Big Data Storage Technology suitable for the Operation and Maintenance of New Generation Power Grid Dispatching Control System Operation

Yin Zhang\textsuperscript{a}, Yansheng Lang\textsuperscript{b}, Song Yang\textsuperscript{c}, Kun Zhao\textsuperscript{c}, Peng Liu\textsuperscript{d} and Feng Han\textsuperscript{e}

China Electric Power Research Institute, Beijing 100192, China

\textsuperscript{*}Corresponding author e-mail: yangsongchris@126.com, zhangyin@epri.sgcc.com.cn, \textsuperscript{b}langys@epri.sgcc.com.cn, \textsuperscript{c}zhaokun@epri.sgcc.com.cn, \textsuperscript{d}liupeng@epri.sgcc.com.cn, \textsuperscript{e}eprihf@126.com

Abstract. The new generation of power grid dispatching control system draws on the cloud computing concept for reference, adopts a new overall architecture of "physically distributed and logically uniform". It applies a large number of new technologies, enhances the relevance of various applications of the new system, and rapidly expands the application data scale, making the operation and maintenance of the power grid dispatching control system increasingly complicated. At present, the State Grid Corporation of China is carrying out research on operation and maintenance technology of dispatching control system based on historical big data mining. In the face of the collection and aggregation of a large number of operation history data and operation indexes of dispatching control system, the traditional relational database is facing great challenges. This paper discusses the data storage technology supporting big data, and through analyzing the characteristics of big data storage and the existing typical databases of NO SQL and NewSQL, puts forward a data storage technology that can meet the needs of mass operation and maintenance data storage and high-speed access of the new generation of power grid dispatching control system.

1. Introduction
With the continuous deepening of energy transformation, a new generation of power system characterized by extensive interconnection, intelligent interaction, flexibility, safety and controllability is being formed, its structure form and system characteristics have undergone major changes, and at the same time new requirements are put forward for the technical support capability of power grid regulation. The new generation of power grid dispatching control system draws on the cloud computing concept for reference, adopts a new overall architecture of “physical distribution, logical unity”, the grid monitoring function is deployed in situ, and the grid analysis and decision-making functions are deployed in a centralized manner. At the same time, a large number of new technologies such as cloud computing, artificial intelligence, and big data analysis are applied, making the new generation of control systems completely different from the traditional dispatch control systems. In the analysis and decision-making center, a large number of applications need to have the entire grid-wide operational data, simulation data and the rest of the data that the monitoring center failed to collect, to provide
complete grid operation data for the analysis decision-making center, and to meet the technical requirements for post-mortem statistical analysis of the control system. In the past, traditional relational databases were used to store data. As more and more data needs to be managed, these systems have appeared to meet the needs of space utilization, massive data storage support, and high data reliability. More and more helpless, and also showed many insurmountable problems in dealing with cloud storage technology and cloud computing technology.

In a big data environment, data storage systems should have features such as support for massive data, high performance, high reliability, scalability, and high-re-use of resources. These characteristics are not available in the traditional storage management systems implemented by relational databases. In the case that these requirements are becoming more and more difficult to meet, it is necessary to consider adopting new technical systems and more effective means for data production, management and storage.

Based on the analysis of the characteristics of big data storage, this paper analyzes the characteristics of typical NoSQL and NewSQL databases for the unified storage management and high-speed access of massive data, and discusses the data storage technology that supports the big data operation and maintenance of power grid control system.

2. Big Data Characteristics

Big data features can be summarized into 4 "V"-Volume, Variety, Value, Velocity, (1) Volume: The volume of data is huge, jumping from TB level to PB level; (2) Variety: A wide range of data types, including engineering data, web logs, pictures, location information, etc.; (3) Value: The value of data is huge, and it is far from being limited to commercial innovation, achieving precise marketing value;(4) Velocity: Fast, commonly known as the "second law", that is, the speed requirement is very high, generally the analysis results are given in the second time, and the value is lost when the time is too long[1,5]. A large amount of new calculation results and process data accumulated in the field of scientific research, which we call engineering application data, is a typical type of big data. In the development of complex systems, a large number of experiments and simulations are often carried out, so that a large amount of engineering application data can be accumulated. These massive development data are stored in various forms such as charts, data, models, etc., and need to be quickly processed to support the development of complex systems. Rapid requirements, for the entire life cycle of complex systems development. Specifically, these data include system models, knowledge (process, design, and text, etc.), algorithms, and traditional structured data-like data (e.g. simulation test data, multidisciplinary optimization data, etc.). Engineering application data such as knowledge, massively structured data, algorithms, models, etc., have typical 4V characteristics [6, 9].

