Monitoring Data Analysis Technology of Smart Grid Based on Cloud Computing

Ziliang Qiu*, Rui Chen and Mengqiu Yan

Shenzhen Power Supply Limited Company, Shenzhen, 51800, China

*Corresponding author e-mail: 878916878@qq.com

Abstract. In the smart grid environment, the state data has the characteristics of wide area, panorama, mass and reliability. The traditional storage hardware uses disk arrays and the database management software uses the relational database system. Due to the poor system scalability, high cost and low reliability, it is difficult to adapt to requirements. Based on the above background, the purpose of this article is to study the research of cloud computing-based smart grid monitoring data analysis technology. Aiming at the problem that the power equipment condition monitoring data is getting larger and larger, and traditional storage methods cannot meet the condition monitoring big data storage problem, a cloud platform-based condition monitoring data storage system is designed. Using the distributed file system HDFS and HBase databases to store status monitoring data, this paper designs a clustering algorithm based on the density cluster structure, DBCLustering, based on the shortcomings of traditional density clustering algorithms. This algorithm first builds the core reachability relationship of the data nodes The index structure CR-Tree is then used to extract a sorted linear table about the reachability relationship of the data. Finally, the clustering results are output according to the results of the linear table. In order to solve the problem of insufficient computing power of the stand-alone version of the algorithm, a clustering algorithm for condition monitoring data based on Spark-RDD-DBClustering algorithm is proposed. This algorithm realizes the parallel application of DBClustering algorithm under Spark platform, and improves the algorithm's ability to process large-scale data.

Keywords: Cloud Computing, Smart Grid, Monitoring Data, Data Processing

1. Introduction

With the continuous advancement of smart grid research and construction, the scale of the power grid has been expanding year by year, and the frequency of power equipment failures has also increased year by year [1]. In the past ten years, large-scale power outages have occurred in domestic and foreign countries due to the failure of power equipment, causing huge losses to economic development and people's lives. The occurrence of power outages is largely due to the inferior aging of power equipment. Therefore, in order to prevent the occurrence of power equipment failure in time, it is necessary to monitor the operating status of key equipment in the smart grid [2]. By monitoring the
status of important equipment, the operating status of relevant power equipment can be obtained in a timely manner, and then the status assessment and fault early warning can be performed to repair or replace the equipment that has problems to improve the safety of the smart grid [3]. In recent years, as the smart grid continues to develop in the direction of digitization, informationization, and intelligence, the breadth, depth, and intensity of condition monitoring of power equipment have continued to increase, and the amount of status monitoring data generated by power equipment has also grown exponentially. Big data for smart grid condition monitoring is gradually forming [4]. By analyzing and researching the big data of condition monitoring, the valuable feature quantities that can reflect the status of the equipment are extracted from the large-scale, diverse formats, and low value density condition monitoring data, which will help improve the state of power equipment assessment. In order to reduce the occurrence of power grid security accidents such as power outages [5]. Therefore, how to safely and efficiently store these state monitoring data and use these massive monitoring data to accurately and timely evaluate the state of power equipment has become a new research topic [6].

Cloud computing technology has received wide attention from technical staff as a highly commercial computing model. Hadoop is a typical open source framework for distributed computing. It fully integrates the storage capabilities of a cluster of multiple nodes. And it has features such as safety and reliability [7-8]. However, more and more studies have proved that the processing efficiency of Hadoop's MapReduce computing model is difficult to satisfy [9]. In order to improve computing efficiency, a new generation of parallel computing framework Spark came into being. The distributed computing open source framework represented by Hadoop / Spark has the efficient and reliable processing of massive data, which provides the possibility for the solution of the above problems [10]. However, because of the difficulty and complexity of this problem, cloud computing technology is still in its infancy in the field of smart grid condition monitoring [11]. Whether in the field of academic research or industrial research, there is less application research to introduce cloud computing technology and big data analysis technology into power systems. There are still many theoretical and technical issues that need further research. The research on the combination of emerging Spark technologies is even rarer. Therefore, this paper introduces cloud computing technology into the field of condition monitoring to improve the storage and analysis capabilities of condition monitoring data [12].

This article aims at the continuous growth of smart grid condition monitoring data, and traditional storage and analysis mode methods are difficult to deal with the large amount of condition monitoring data. The cloud computing platform Hadoop distributed storage system and Spark memory batch processing technology are introduced into the storage of state monitoring data. Based on the analysis methods, a cloud computing-based smart grid condition monitoring data storage and analysis system is designed. Use HDFS file system and Hbase database to achieve effective storage of condition monitoring data, improve the density clustering algorithm, and use RDD programming model to achieve parallelization of the algorithm, which is used for clustering partitioning and status evaluation of the condition monitoring data. Through theoretical analysis and cluster construction, the proposed algorithm is tested experimentally and analyzed with examples, and the experimental results are analyzed and the algorithm performance is compared.

