A new whole process analysis framework big data based for E-commerce application

Dujuan Zhou[a], Junshen Hong[b], Junting Ou[c], Zizhao Yuan[d], Fanbiao Bao[e]

[a]17424@bitzh.edu.cn; [b]221205101105@bitzh.edu.cn; [c]211205101897@bitzh.edu.cn; [d]210110101171@bitzh.edu.cn; [e]04033@bitzh.edu.cn

Beijing Institute of Technology, Zhuhai, China

Abstract. With the rapid development of the Internet data era, the number of users of e-commerce websites has increased dramatically, and the corresponding operation data also shows a surge trend. However, many e-commerce enterprises still use traditional databases, which are difficult to effectively handle massive data. This paper proposes a full-process processing analysis system based on Hadoop by deeply analyzing the Hadoop big data ecosystem technology. Innovative use of Kafka as a buffer to prevent server crashes. A customized interceptor is used on Flume for data cleaning to avoid subsequent data warehouse parsing problems, and Spark is used to replace the underlying engine of Hive to improve computational efficiency. The whole system includes four modules: data collection, data warehouse, fully automated task scheduling and data visualization, which can effectively reduce repetitive data development of e-commerce big data enterprises in practical applications, efficiently analyze massive and real historical data, and generate visualization reports.

Keywords: Data Warehouse, Kafka buffering, Data visualization, Hadoop ecosystem, Spark's underlying engine

1. Introduction

With the big data era of the Internet, the popularity of smartphones and online shopping has driven the rise of e-commerce [2] companies. However, some companies still use traditional databases [1], which make it difficult to process large-scale data and perform comprehensive analysis, resulting in a competitive disadvantage. As disk prices have fallen, companies are adopting big data platforms for centralized analysis. Despite the gradual maturation of big data technology, the industry chain is still not perfect, and big data applications are still in the primary stage, with a broad development prospect in the future.

This paper proposes a comprehensive framework design strategy for data analysis system that integrates massive data collection, storage, computation and analysis. Aiming at the current demand for large-scale data analysis of ecommerce websites in the Internet industry [3], the fullprocess big data processing and analysis platform based on Hadoop is designed by studying the related technologies of Hadoop big data ecosystem [4]. Figure 1 shows that the e-commerce data analysis system is divided into five major functions, i.e., data collection function, data warehouse function, data quality management function full-process task scheduling function and data visualization function. Each functional module contains a different big data framework.

ECIT 2024, March 15-17, Changsha, People's Republic of China
Copyright © 2024 EAI
DOI 10.4108/eai.15-3-2024.2346165

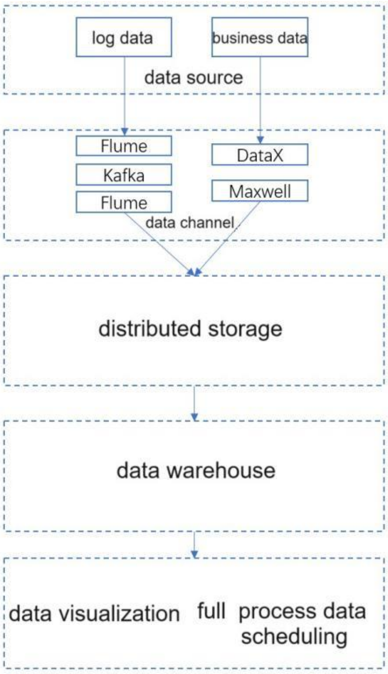


Fig. 1. Module architecture diagram.

The system uses Flume to collect all user behavior data for the enterprise and uses DataX [5]and Maxwell to synchronize daily full service data and daily incremental service data. User behavior data is synchronized to HDFS using Flume, and full and incremental service data is synchronized to HDFS using Maxwell+Flume and DataX, respectively. A Kafka cluster[6] was configured as a buffer between Flume at the collection layer and the synchronization layer to prevent the server from going down due to excessive data volume. An ETL interceptor[9] was configured on the collection tier Flume to perform simple data cleansing and filter illegal data to prevent Hive from parsing subsequent sequences. To improve data reliability and computation speed, HDFS distributed file[8] system is used as the storage medium. Spark[7] is used to replace the Hive computation engine to maximize the computation speed of query statements Using the workflow scheduling tool, DolphinScheduler[10], the execution process of all data collection and synchronization scripts in the system is set up as a work node to form a corresponding workflow. After setting the time, the automatic workflow can be started at the specified time, which greatly reduces the work efficiency. Using Superset as a visualization tool , various indicator charts can be drawn, such as order statistics by province and user retention rate. Finally, through module testing and overall system testing, some parameters of the system were optimized to verify the feasibility and stability of the platform.