When doing big data storage, you must consider the following points:

(1) Safety: Regardless of the technology used, it is necessary to ensure the security of the data. It must be considered whether there are organizations or individuals that are not allowed to access, resulting in data leakage; while isolating intruders, internal security needs to be considered to ensure that unauthorized users cannot access the data.

(2) Timeliness: The timeliness affecting data access mainly includes the access speed of physical devices and the aging time of network transmission.

(3) Reliability: To ensure the reliability of data, data must be redundant to ensure that data is not lost. The more critical the data, the more replica data will be configured, and multiple copies will also pose security issues. When a certain type of data is discarded, how to delete all copies of the data is also a problem that must be considered.

3. NoSQL database technology and features

3.1. NoSQL database

NoSQL database is also called non-relational database. It is a general term for a series of data management systems that are quite different from relational databases. The main difference is that it does not use SQL as the basic query language, because they are not The relational model is used as its
main data model, and the SQL interface is not provided. It is collectively referred to as "NoSQL"[10] (Fig.1).

3.2. NoSQL Database Features
General NoSQL database has the following characteristics:

(1) The database has good expansibility: In the process of database operation, the system can be extended by adding nodes to adapt to the further demand for performance when the data grows to a certain scale, which makes up for the lack of scalability of the relational database [11].

(2) The ability to copy and partition (slice): Most of the NoSQL databases can use the replication function to achieve data redundancy and system disaster recovery mechanism, and realize the operation of large-scale clusters.

(3) Simple interface: NoSQL databases generally do not provide their own development language packages, only provide session-level interfaces or protocols, simplifying the interface mode.

(4) A weaker transaction model: Only support weaker transactions, only need to meet the final consistency requirements.

(5) Using distributed indexes: In a distributed cluster, documents can be stored on different servers, data can be quickly located in a distributed index, and data can be stored in memory to improve data read and write speeds and improve access performance [12].

(6) No need to pre-defined mode: There is no need to pre-define the table structure, and you can dynamically add or modify attributes for data records at any time without affecting existing data. Customized data formats can still be stored in the process of software implementation, instead of creating fields for data to be stored in advance like relational databases.

3.3. Storage Model Based on NoSQL
The MongoDB database is a kind of NoSQL database for document storage, which can realize the storage and management of massive data. Its characteristics have the following characteristics: First, it is convenient to scale the storage nodes horizontally; second, the data can be automatically backed up in multi-nodes to improve the high availability; third, the document object is used as a storage unit, with fine granularity and fast access; fourth, the response to the request is fast, the speed is generally millisecond. MongoDB can support many operations similar to those in relational databases. It can also provide many functions in a relational database. The syntactic features of its execution statements are similar to the SQL syntax of relational databases. The database does not need to define the schema and
database structure in advance. It uses BSON syntax format to store data. The supported data structure is very loose. It supports complex structures such as embedded documents. It can create multi-level indexes or index each column separately. It is very flexible and convenient to use.

The unit of MongoDB data storage is a collection, which can be grouped and stored, has a unique identifier, and can contain multiple binary files in the collection. In storage, Key-value mode is used, where Key is a keyword for file data, and Value corresponds to a binary stream of a file. The basic idea is to convert the row concept in the table into a more flexible document model [13].

MongoDB is a distributed database. The automatic fragmentation mechanism is implemented in MongoDB. MongoDB can slice the data set and distribute the fragmented data blocks evenly to each node of the MongoDB cluster. The sliced MongoDB cluster is deployed. The MongoDB cluster consists of a router server, a configuration server, and a slice. The routing server is the front end of the client access cluster, and the client accesses the MongoDB cluster through the router server for data reading and writing; Two mapping relationships are recorded on the configuration server, one is the mapping relationship between the data block and the slicing server node, the other is the mapping relationship between the interval of the key value and the data block; The shard is composed of a number of shard node servers for storing actual data, and the actual data is cut into data blocks of the same size and stored on the shard node server. Each shard is a replica set. The replica set has data redundancy backup function and stores data copies on multiple shard node servers. When the client accesses the MongoDB cluster through the router server, the routing server obtains through the configuration server. The actual data information, you can find the actual server's database server where the database is located to operate the data. When the client writes data to the cluster, the routing server determines whether the written data exceeds the set data block size, if it exceeds, the data is divided into two database storages. The architecture diagram of MongoDB is as follows (Fig.2):

4. NewSQL database
Determining the performance of the SQL database is the communication overhead between the client and the server, and the transaction processing overhead on the server. If the processing on the server is
roughly classified, there are mainly four bottlenecks, and the response to these bottlenecks is decisive key. The 4 bottlenecks are as follows:

1) Logging: In order to prevent failures such as disk crashes, most relational databases perform two writes, one write to the database and one write to the log. And to prevent log information loss, you must ensure that the data is indeed written to disk. In this way, even if the database crashes due to some problems, you can restore the state before the fault according to the contents of the log.

