ISSN 2616-7433 Vol. 6, Issue 5: 48-55, DOI: 10.25236/FSST.2024.060507

Design and Realization of Tax Big Data Supervision System Based on Distributed Real-Time Control

Sitong Chen[1,a,*], Yanbin Long[1,b]

[1]University of Science and Technology Liaoning, Anshan, China
[a]2934788312@qq.com, [b]1034182681@qq.com
[*]Corresponding author

Abstract: *With the progression of tax informatization, there has been a substantial surge in real-time data generated by tax systems across various levels. This necessitates a more robust system for real-time supervision and efficient data processing. However, the traditional centralized management and control system falls short in terms of timeliness and resource utilization when confronted with voluminous data. To address these challenges, this paper introduces a tax big data supervision system and method rooted in distributed real-time management and control. By implementing this distributed system in both superior and subordinate units, the proposed approach facilitates real-time data access, comparison calculations, and alarm notifications. Consequently, it enhances data processing speed and optimizes resource utilization, thereby proving pivotal in improving tax supervision efficiency and optimizing resource allocation.*

Keywords: *tax; big data; Distributed real-time control; Kafka; Flink; ElasticSearch*

1. Introduction

With the swift evolution and widespread adoption of information technology, the tax domain is experiencing a profound transformation in data management. Daily operational and administrative activities within the tax system at various levels generate a substantial volume of real-time data. This data not only documents the tax-related activities of enterprises but also reflects the pulse of the market economy, serving as a critical foundation for tax oversight and decision-making. However, the explosive growth of this data poses unprecedented challenges for tax departments[1].

The conventional tax data processing and supervision system, characterized by a centralized architecture, requires the aggregation of all data to a central server for processing and analysis. When confronted with massive datasets, this architecture often proves inadequate, resulting in inefficient data processing, prone to delays, and unable to meet the stringent demands of real-time accuracy in tax supervision. Furthermore, centralized systems are associated with elevated risks of single-point failures, high resource consumption, and limited scalability, rendering them ill-suited to handle the demands of tax big data processing[2].

To address these challenges, this paper introduces a tax big data supervision system and method grounded in distributed real-time control. This system, leveraging a distributed architecture, disperses data across multiple nodes for parallel processing and analysis, significantly enhancing data processing efficiency and real-time performance. Simultaneously, the system incorporates cutting-edge data control technologies, enabling comprehensive supervision and intelligent analysis of tax big data, and providing the tax department with more precise, timely, and comprehensive decision-support[3].

Specifically, this paper first delves into the challenges and limitations encountered in tax big data processing. It then expounds on the principles and advantages of distributed real-time control technology. Subsequently, it details the architectural design and implementation methodology of the tax big data supervision system, emphasizing critical aspects such as data collection, storage, processing, analysis, and visualization. Finally, the paper validates the system's effectiveness and superiority through rigorous experiments, envisioning its extensive application prospects in tax supervision[4].

In summary, this research aims to propel the informatization and intelligentization of tax supervision, elevate the service standards and supervisory efficiency of tax departments, and provide robust support for the healthy development of the market economy and the enhancement of taxation law enforcement. By harnessing distributed real-time control technology, we are confident in overcoming the limitations of traditional tax data processing and supervision systems, achieving efficient processing and accurate supervision of tax big data, and injecting new vitality into the development of the tax domain[5].

2. Project content and methodology

2.1 System architecture

In order to meet the needs of real-time processing and supervision of tax big data, the tax big data supervision system designed in this paper adopts a distributed architecture and combines a variety of advanced technical components. The system mainly includes the distributed search engine ElasticSearch and the distributed real-time control system deployed at the upper level and each lower level[6].

In the upper-level distributed real-time control system, its core function is to receive and manage control rules, as well as to summarize and display alarm information from lower-level systems. Specifically, the higher-level system receives user-input control rules through the user interface, which define the compliance criteria and processing logic of tax data. The parent system then disseminates these rules to all lower-level distributed real-time control systems to ensure that all levels of data processing follow uniform rules and standards[7].