2. Method

2.1 Hadoop Cloud Platform

Hadoop is a large-scale distributed processing computing platform. It is an open source project of the Apache Software Foundation. It can not only run in a single machine environment, but also connect thousands of computers on the Internet to organize their computing capabilities for large-scale data calculations. Hadoop can effectively process massive distributed work into a computing cluster for corresponding calculations.
The focus of Hadoop processing is to deal with large-scale data problems that cannot or are difficult to complete in the current single-machine environment. For Hadoop, the amount of data of several hundred GB can only be a relatively small data size, which cannot show the computing advantages of Hadoop. Hadoop was used to calculate terabytes or even petabytes of data in the initial design stage. With such a large scale, data must be stored on different hardware resources, because the existing hardware resources cannot meet the storage of such large-scale data, and large-capacity data also causes computer addressing to waste a lot of time. HDFS (Hadoop Distributed File System) is a Hadoop-specific distributed file system. It distributes large-scale data on multiple Hadoop node computers for storage, and groups input data. The programming development under the Hadoop platform must be performed in accordance with the MapReduce programming mode.

Hadoop has two major features: HDFS and MapReduce. HDFS distributed file system is a file storage system of Hadoop. It implements distributed storage of data and provides basic support for data storage and classification. Therefore, HDFS is a guarantee for program running on the Hadoop platform. The characteristic of HDFS is that it is a complete distributed file system and has good fault tolerance performance. Generally, the number of nodes in HDFS is not fixed. It can name a separate space and use it to ensure the consistency of data. In HDFS systems, the failure of storage nodes is common and cannot be regarded as a failure. It divides a large number of files into 64MB data blocks for processing, and each data block is backed up, so the stability of the file can be guaranteed.

There are 4 core components of HBase database, namely core function modules. They are: Client Client, Coordination Service Module Zookeeper, Master Node HMaster and Region Node Region Server. HBase organizes and stores data in the form of a table. A HBase table consists of rows and columns. The row time in the HBase table is distinguished by Rowkey. The row key is also used to uniquely identify the identity of a row. The rows in the HBase table are sorted by Rowkey.

The main programming mode under the Hadoop platform is the MapReduce mode. The programs under the entire Hadoop platform are completed according to the MapReduce programming style. The programs in the MapReduce framework can easily perform calculations on a large amount of data. The MapReduce programming style is simple and easy to learn, and is suitable for running on distributed systems.

MapReduce is a model framework for parallel computing of massive data in Hadoop. Its original design is to perform parallel computing. The main components of the MapReduce programming model are Map and Reduce. The implementation of MapReduce is to divide and divide the entire task by a divide-and-conquer idea, and then perform statistical calculations on the results of each node at the end of the section to obtain the final result.

### 2.2 Spark Cloud Computing Technology

Spark cloud computing technology is an improvement over the existing cloud computing, and it has greatly improved the existing cloud computing framework in terms of speed, ease of use, and complex analysis and construction. Because Spark is a memory-based calculation method, its fast operation speed makes it the most obvious advantage.

Spark includes three main components, data storage, API, and resource management. Data storage refers to the data source of the distributed file storage system of the Hadoop platform, including HDFS, HBase, and Cassandra, which are compatible with Spark; API is an interface for operating Spark. You can implement Spark programming by calling the corresponding API, Spark programming the language is mainly Scala; Spark, as a relatively complete computing framework, can be deployed on either a server or a distributed computing framework.

The entry point for all operations and scheduling of the Spark platform is Spark Context, so the driver must operate and schedule Spark through Spark Context. It can schedule DAG Scheduler jobs and Task Scheduler tasks during initialization.

RDD, the elastic distributed data set, is the most characteristic and core concept in Spark. It is a collection of data. RDD can be regarded as a special class defined by Spark. Most of the operations of Spark are based on RDD. Of: When RDD is created, it needs to obtain the required data from the
source file and keep the data to be processed in the RDD's memory. In addition, RDDs can also be created through Spark's transformation operation process.

Another characteristic concept of Spark is shared variables, that is, Spark data can be shared, and the entire task is parallelized in this form. When Spark runs the same function on different nodes, it is necessary to back up the data used in each function so that other data can be accessed. Data sharing on the Spark platform mainly uses two forms of shared variables, one is an accumulator and the other is a broadcast variable.