After concluding the introductory paper, the rest of the paper is organized as follows: section II designs the big data analytics system, where section II(A) designs the data collection module, section II(B) the design of the data warehouse, and section II(C) the module design. Section III implements the e-commerce big data analytics system, where section III(A) data acquisition module implementation, section III(B) data warehouse environment implementation, section III(C) data warehouse modeling implementation, section III(D) Implementation of modules. Implementation of data quality module, and section IV system testing and data analytics results,. Where Section IV(A) uses DolphinScheduler tool to set up scheduled tasks, Section IV(B) demonstrates using data analysis visualization charts and finally Section V presents conclusions and future work directions.

2. Design of big data analysis system

2.1 Design of data acquisition module

The data acquisition module is divided into two parts, which respectively synchronizes the user behavior data and the business data in the database obtained by frontend embedding on the Web or App side to the Hadoop database. The big data components used in the synchronization methods are also different because of the different synchronization requirements of the two parts of data.

The process of transferring user behavior data from the log server to HDFS is directly synchronized to HDFS using the data channel Flume framework. Because log servers are deployed on different nodes, deploying Flume on a single node will cause cross-node transmission of data, which is very low transmission efficiency. In this case, Flume channels need to be deployed on each node of the log server. The multi-node and multi-channel method is adopted to avoid the problem of cross-node data transmission and improve the transmission speed. Another key problem is that the amount of data in general e-commerce enterprises is very large, and it is likely that the data sent directly to the hdfs will directly break down, so at this time, it is necessary to design a buffer layer kafka, and send the data to the kafka cluster through flume to achieve a peaking processing operation. The data is slowly consumed when there is free time, consumed through Flume at layer 2, and sent to the HDFS.

There are two synchronization methods for importing service data from MySQL to HDFS: daily full data synchronization and daily incremental data synchronization. The daily volume of full data is usually very large, so it also needs to be sent to the Kafka cluster first. You need to use the Maxwell tool to collect all service data tables in MySQL to a Kafka cluster.

We use DataX to import the service data from the database to HDFS. DataX is a framework for transferring data between relational databases and HDFS and is part of the Hadoop ecosystem. It translates the import and export commands into MapReduce programs to execute, and interacts with the database through JDBC. The whole process is completely automated, so as to realize efficient data transmission.

2.2 Data warehouse design

The data warehouse uses HDFS logs and service data collected to obtain service requirements through cleaning, modeling, and analysis, providing reference for enterprise leaders and related personnel. The analytics tools for the data warehouse use Hive and the underlying engine uses Spark to speed up execution.

This system the number of the warehouse is divided into five layers, respectively is: 1) the ODS (Operation Data Store) layer: the original Data layer, mainly used for the storage of raw Data collected without modify the content and it as a Data preparation in Data warehouse area. 2) the DIM (Dimension) layer: the public Dimension layer, is based on the theory of dimensional modeling structure, and to store the dimensional model of Dimension table and maintain the consistency of the Dimension information. 3) DWD (Data Warehouse Detail) layer: data detail layer, which is mainly to clean up the data of ODS layer, perform dimension degradation and dimension modeling, adopt column storage, accelerate queries, and save the minimum granularity running records of each business process. 4) the DWS (Data Warehouse Summary) layer: Summary Data layer, on the basis of DWD layer statistics for each project object behavior that day to wide and divided into some tables and to summarize the project by the day. Each row of the DWS layer's wide table is usually about the data that should be the subject object for a day. 5) ADS (Application Data services) layers: Data Application layer, mainly to provide Data for various statistical statements. The ADS layer data is imported from DWS. The ADS layer is directly related to the demand, usually a demand table, and each statistical result must have a corresponding date attribute.

2.3 Modular design

Design of the whole process task scheduling module: when conducting statistics of complex indicators in the e-commerce big data analysis system, a large number of work nodes are generated to form a workflow with execution dependency. Manually managing the workflow leads to reduced efficiency and increased costs. Therefore, Dolphin Scheduler is used for regular scheduling to realize automatic system execution and email notification for success or failure to improve work efficiency.

Data visualization module design: The system uses Superset (SP) as a data visualization tool, supporting multiple data source connections, rich chart display forms and custom dashboards. According to the ADS layer result data in the database to create tables, through the DataX tool export and use SP for visualization display, to achieve a variety of reports and statistical functions

Data quality management module design: including detection module, alarm module, visualization module and scheduling module. The detection module is responsible for detecting the data warehouse indicators, the alarm module reads the detection results and carries out the abnormality alarm, the visualization module displays the detection results, and the scheduling module manages the whole detection process.