At the same time, the subordinate distributed real-time control system is responsible for processing the massive tax data in the data source at the same level. These data sources may include tax declaration records, invoice information, enterprise tax files, etc. The lower level system writes these data into the distributed message queue Kafka in real time. Kafka's high throughput and low latency characteristics ensure the real-time and reliability of the data. After the data is written to Kafka, the lower level system uses the distributed streaming computing engine Flink to perform real-time calculation and processing of the data in Kafka. Flink's powerful stream processing capability enables the system to carry out real-time comparison and calculation on massive data, and write the data conforming to the control rules into the alarm information in Kafka.

Finally, the lower level system pushes the generated alarm information to the upper level system. The superior system summarizes and displays these alarm information, provides the visual query function of the user interface, and facilitates users to monitor the compliance and processing of tax data in real time. At the same time, the superior system also saves the alarm information to the distributed search engine ElasticSearch, so that users can conduct historical query and analysis[8].

2.2 Methodological process

The specific process of this method is as follows.

1) The higher-level distributed real-time control system receives control rules entered by users through the user interface and stores them in the system.

2) The higher-level system sends the stored control rules to the lower-level distributed real-time control systems, ensuring that all levels of data processing follow uniform rules and standards.

3) Distributed real-time control systems at all levels write massive tax data from their own data sources to the distributed message queue Kafka in real time. This step ensures the real-time and reliability of data.

4) Use Flink, a distributed streaming computing engine, to compare and calculate the real-time data in Kafka. Specifically, Flink processes and analyzes real-time data according to preset control rules, and marks the data that meets the rules as alarm information.

5) Write the generated alarm information into Kafka so that the lower level system can push it to the upper level system. This step realizes the real-time generation and transmission of alarm information.

6) After the lower level system pushes the alarm information to the upper level system, the upper level system will summarize and display the information. The user interface provides a visual query function to facilitate users to monitor the compliance and processing of tax data in real time. At the same time, the superior system also saves the alarm information to ElasticSearch for the user to conduct historical query and analysis. This step realizes the visualization and persistent storage of alarm information.

3. System implementation and deployment

3.1 System modules and functions

3.1.1 Control management module

The control management module is one of the core components of the system, which is mainly responsible for the formulation, issuance and management of control rules. Through this module, the higher-level units can formulate a series of control rules, which define data compliance standards, anomaly detection logic and corresponding processing measures. Once the rules are formulated, the control management module will send them to each lower-level distributed real-time control system to ensure the consistency of data processing in the whole tax system.

3.1.2 Data access module

The data access module is responsible for accessing the real-time data generated by tax systems at all levels to the distributed real-time management and control system. This module needs to process data from different data sources, including tax declaration data, invoice data, enterprise tax information, etc. The data access module monitors data changes in real time by establishing a connection with the data source, and transmits the newly generated data to the distributed message queue Kafka quickly and accurately, providing a data basis for subsequent real-time processing.

3.1.3 Management and Control Comparison Module

The control and comparison module uses the distributed streaming computing engine Flink to process and analyze the real-time data in Kafka. The module will compare and calculate the real-time data according to the control rules issued by the control management module, and check whether the data meets the preset compliance standards. Once abnormal data is found, the control and comparison module will immediately generate alarm information and write it into Kafka so that the lower system can push it to the upper system.

3.1.4 MySQL database

MySQL database plays the role of storing and managing data in the system. It is mainly used to store user information, control rules, alarm information and other key data. Through MySQL database, the system can realize persistent storage and efficient query of data, and provide users with stable and reliable data services.

3.2 System deployment

In order to ensure the stability and efficiency of the system, the distributed real-time control system proposed in this paper adopts the distributed deployment method. Specifically, the superior distributed real-time control system is deployed in the superior unit, which is responsible for the management of the control rules of the whole tax system and the summarization of alarm information; while the distributed real-time control system of each lower level is deployed in each lower level unit, which is responsible for the processing of real-time data generated by the data source of this level. This deployment mode fully utilizes the parallel processing capability of the distributed system, which effectively improves the data processing efficiency and system scalability.