2) Locking: Before you can manipulate records, you need to lock things in order to prevent other threads from modifying the records. This also creates a huge expense.

3) Latching: Latch is an exclusive way to access shared data structures such as locks and B-trees.

4) Buffer Management: In general, the data in the database is written to a fixed-length disk page, for which data is written to which page, or which page of data is cached in memory, it needs to be managed by the database. This is also a very expensive process.

To achieve a high-speed database system, the above four bottlenecks must be eliminated, and the above bottlenecks are not inherent in SQL databases. NoSQL is considered to be fast because it takes into account the distributed environment at the beginning of the design and distributes it through multiple nodes. However, SQL databases can also distribute processing across multiple nodes. In addition, even a NoSQL database, as long as it is designed to disk write operations, and cache management under a multi-threaded architecture, it is difficult to avoid one or more of the above bottlenecks. Therefore, a new database, NewSQL, has emerged.

NewSQL is an abbreviation for a variety of new extensible, high-performance SQL databases. It brings the advantages of relational models into the distributed architecture and takes SQL functionality into account from the outset. And the unnecessary components in the traditional relational database are simplified to improve the efficiency of execution. As a result, the NewSQL database can almost seamlessly and completely replace the relational database of the original system. In summary, NewSQL has the following advantages:

1) High scalability.
2) Support for SQL statements.
3) Support ACID consistency constraints.
4) High availability.
5) Support for Hadoop integration, etc.

4.1. VoltDB database

VoltDB is a database system that is heavily tuned for specific areas (OLTP). It is a SQL database system for large-scale transaction processing applications. Its features include the following:

1) High throughput, low latency: Through memory computing, stored procedures and serial data access implementation. VoltDB provides high throughput, low latency SQL operations. In general, it avoids disk stalls through memory calculations, avoids user stalls through stored procedures, and serializes data access within cluster nodes, avoiding the overhead of traditional database locks and buffer management. In addition, VoltDB is not pure Java development, its SQL execution engine is written in C++, so it is not affected by the GC pause.

2) Scalability: The multi-partition design of VoltDB allows data to be spread across partitions, and each partition can provide concurrent access, improving performance and unlocking. So in theory, the horizontal expansion of VoltDB can make the performance linearly improved.

3) High availability: Synchronous multi-master replication. VoltDB uses K-safety, active-active, snapshot, and WAL mechanism combination mechanisms to ensure high data availability.

4) Persistence: Innovative Technology combination of Database Snapshot and Command Log. Although the high availability of VoltDB can reduce the probability of crash, but the failure will occasionally occur, and the DBA sometimes has to stop maintenance on a regular basis. Therefore, VoltDB provides high-performance snapshots and command logs to support various persistence requirements. For logs, VoltDB supports both synchronous and asynchronous, as well as configuration such as time interval to flush to disk.
4.2. VoltDB architecture

In the traditional database, there is a database server dedicated to data storage, and the application is queried by SQL (Fig.3). For operations that need to be repeated, you can use some means to call the stored procedure written on the server in advance to reduce the traffic to a certain extent. The implementation of stored procedures is different in various RDBMS, some are written in C language and loaded into the server, and some are extended to write stored procedures.

The VoltDB database is a distributed, extensible, internally shared-nothing in-memory database. Guarantee transaction consistency and performance through parallel single threading, All transactions are implemented as java stored procedures All stored procedures (transactions) are globally ordered, while avoiding the locks, latches, and resource management overhead of traditional databases. Therefore, it is guaranteed that each transaction will continue to execute the next transaction after parallel execution on all partitions, and the transaction will not execute in disorder, stored procedures internally support grouping, multi-join, aggregation, functions, etc. That is to say, although VoltDB is a SQL database, it cannot execute SQL from the client. The process of accessing the database itself is stored as a stored procedure in the data server, and the client uses a method similar to remote calling (Fig.4).