3. Implementation of e-commerce big data analysis system

Figure 2 shows the data flow design of this system. Data in general e-commerce enterprises is divided into two parts, business data and user behavior data. These two parts of data are generally obtained in the enterprise through the front-end data burial point in Web or App.

The service data is forwarded to the Mysql database through Nginx and the user behavior data is written to local files through the logging server. To avoid data skewing, Nginx triage plays a key role. The log file data is sent to the Kafka buffer layer through Flume, and then gradually sent to HDFS by Flume to solve the problem of large data volume directly sent to HDFS may cause a crash. Business data is divided into synchronization and incremental synchronization, using DataX to synchronize all the full amount of data to HDFS, while Maxwell filters the new and changed data, synchronizes it to Kafka first, and then sends it to the HDFS of the Hadoop cluster through Numc.

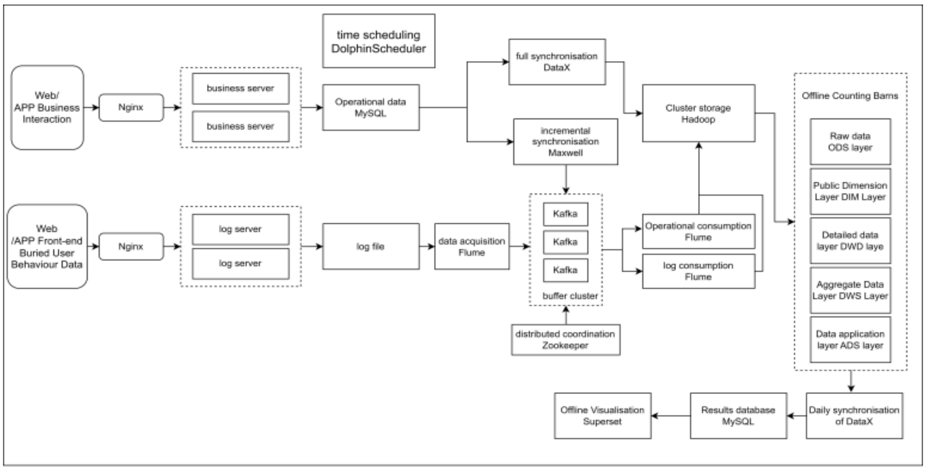

time scheduling DolphinScheduler
Web/ APP Business Interaction
Nginx
business server
business server
Operational data MySQL
full synchronisation DataX
incremental synchronisation Maxwell
Cluster storage Hadoop
Offline Counting Barns
Raw data ODS layer
Public Dimension Layer DIM Layer
Detailed data layer DWD laye
Aggregate Data Layer DWS Layer
Data application layer ADS layer
Web /APP Front-end Buried User Behaviour Data
Nginx
log server
log server
log file
data acquisition Flume
Kafka
Kafka
Kafka
buffer cluster
Operational consumption Flume
log consumption Flume
distributed coordination Zookeeper
Offline Visualisation Superset
Results database MySQL
Daily synchronisation of DataX

Fig. 2. Design of the full flow chart of system data.

The offline warehouse is divided into five layers: ODS, DIM, DWD, DWS, and ADS. The final metrics are stored in the ADS layer, which is fully synchronized to MySQL on a daily basis through DataX to improve query speed. Data is queried directly from MySQL using the Superset visualization tool. The entire warehouse schedules processes through the Dolphinscheduler tool to solve complex workflow scheduling problems in the form of task chains to reduce labor costs. The article describes the platform technical requirements, requirements analysis, overall design, as well as specific implementation work in the data collection and data warehouse data visualization module.

3.1 Data acquisition module implementation

As shown in figure 3 data acquisition module design, data is stored in HDFS so you need to build a Hadoop cluster. To realize the log data collection module, it is necessary to build 2 layers of Flume, a Zookeeper cluster and a Kafka cluster. The DataX and Maxwell frameworks need to be installed to collect service data.

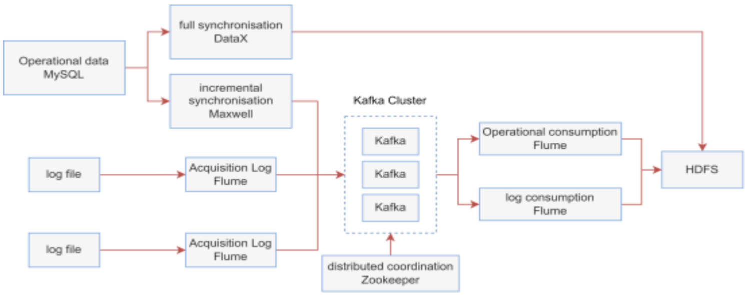


Fig. 3. Design drawing of data acquisition module.