In addition, the system also includes a distributed control platform, which provides a unified access portal and management interface for system users. Through the distributed control platform, users can easily manage their own control rights, query alarm information and monitor the system's operation status. The introduction of the platform further simplifies the operation process of the system and improves the user experience.

3.3 Procedures for the realization of functional modules

In general, this paper describes in detail the modules and their functions of the distributed real-time control system, and introduces the deployment mode of the system. Through the realization and deployment of this system, we can realize the efficient processing, real-time supervision and intelligent analysis of tax big data, and inject new vitality and power into the development of tax field.

The program to implement the above functional modules will involve the integration of multiple components and technologies, including distributed message queues (such as Kafka), distributed streaming computing engines (such as Flink), databases (such as MySQL), and possible Web service frameworks (for management and control platforms). The following is a simplified example of how to use these technologies to implement the above functional modules. Please note that this is only a conceptual example, and the actual implementation will be more complex.

3.3.1 Control management module (pseudo-code)

```
java
// the category of control rules
class ControlRule {
    String ruleId;
    String description;
    // ... Other rule attributes
}
// Control management services category
class ControlManagerService {
    // Storage control rules, which may be databases or configuration centers in practice
    Map<String, ControlRule> controlRules = new HashMap<>();
    public void addRule(ControlRule rule) {
        controlRules.put(rule.ruleId, rule);
        //Distribute rules to lower level systems (through message queue, RPC, etc.)
    }
    public void removeRule(String ruleId) {
        controlRules.remove(ruleId);
        // Notify subordinate systems of the removal of rules
    }
    public List<ControlRule> getAllRules() {
        return new ArrayList<>(controlRules.values());
    }
}
```

3.3.2 Data access module (using Kafka Producer API)

```
java
import org.apache.kafka.clients.producer.*;
//Kafka producer configuration
Properties props = new Properties();
props.put("bootstrap.servers", "localhost:9092");
props.put("key.serializer", "org.apache.kafka.common.serialization.StringSerializer");
props.put("value.serializer", "org.apache.kafka.common.serialization.StringSerializer");
// create producers
Producer<String, String> producer = new KafkaProducer<>(props);
// data access methods
public void ingestData(String topic, String data) {
    ProducerRecord<String, String> record = new ProducerRecord<>(topic, data);
    producer.send(record, new Callback() {
        @Override
        public void onCompletion(RecordMetadata metadata, Exception exception) {
            if (exception != null) {
                // handling exceptions
            } else {
                // The data was sent successfully
            }
        }
    });
}
// Close the producer
producer.close();
```

3.3.3 Control comparison module (using Flink)

```
java
import org.apache.flink.streaming.api.datastream.DataStream;
import org.apache.flink.streaming.api.environment.StreamExecutionEnvironment;
import org.apache.flink.streaming.connectors.kafka.FlinkKafkaConsumer;
//Flink environment configuration
StreamExecutionEnvironment env = StreamExecutionEnvironment.getExecutionEnvironment();
env.setParallelism(1);
//Kafka Consumer Configuration
Properties properties = new Properties();
properties.setProperty("bootstrap.servers", "localhost:9092");
properties.setProperty("group.id", "test");
//Create Flink Kafka Consumer
FlinkKafkaConsumer<String> consumer = new FlinkKafkaConsumer<>(
    "tax-data-topic",
    new SimpleStringSchema(),
    properties
);
//Read data from Kafka
DataStream<String> stream = env.addSource(consumer);
// Control comparison logic (pseudo-code)
DataStream<String> alerts = stream.filter(new FilterFunction<String>() {
    @Override
    public boolean filter(String taxData) throws Exception {
        //Analyze taxData and apply control rules
        //If the data does not conform to the rules, return true
        return isViolation(taxData);
    }
});
//Write alarm information to Kafka (Flink Kafka Producer needs to be configured)
alerts.addSink(...);
//Execute Flink Job
env.execute("Tax Data Monitoring Job");
```

Please note that the above code is only an example and does not include complete error handling, resource management and optimization. In practical applications, you need to consider the use of connection pooling, exception handling, logging, monitoring, performance optimization and other practices. In addition, the configuration and deployment of Flink jobs will be more complex, usually involving advanced concepts such as cluster settings, state management, fault tolerance processing, and watermarks.