VoltDB is a high-performance cluster open source SQLRDBMS (Database Management System) with optimized throughput. Compared with NoSQL, VoltDB not only achieves the good scalability of NoSQL, high-throughput data processing, but also does not give up the original traditional relational database, transaction support-ACID.
5. Data storage technology suitable for Big Data operation and maintenance of power grid dispatching control system

This article tested the performance of VoltDB, mongoDB, and redis+ mongoDB. Redis is a high performance memory key-value database. Can be used with databases such as mongoDB to improve the performance of databases such as mongoDB as a cache. The statistical TPS performance refers to the number of transactions executed per second, including three read operations, two of which are reading of the lightweight table, one for the large table view, and one for the primary key concurrently. The test server is configured as: Processor: 2*E5 2680V3; RAM: 64GB DDR4 2133; Network: 1000Mbps.

| Cluster configuration | VoltDB | MongoDB | Redis+MongoDB |
|-----------------------|--------|---------|---------------|
| 1 node                | 22000  | 5940    | 25600         |
| 2 nodes               | 27800  | 8400    | 32500         |
| 3 nodes               | 31600  | 11450   | 38900         |

After analysis, VoltDB is not necessarily higher in performance than redis memory Key-value database. But VoltDB's SQL and transaction capabilities are better than MongoDB, and Key-value databases tend to have much less performance if they need to apply control over transactions. Redis+MongoDB also has the following drawbacks:

1. Read and write logic is complicated.
2. There is data consistency problem with mongoDB, reliability is difficult to guarantee.
3. High cost of database maintenance.

![Figure 5. New generation control system big data operation and maintenance system structure.](attachment:figure5.png)
The SQL processing capability of VoltDB provides a simple and convenient access interface and high access speed for the classification, prediction, association, and clustering and data visualization of massive operational and maintenance data in a new generation of regulatory systems.

Therefore, we can think that the distributed VoltDB database is a very powerful memory relational database, which is suitable for regulating system operation and maintenance big data analysis and other services.

6. Conclusion
Storage technology is the key to the development of current big data processing technology, and its storage efficiency is directly related to the processing power of the entire big data processing platform. This paper analyzes the characteristics of typical NoSQL and NewSQL databases by analyzing the data characteristics of big data. By comparison, when faced with the massive operation and maintenance data storage and high-speed access of the dispatch control system, the application scope and prospect of the NewSQL system represented by voltDB will be more extensive.

Acknowledgements
This work was supported by Science and Technology Project of SGCC (DZ71-18-010), named Key Technologies and application of Dispatching Control System Testing and Integrated Operation and Maintenance.

References
[1] EPRI. The whys, whats, and hows of managing data as an asset [R]. USA:EPRI, 2014.
[2] D.X. Zhang, X. Miao, L.P. Liu, Y. Zhang and K.Y. Liu. Research on Development Strategy for Smart Grid Big Data, in: Proceedings of the CSEE. 2015 (01).
[3] S.X. Zhang, B.Z. Zhao, F.Y. Wang. Shot-term Power Load Forecasting Based on Big Data, in: Proceedings of the CSEE. 2015 (01).
[4] G.B. Li, H.Luo. Research on Intelligent Data Analysis Technology under Big Data, in: Science & Technology Information. 2013 (30).
[5] B.L. Sun. "Big Data" technology and its application in the power industry, in: Electric Age. 2013 (08).
[6] H.Y. Yao. Big data and cloud computing, in: Information Technology & Standardization. 2013(05).
[7] X.F. Meng, X. Ci. Big Data Management: Concepts, Techniques and Challenges, in Journal of Computer Research and Development. 2013 (01).
[8] G.J. Li, X.Q. Cheng. Research Status and Scientific Thinking of Big Data, in: Bulletin of Chinese Academy of Sciences. 2012 (06).
[9] W. Rao, J.Y. Ding, Q.K. Lu. Cloud Computing Platform Construction for Smart Grid, in: East China Electric Power. 2011 (09).
[10] W.H. Du. On MongoDB cloud-based data management technology research and application, in: Network Security Technology & Application. 2014 (08) 89 - 90.
[11] R.X. Ni. A Method of Heterogeneous Data Query Based on JSON, in: Radio Communications Technology, 2013. 39 (1) 73 - 76.
[12] M. Wu, L.S. Ni. Electronic Evidence Acquisition from MongoDB, in: Chinese Journal of Forensic Sciences. 2011 (03)54 - 55.
[13] Banker, k. MongoDB in Action [M]. Beijing: Post & Telecom Press. 2012.