Hadoop cluster implementation. The configuration of each node in the Hadoop cluster is similar, and it only needs to be set up on the Hadoop102 server, distributed to Hadoop103 and Hadoop104,and then make minor adjustments.

3.2 Data warehouse environment implementation

1) Hive implementation

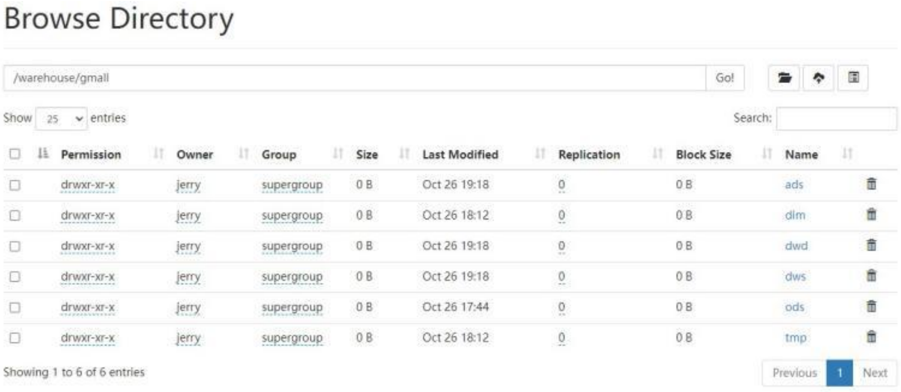
After hive is installed on node Hadoop102, install the MySQL driver package in the lib directory of Hive, create the hive-site.xml file in the $HIVE_HOME/conf directory, and configure Metastore to MySQL. After the Hive metadata is created and initialized in MySQL the current database is displayed after Hive is started, indicating that Hive deployment is successful.

2) Hive on Spark implementation

Since Hive uses Spark as the execution engine, Spark and Hive need to be installed on the same server. In this case, we installed both of them on Hadoop102. *3)* Data warehouse development and implementation This document uses DataGrip and requires JDBC connect to Hive. Therefore, you need to enable HiveServer2.

3.3 Data warehouse modeling implementation

A 5-tier data warehouse system containing ODS layer, DIM layer, DWD layer, DWS layer, and ADS layer is constructed using Hive, forming a layered structure independently of each other to minimize data dimensions and improve data analysis efficiency. In the top-down data flow, a large amount of multi-source data from ODS layer is gradually transformed into detailed report data in ADS layer. Each layer of data warehouse supports query operations between different data sources and between different dimensions and levels within data sources. The storage structure in the distributed system is shown in Figure 4, and the specific implementation of each layer of data warehouse is described in detail below.



Browse Directory

/warehouse/gmall

Show 25 entries

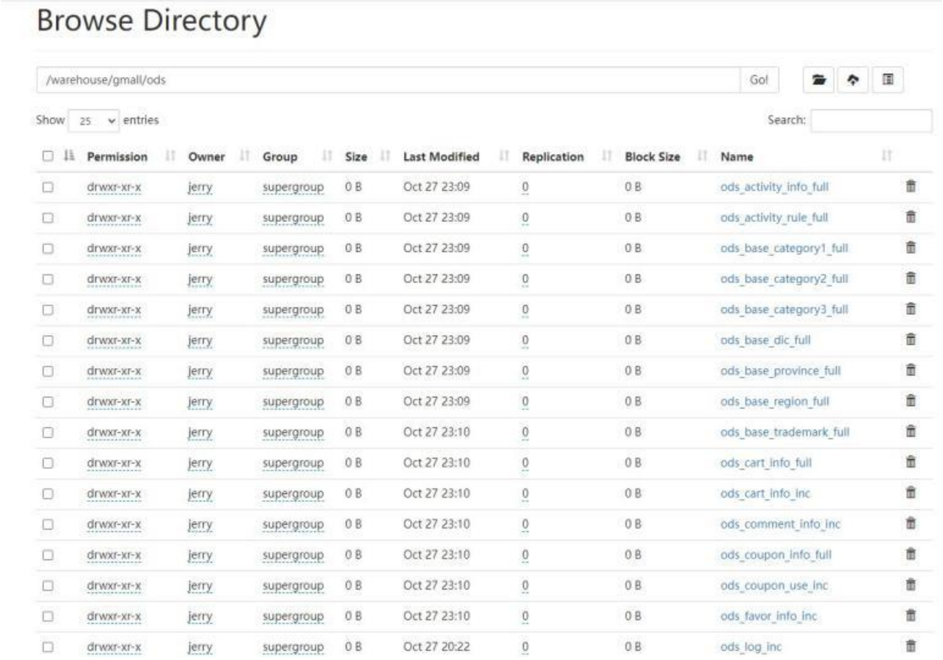
□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 26 19:18	0	0 B	ads
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 26 18:12	0	0 B	dim
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 26 19:18	0	0 B	dwd
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 26 19:18	0	0 B	dws
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 26 17:44	0	0 B	ods
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 26 18:12	0	0 B	tmp