4. Experimental results and analysis

After a series of experiments, the tax big data supervision system and method proposed in this paper show significant advantages. The following is a detailed analysis of the experimental results:

4.1 Improved timeliness

Compared to traditional centralized control systems, the new system exhibits significantly lower latency when processing vast quantities of tax data. This translates into a substantially shorter timeframe from data entry to the generation of processing results, empowering the tax department to swiftly react to a wide range of scenarios—whether routine tax matters or unforeseen tax issues.

Firstly, the new system stands out in terms of timeliness, offering a marked improvement over the conventional centralized management and control system. When handling extensive tax data, the new system's delay is notably diminished, meaning the time it takes for data to move from input to output is considerably reduced. This enhancement allows the tax department to promptly obtain processing outcomes, facilitating a swift response to both routine tax affairs and emergency tax events.

For the tax department, enhancing time efficiency is paramount. In daily operations, rapid data processing and analysis enable tax officials to accomplish their duties more efficiently, minimizing waiting times and ultimately boosting productivity. The new system's quick response capability is particularly indispensable during unexpected situations, such as monitoring and addressing tax fraud, evasion, and other such behaviors. By analyzing vast datasets in real time, the system assists the tax department in promptly pinpointing issues and taking timely action to prevent losses from escalating.

Moreover, this improved timeliness contributes to enhancing the overall quality and precision of tax administration. Swift data processing and analysis empower the tax department with a more comprehensive understanding of the tax landscape, facilitating the timely identification and rectification of potential errors or omissions. This ensures the accuracy and fairness of tax collection while providing robust data support for the tax department to make more informed and rational decisions.

In summary, the new system's notable advantages in timeliness have delivered heightened operational efficiency, more precise management decisions, and an enhanced service experience for the tax department. These benefits are unparalleled by traditional centralized management and control systems, highlighting the significant value of the new system in practical applications.

4.2 Optimization of resource utilization

The new system not only improves the speed of data processing, but also optimizes the utilization of computing resources. Through reasonable resource allocation and scheduling strategies, the system is able to process a large amount of data while maintaining high resource utilization efficiency and avoiding waste of resources.

Secondly, the optimization results of the new system in terms of resource utilization are also quite remarkable. When traditional data processing systems face massive tax data, they often have uneven resource allocation, some resources are overloaded and others are idle. However, the new system successfully solves this problem by introducing reasonable resource allocation and scheduling strategies.

The new system is capable of dynamically adjusting the allocation of resources according to the real-time demand for data processing and the utilization of resources. This means that when processing large amounts of data, the system is able to ensure that each processing task receives sufficient resources, while avoiding over-allocation and waste of resources. This kind of refined resource management not only improves the efficiency of resource use, but also enables the system to maximize the data processing capacity under limited resources.

In addition, the resource optimization strategy of the new system has helped to reduce the operating costs of the tax department. By reducing unnecessary resource consumption and waste, the new system can meet data processing needs while reducing the energy consumption and maintenance costs of hardware equipment. This is an important advantage for tax departments that need to process large amounts of tax data.

Overall, the optimization of the new system in terms of resource utilization not only improves the data processing capacity of the system, but also brings more economical operating costs to the tax department. This optimization is an important progress in the field of tax big data processing, and also provides a useful reference for the design and development of other similar systems.

4.3 High throughput

Experimental results show that the new system is able to maintain a high throughput when processing a large amount of concurrent data. This means that the system is able to process more data in a short period of time to meet the demand of the tax department for big data processing.

