filtering conditions, resulting in the decline of connection performance. Therefore, the optimization of data skewing connection algorithm needs to be further studied.

References

1. Tang, X., Song, Z.: Evolutionary game analysis of collaborative consumption behavior of participants in sharing economy. Enterp. Econ. **461**(01), 66–72 (2019)
2. Wang, S., Xue, X., Ge, Y., et al.: Geo-Economic strategies assessment based on computational experiment: taking the customs clearance time adjustment of China-Indonesia and China-Vietnam as an example. Econ. Geogr. **39**(02), 12–21+63 (2019)
3. Li, G.: The benefit distribution of tourism enterprise strategic alliance under active cooperation mode. Enterp. Econ. **03**, 145–152 (2020)
4. Xu, W., Guo, S., Wang, L., et al.: Dynamic control system of material flow schedule for building material equipment manufacturing enterprise oriented to production task. Comput. Integr. Manuf. Syst. **25**(03), 105–118 (2019)
5. Zhao, X., Zhou, Z., Wu, Y., et al.: Construction of village resource management system based on perspective of poverty alleviation development. Mod. Electron. Tech. **43**(10), 103–107 (2020)
6. Liu, S., Bai, W., Liu, G., et al.: Parallel fractal compression method for big video data. Complexity **2018**, 2016976 (2018)
7. Liu, S., Fu, W., He, L., Zhou, J., Ma, M.: Distribution of primary additional errors in fractal encoding method. Multimedia Tools Appl. **76**(4), 5787–5802 (2014). https://doi.org/10.1007/s11042-014-2408-1
8. Su, J., Li, Y.: Practice teaching reform of agricultural and Forestry Economics and Management undergraduate major based on rural revitalization strategy. For. Econ. **41**(04), 94–98 (2019)
9. Li, J.: Theoretical design and exploration of modern economic management system in public hospitals. Friends Account. **21**, 2–8 (2020)
10. Liu Y.: Study on the particularity and innovation path of village collective Economic Management in China. Agric. Econ. (04), 40–42 (2020)
11. Zhao, C., Sun, J.: Try to discuss computer information technology and economic management optimization integration. Economics **3**(4), 85–86 (2020)
12. Yu, S.: Analysis on internal control method of enterprise economic management risk. Economics **4**(1), 21–23 (2021)