Showing 1 to 6 of 6 entries

Fig. 4. Hive warehouse layout in the HDFS directory.

1) ODS layer data warehouse implementation

ODS is the original data storage layer, mainly storing original data. Design points include: first, the function of the ODS layer is to save raw data without processing, and the table structure design relies on the data structure synchronized from the business system. Ensure that the table structure is consistent with the data warehouse. Second, the data storage format of the ODS layer table requires simplicity, and the highly compressed gzip format is chosen to meet the storage needs of large amounts of historical data. When user behavior and service data are synchronized to HDFS, the corresponding table data will be transferred to the ODS table.The implementation of the ODS layer is shown in Figure 5.



Browse Directory

/warehouse/gmall/ods

Show 25 entries

Search:

□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name	
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_activity_info_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_activity_rule_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_base_category1_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_base_category2_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_base_category3_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_base_dic_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_base_province_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:09	0	0 B	ods_base_region_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_base_trademark_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_cart_info_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_cart_info_inc	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_comment_info_inc	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_coupon_info_full	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_coupon_use_inc	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:10	0	0 B	ods_favor_info_inc	🗑
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 20:22	0	0 B	ods_log_inc	🗑

Fig. 5. Hive warehouse layout in the HDFS directory.

2) DIM layer data warehouse implementation the DIM layer, constructed on the basis of dimensional modeling theory, is used to store the dimension tables in the dimensional model and the dimension tables in the dimensional model and to keep the dimensional information consistent. the DIM layer and the subsequent DWD layer are constructed taking into account the relationship with the Service Bus Matrix, where the Business Bus Matrix contains all the facts required for the dimensional model (the business processes) as well as the relationship between the dimensions and each of the business process's relationship to each dimension.

3) DWD layer data warehouse implementation

The data storage format of the DWD layer is the same as that of the DIM layer, and the working principle is roughly the same. Roughly speaking, a business process corresponds to a transactional fact table, and the division of data domains is ultimately the division of fact tables. Finally, the data domains divided in this paper include five data domains, namely, transactional domain, tool domain, interactive domain, traffic domain and user domain. The table name should also represent each business process. The fact table is fully synchronized according to the periodic snapshot fact table during partitioning (full as suffix), and the transactional fact table stores the most detailed operations of each business. For example, the single transaction fact table stores each single order operation. The fact table stores data according to a partition every day, and the order records of the day are stored in each partition. Such data is most appropriate with incremental synchronization (suffix inc).

In this paper, 19 fact tables are designed for the DWD layer, which indicates that the one ending in inc is a transactional fact table, and the one ending in full is a periodic snapshot fact table.

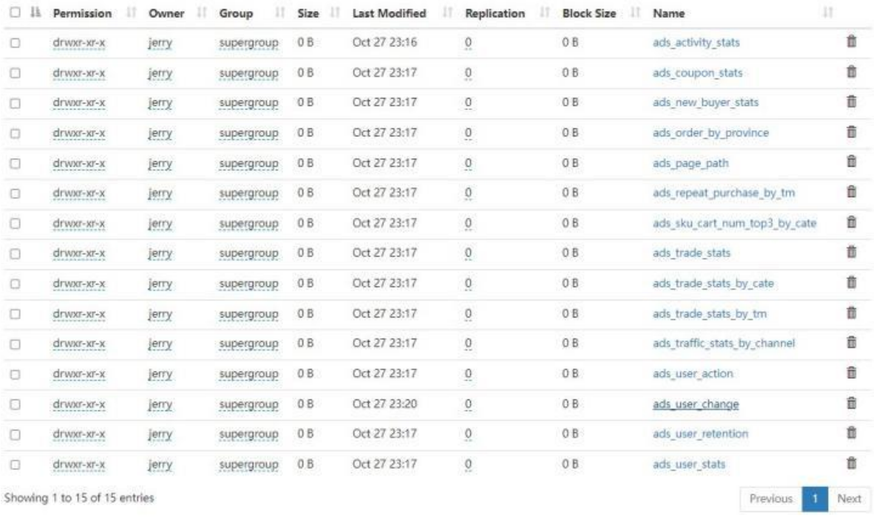
4) DWS layer data warehouse implementation