Real-time, accurate information support: The new system provides real-time, accurate information support for tax supervision by identifying and processing data that meets control rules in a timely manner. This is critical for tax authorities as it helps them make more informed decisions and improve the efficiency and accuracy of tax administration.

Good scalability and adaptability: In addition to the above advantages, the new system also shows good scalability and adaptability. Whether adding new data processing modules or adapting to new tax policy changes, the system can be flexibly expanded and deployed to meet the ever-changing needs of the tax department.

In addition, the new system has been experimentally verified to exhibit excellent performance in handling large amounts of concurrent data, i.e., it is able to maintain a high throughput. Throughput is an important indicator of a system's ability to process data, which indicates the amount of data that can be successfully processed by the system in a unit of time. For the tax department, dealing with huge amount of tax data is a great challenge, and the high throughput feature of the new system enables it to meet this challenge easily.

Specifically, the new system achieves efficient processing of large amounts of concurrent data through the use of advanced concurrent processing technology and optimization algorithms. This means that the system is able to process more data in a short time, whether it is daily tax declaration data or complex tax audit data, all of which can be processed quickly and accurately. This high throughput feature not only meets the tax department's demand for big data processing, but also provides more powerful data support for tax management.

The benefits of high throughput are manifold. First of all, it improves the speed and efficiency of data processing, enabling the tax department to obtain the required information in a more timely manner, thus speeding up the decision-making process. This is important for quick response to tax problems and formulation of reasonable tax policies. Secondly, high throughput also helps to reduce the load and pressure of the system, ensuring that the system can maintain stable performance when dealing with large amounts of data. This is essential to ensure the continuity and reliability of tax administration.

To sum up, the high throughput of the new system gives it significant advantages in processing massive tax data. This advantage not only improves the efficiency and accuracy of tax management, but also provides tax departments with more powerful data processing capabilities, laying a solid foundation for future tax management innovation.

5. Conclusions

This paper has successfully designed and implemented an innovative tax big data supervision system and its methods. This system is based on distributed real-time management and control technology. By deploying a distributed real-time management and control system between superior and subordinate units, it ensures real-time access to data, efficient comparison and calculation, and timely alarm push. Through experimental verification, this system and method not only significantly improve the timeliness of data processing and resource utilization, but also provide a solid technical support and accurate information basis for tax supervision.

Looking ahead, we will continue to optimize the system's architectural design and algorithmic implementation, with a view to further enhancing the system's processing efficiency and intelligence. Specifically, we will explore the introduction of more advanced distributed computing frameworks and machine learning algorithms to optimize the data processing process and improve the accuracy of analysis and decision-making. At the same time, we will also actively seek opportunities to apply the system to more scenarios and fields, with a view to bringing its value into play in a wider scope and promoting the modernization and intelligence of tax supervision.

References

[1] Pang Kaiwen. Research on enterprise tax risk management in the context of big data [J]. Economy and Trade of the Times, 2021(07).

[2] Sang Zhijie. Discuss enterprise tax risk management under big data [J]. Management and Technology of Small and Medium-sized Enterprises (the first ten day issue), 2020(11).

[3] Ren Changling. Analysis and strategy of enterprise tax risk management in the context of big data [J]. Small and Medium-sized Enterprises in China, 2020(11).

[4] Hou Wen. Analysis and strategy of enterprise tax risk management in the context of big data [J]. Marketing, 2020(42).

[5] Liang Qian. Application of big data technology in tax risk prevention and control [J]. Scientific and Technological Information, 2020(26).

[6] Rong Yan. Analysis and strategy of enterprise tax risk management in the context of big data [J]. Chinese and Foreign Entrepreneurs, 2020(19).

[7] Tang Ziying. Research on enterprise tax risk management in the context of big data [J]. Tax Payment, 2020(17).

[8] Tan Juan. Analysis and strategy of enterprise tax risk management in the context of big data [J]. Tax Payment, 2020(04).