All Spark operations are implemented through RDDs. There are two forms of RDDs. One is a parallel collection that can be used to receive the RDDs that have been created for parallel processing. The other is a Hadoop data set. When running the function on each record, the file system must be a storage system supported by HDFS or Hadoop. RDDs have mutual dependencies. These relationships are divided into narrow and wide dependencies.

3. Experiment
Step1: First introduce the cloud computing related technologies, including the most commonly used Hadoop, and a new type of cloud computing technology Spark. A brief comparison between Hadoop and Spark shows that Spark is more suitable for the analysis of big data for condition monitoring. Then describe the two algorithms DBSCSN and OPTICS suitable for clustering and partitioning the condition monitoring data, and design a condition monitoring data storage system based on the cloud computing platform. The design idea, overall architecture and specific module design of the storage system are carried out. Set forth. This paper focuses on the use of Hadoop's distributed file system HDFS and database Hbase to store status monitoring data, and designs corresponding data tables to lay the foundation for the next step of analysis of status monitoring data.

Step2: Following the example of transmission line insulator leakage current, a cluster analysis algorithm for the condition monitoring data of power equipment is designed. Four eigenvalues in the time domain and frequency domain of the leakage current were extracted to form a leakage current data set. A cluster structure clustering algorithm DBClustering based on density is designed. This algorithm is applied to the clustering division of leakage current data, and a matlab simulation experiment is designed to verify its feasibility.

Step3: At last, a state monitoring data analysis model based on the cloud computing platform is designed, and the specific functions of each component of the model are described; the DBClustering clustering algorithm is improved in parallel, and the RDD-DBClustering parallel algorithm based on Spark is designed. The design idea and specific process of the algorithm are explained in detail, so that the algorithm can process the condition monitoring data analysis problem in parallel on the Spark cloud computing platform, and improve the processing efficiency of the algorithm. The construction process of Hadoop and Spark cloud computing platforms, and the use of the built cloud computing platform for condition monitoring big data feature extraction and clustering test experiments. The original insulator leakage current data collected in the laboratory was used to extract features from the cloud computing platform to form a cluster test sample set. A clustering experiment was performed on the RDD-DBClustering parallel algorithm to test the processing efficiency of the parallel algorithm and parallelism.

4. Discuss

4.1 Analysis of Experimental Results
In this paper, the Hbase database of Hadoop system is used to store the leakage current data of transmission line insulators. According to the characteristics of the acquisition equipment frequency and leakage current signal, according to the design specifications of the HBase data sheet, the design leakage current data storage table is shown in Table 1.
Table 1. Insulator leakage current table

| Xingjian | Timestamp                  | Column family | Current |
|----------|----------------------------|---------------|---------|
|          | Mac1:Time1 201904090201954000 | V1            | V2      | ...     | V200     |
|          | Mac1:Time2 201904090204532000 | V1            | V2      | ...     | V200     |
|          | Mac1:Time3 201904090202232000 | V1            | V2      | ...     | V200     |

The size of the leakage current data obtained due to experimental conditions is smaller than the big data of condition monitoring generated in the actual environment. In order to analyze the big data of condition monitoring, the collected leakage current data set is now randomly copied to obtain the test big data. In the experiment, the leakage current data was clustered in a single machine and cluster environment, and the running time of the DBClustering clustering program was tested. The experimental results are shown in Figure 1.

![Figure 1. Parallel experiment results of clustering analysis of leakage current](image)

From Figure 1, it can be seen that when the amount of test data is between 10,000 to 40,000 rows, the time required to run the program on a single machine is lower than the running time of the Hadoop cluster. It takes some time, and as the data set grows, the computing advantage of the cluster becomes more and more obvious. By comparing the Hadoop cluster test time with Spark test time, we can see that compared to the existing mature Hadoop technology, Spark's parallel performance advantage is obvious.

4.2 Storage and Management of Condition Monitoring Data

Compared with Internet applications, the reliability requirements of smart grid condition monitoring data are higher. Therefore, cloud computing platforms are required to provide more reliable data storage and management methods. This topic uses the Master / Slave architecture to split the data into multiple data blocks and store them on different storage nodes. A cluster consists of a NameNode and a number of Datanodes. Name-node is a central server that is responsible for managing the namespace of the file system and clients' access to files, and Datanodes is responsible for managing the storage on the nodes. This single-node design greatly simplifies the design and implementation of the system, but it brings the problem of machine size limitation and single-point failure.