The DWS layer is designed according to the needs of the upper layer, and will be analyzed subject objects as modeling drivers and construct summary tables. On the basis of the DWD layer, the daily behavior of each topic object is counted and divided into several topic wide tables, and each topic is summarized according to H. Each row of the DWS layer's wide table is usually about the data that should be the subject object for a day

The specific work is to divide the business indicators provided by the business needs according to certain rules, and summarize them into a complete indicator system, extract all the derived indicators from the above indicator system, and extract all the derived indicators with the same business process, the same statistical cycle and the same statistical granularity separately, and make a summary table. The summary model is constructed according to the organized index system (mainly derived index).

5) ADS layer data warehouse implementation

ADS layer as the data application layer, its main function is to provide information for the final need, which stores several bins in the subsequent various applications needed statistics, a table and a need corresponding. Each table in the DWS layer is a summary table, and after the ADS layer is concluded with the DWS layer, the common calculation results are extracted in DWS and the corresponding indicators are summarized to meet the final business needs. The specific implementation of ADS layer is shown in Figure 6.



□	Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:16	0	0 B	ads_activity_stats
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_coupon_stats
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_new_buyer_stats
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_order_by_province
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_page_path
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_repeat_purchase_by_tm
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_sku_cart_num_top3_by_cate
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_trade_stats
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_trade_stats_by_cate
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_trade_stats_by_tm
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_traffic_stats_by_channel
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_user_action
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:20	0	0 B	ads_user_change
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_user_retention
□	drwxr-xr-x	jerry	supergroup	0 B	Oct 27 23:17	0	0 B	ads_user_stats

Showing 1 to 15 of 15 entries

Fig. 6. Hive warehouse layout in the HDFS directory.

3.4 Implementation of modules

Implementation of Data Visualization Module: The system's service metrics, stored in the ADS layer of the data warehouse, are visualized for display. To enhance query speed, data metrics in the ADS layer are synchronized daily to the MySQL server using DataX. The reporting application directly queries data from MySQL for reporting. HDFS to MySQL data transfer is facilitated by HDFSReader and MySQLWriter in DataX.

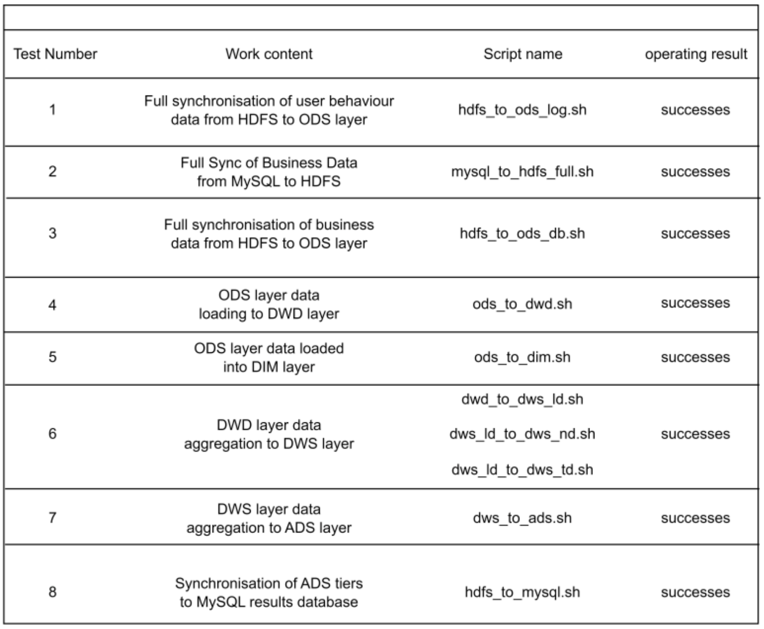
Implementation of Data Quality Module: The data quality monitoring function is implemented using Python and Shell scripts. The monitoring results are stored in a pre-built MySQL database, detailed in the Appendix. The detection module's single rule script, comprising five classes, performs checks for null values, duplicate IDs, rounding, value-by-value growth, and year-on-year data growth. The results are inserted into a MySQL table.

4. Results of system test and data analysis

In this section, we will test the system function and performance using the automatic task scheduling tool DolphinScheduler, and present data analysis visualizations using the Superset tool. Since the user behavior data and business data of e-commerce enterprises are private information of enterprises, this system simulates real enterprise data through Python script for testing.

4.1 The DolphinScheduler tool was used to set scheduled tasks

A total of 10 SHELL scripts for data import were automatically run. Figure 7 shows the test results.

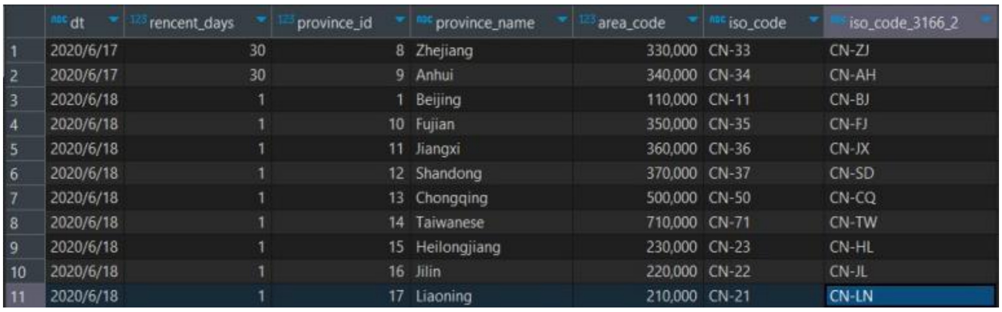


Test Number	Work content	Script name	operating result
1	Full synchronisation of user behaviour data from HDFS to ODS layer	hdfs_to_ods_log.sh	successes
2	Full Sync of Business Data from MySQL to HDFS	mysql_to_hdfs_full.sh	successes
3	Full synchronisation of business data from HDFS to ODS layer	hdfs_to_ods_db.sh	successes
4	ODS layer data loading to DWD layer	ods_to_dwd.sh	successes
5	ODS layer data loaded into DIM layer	ods_to_dim.sh	successes
6	DWD layer data aggregation to DWS layer	dwd_to_dws_ld.sh dws_ld_to_dws_nd.sh dws_ld_to_dws_td.sh	successes
7	DWS layer data aggregation to ADS layer	dws_to_ads.sh	successes
8	Synchronisation of ADS tiers to MySQL results database	hdfs_to_mysql.sh	successes

Fig. 7. Task scheduling test results.

4.2 Data analysis visual chart presentation

After the whole-process task scheduling in the first step, all user behavior data and business data are summarized and transferred to the MySQL result database, as shown in Figure 8.



	dt	rencent_days	province_id	province_name	area_code	iso_code	iso_code_3166_2
1	2020/6/17	30	8	Zhejiang	330,000	CN-33	CN-ZJ
2	2020/6/17	30	9	Anhui	340,000	CN-34	CN-AH
3	2020/6/18	1	1	Beijing	110,000	CN-11	CN-BJ
4	2020/6/18	1	10	Fujian	350,000	CN-35	CN-FJ
5	2020/6/18	1	11	Jiangxi	360,000	CN-36	CN-JX
6	2020/6/18	1	12	Shandong	370,000	CN-37	CN-SD
7	2020/6/18	1	13	Chongqing	500,000	CN-50	CN-CQ
8	2020/6/18	1	14	Taiwanese	710,000	CN-71	CN-TW
9	2020/6/18	1	15	Heilongjiang	230,000	CN-23	CN-HL
10	2020/6/18	1	16	Jilin	220,000	CN-22	CN-JL
11	2020/6/18	1	17	Liaoning	210,000	CN-21	CN-LN

Fig. 8. Task scheduling test results.

After all the ADS layer data is transferred to the MySQL result database, the Superset tool can be used to realize the visual report of each business indicator. After connecting MySQL on the Superset tool and obtaining the indicator data required by the corresponding report, the corresponding visual chart is made and designed into a digital large screen. In this paper, the user path analysis, order statistics of various provinces, order numbers of each category and crystal brand are taken as examples to make Sankei charts, maps, pie charts and bar charts respectively, as shown in Figure 9.