For machine scale issues, since it is a data / computation-intensive system, a single cluster designed by a single master can support tens of thousands of machines, which can meet the needs of the condition monitoring platform; NameNode exists in a single point in the HDFS cluster. The fault must be manually intervened. Currently, Hadoop does not support automatic restart or switch to other NameNodes. For the single point of failure of Datanode, it is prepared to be solved by the distributed
election algorithm Paxos. When a node fails, its task can be passed to other nodes. Paxos algorithm is a consensus algorithm based on message passing. It is considered to be the most effective among all consensus algorithms. Although Hadoop provides the distributed lock service ZooKeeper, it does not follow the Paxos protocol, but a protocol designed and optimized based on itself, so its theory has not been fully proven.

HDFS architecture supports data balancing strategies. If the remaining disk space of a Datanode drops to a certain level, the system will automatically transfer the data to other nodes. When there is a high demand for a file, a plan should be started to create a new copy of the file and balance other data in the cluster. At present these balancing strategies have not been implemented.

5. Conclusion
With the continuous advancement of smart grid construction and research, the level of grid intelligence and digitization is getting higher and higher, and the scale of monitoring data collected during the state monitoring of power equipment is also increasing. Therefore, how to store the big data of condition monitoring more effectively and deal with the analysis more efficiently has become a hot issue in the field of condition monitoring. This paper introduces Hadoop distributed file system for distributed storage and Spark cloud computing technology for parallel computing, which improves the ability to process large-scale data. In addition, this paper improves the traditional density clustering algorithm and applies it to the clustering division of condition monitoring data, which has a certain auxiliary role in improving the state assessment of power equipment.

References
[1] Chang Wang, Yongxin Zhu, Yan Ma. A Query-oriented Adaptive Indexing Technique for Smart Grid Big Data Analytics[J]. Journal of Signal Processing Systems, 2017(4):1-13.
[2] Giovanni Artale, Antonio Cataliotti, Valentina Cosentino. A new low cost power line communication solution for smart grid monitoring and management[J]. IEEE Instrumentation & Measurement Magazine, 2018, 21(2):29-33.
[3] Nielsen, Jimmy J, Ganem, Hervé, Jorguseski, Ljupco. Secure Real-Time Monitoring and Management of Smart Distribution Grid using Shared Cellular Networks[J]. IEEE Wireless Communications, 2017, 24(2):10-17.
[4] Iman Sadeghkhan, Mohamad Esmail Hamedani Golshan, Ali Mehrizi-Sani. Transient Monitoring Function-Based Fault Detection for Inverter-Interfaced Microgrids[J]. IEEE Transactions on Smart Grid, 2018, 9(3):2097-2107.
[5] Xiaohui Li, Yu-Chu Tian, Gerard Ledwich. Minimizing Multicast Routing Delay in Multiple Multicast Trees with Shared Links for Smart Grid[J]. IEEE Transactions on Smart Grid, 2018, PP(99):1-1.
[6] Yahya Al-Dhuraibi, Fawaz Paraiso, Nabil Djarallah. Elasticity in Cloud Computing: State of the Art and Research Challenges[J]. IEEE Transactions on Services Computing, 2017, PP(99):1-1.
[7] Yumin Wang, Jiangbo Li, Harry Haoxiang Wang. Cluster and cloud computing framework for scientific metrology in flow control[J]. Cluster Computing, 2019, 22(1):1-10.
[8] Zijian Cao, Jin Lin, Can Wan. Optimal Cloud Computing Resource Allocation for Demand Side Management in Smart Grid[J]. IEEE Transactions on Smart Grid, 2017, 8(4):1943-1955.
[9] Jiaxi Liu, Zhibo Wu, Jin Wu. A Weibull distribution accrual failure detector for cloud computing[J]. Plos One, 2017, 12(3):e0173666.
[10] Jian Shen, Member, IEEE. Anonymous and Traceable Group Data Sharing in Cloud Computing[J]. IEEE Transactions on Information Forensics & Security, 2018, 13(4):912-925.
[11] Shahin Vakilinia. Energy efficient temporal load aware resource allocation in cloud computing datacenters[J]. Journal of Cloud Computing, 2018, 7(1):2.
[12] Kyosuke Hashimoto, Hiroyuki Masuyama, Shoji Kasahara. Performance analysis of backup-task
scheduling with deadline time in cloud computing[J]. Journal of Industrial & Management Optimization, 2017, 11(3):867-886.