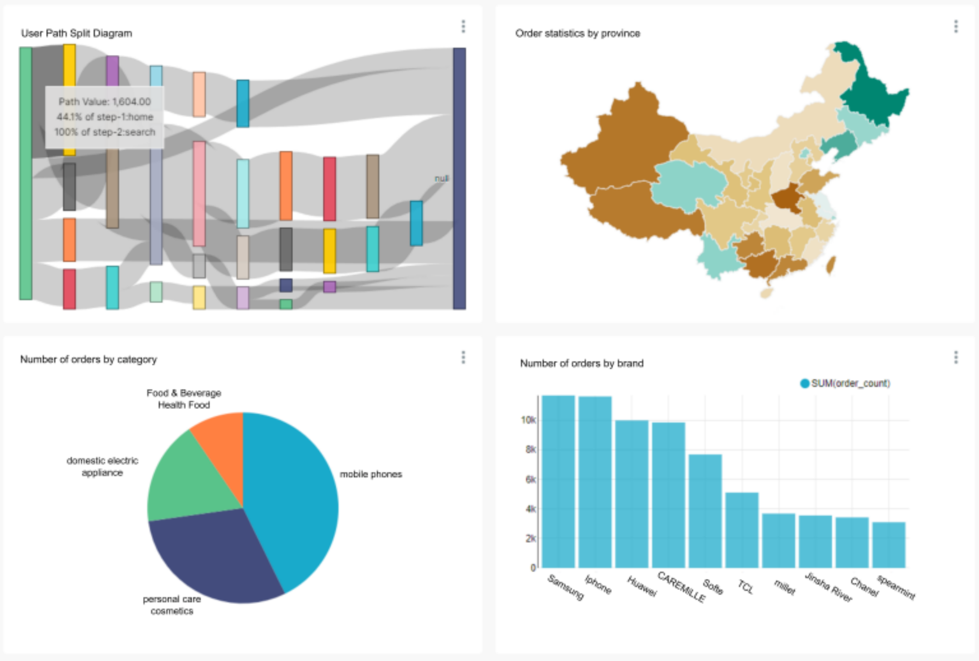


Fig. 9. Task scheduling test results.

5. Conclusion

In today’s big data era, companies are paying more and more attention to the value of data, especially in the field of e-commerce, where competition among e-commerce companies has become fierce with the increase of online shopping. In order to better analyze user and merchant data, make operational and marketing decisions, and improve competitiveness, this paper designs and develops an e-commerce data analysis system based on Hadoop big data ecosystem. The main work includes:

1) Analyze the status quo of e-commerce big data and enterprise needs, conduct in-depth research on Hadoop big data ecosystem, and design a full-process processing and analysis system.

2) Introduce the related theories and technologies used in the core components of the system, including the Hadoop framework, Yarn components, dimensional modeling theory in data warehouse technology, and briefly introduce components such as Flume, Kafka, Zookeeper, Maxwell, DataX, and so on.

3)According to the needs of the e-commerce company, clarify the development objectives and requirements of the system, complete the design of each module and propose the overall architecture. Build system modules step by step and conduct feasibility testing.

4) Complete the whole process through the task scheduler to achieve the collection, synchronization, and analysis of e-commerce data, and show the results through visual indicators. The reliability of the system is verified.

Although the proposed Hadoop-based e-commerce data analytics system provides a solution in terms of ecommerce big data, the system still needs further validation and improvement.

References

[1] Ahmed K M, El-Makky N M, Taha Y. Effective data mining: a data warehouse-backboned architecture[C]//Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research. 1998: 1.

[2] Jain V, Malviya B, Arya S. An overview of electronic commerce (e-Commerce)[J]. Journal of Contemporary Issues in Business and Government, 2021, 27(3): 665-670.

[3] Ocient Announces General Availability of Hyperscale Data Warehouse Version 19[J].Telecomworldwire,2022.

[4] Mazumder S, Dhar S. Hadoop ecosystem as enterprise big data platform: perspectives and practices[J]. International Journal of Information Technology and Management, 2018, 17(4): 334-348.

[5] Lei H, Blount M, Tait C. DataX: an approach to ubiquitous database access[C]//Proceedings WMCSA’99. Second IEEE Workshop on Mobile Computing Systems and Applications. IEEE, 1999: 70-79.

[6] Liu J C, Hsu C H, Zhang J H, et al. An event-based data processing system using Kafka container cluster on Kubernetes environment[J]. Neural Computing and Applications, 2023: 1-18.

[7] Salloum S, Dautov R, Chen X, et al. Big data analytics on Apache Spark[J]. International Journal of Data Science and Analytics, 2016, 1: 145-164.

[8] Borthakur D. The hadoop distributed file system: Architecture and design[J]. 2008.

[9] Yan X, Kuang M, Zhu J, et al. Reachability-based cooperative strategy for intercepting a highly maneuvering target using inferior missiles[J]. Aerospace Science and Technology, 2020, 106: 106057.

[10] Fang L, Yang Z, Qin S, et al. Meta-process: a noval approach for decentralized execution of process[C]//2021 International Conference on Service Science (ICSS). IEEE, 2021: 38-44.