PAPER • OPEN ACCESS

Design and Implementation of Power Grid Graph Data Management Platform Based on Distributed Storage

To cite this article: Hongbin Qiu *et al* 2019 *IOP Conf. Ser.: Earth Environ. Sci.* **234** 012026

Design and Implementation of Power Grid Graph Data Management Platform Based on Distributed Storage

Hongbin Qiu [1,a]*, Aihua Zhou[1,b], Bin Hu[1,c], Bo Chai[1,d], Yan Song[2,e] and Rui Chen[2,f]

[1]Global Energy Interconnection Research Institute Co. Ltd, Beijing, China, 102209 Artificial Intelligence on Electric Power System State Grid Corporation Joint Laboratory (GEIRI)
[2]Electric Power Research Institute of State Grid Shanghai Municipal Electric Power Company, Shanghai, China
[a]qiuhongbin@geiri.sgcc.com.cn, [b]zhouaihua@geiri.sgcc.com.cn, [c]hubin@geiri.sgcc.com.cn, [d]chaibo@geiri.sgcc.com.cn, [e]songyan@sh.sgcc.com.cn, [f]chenr@sh.sgcc.com.cn

Abstract. With the construction of smart grids, the number of power device has increased rapidly, and various types of services in power grid management have been interconnected and refined. This has made how to store and effectively manage the massive data generated in power grid information management has become a bottleneck. This paper designs and implements a power grid graph data management platform based on distributed database HBase. Experiments on power grid topology data show that the graph data management platform proposed in this paper only takes nearly 80% of the time based on the graph database Neo4j when the scale is large enough, which can realize the storage and analysis of large-scale power grid topology data, and meet the needs of power grid informationization.

1. Introduction

In recent years, with the rapid construction and in-depth development of smart grids, new energy sources, distributed power sources, electric vehicles, and energy storage devices have emerged in large numbers, and user’s load of power terminals has shown a new trend of rapid growth, large changes, and diversification[1]. The power grid architecture and operating methods are becoming more and more complex. At the same time, the level of power grid management has been continuously improved, and the main business of power grid marketing, power distribution, dispatching, etc. has been gradually penetrated and refined, which has made traditional power information systems face new challenges[2].

In the power information system, the power grid topology describes the physical layout structure of the power grid device nodes and the transmission lines connected to them, including the physical relationship of the power grid, customer relationships, and asset relationships forming a large network topology, and on this basis, massive power grid operation data, status monitoring data and smart meter data are generated[3]. These power grid topology data become the basis for business management of marketing, power distribution, dispatching, etc. in the entire power grid operation, and also the basis of the entire power information system[4]. The efficient analysis of these data can effectively support the integrated application of power grid marketing and distribution[5].

However, with the rapid increase of power grid device and the development of refined management, the traditional power grid topology analysis has been unable to meet the needs, which becomes an important issue that needs to be solved in power informationization[7]. Since the topology of the power grid is represented as a graph structure, in recent years, power grid topology analysis based on graph data has become the mainstream, and power grid topology data is also called power grid graph data[8].

Most power grid graph data uses existing mature graph databases[9,10], such as Neo4j[11]. Since most of these databases only use graph data to represent logical relationships, there is no use of distributed storage or insufficient support for this, and massive power grid topology analysis cannot provide sufficient performance.

This paper designs and implements a power grid graph data management platform, which designs the logical structure of power grid topology data through graph data, and realizes the storage structure of power grid topology data based on distributed NoSQL database HBase[12]. Finally, the power grid topology analysis algorithm is implemented on this platform. Experiments on real data of electric power show that the NoSQL-based power grid graph data management platform proposed in this paper realizes the storage and analysis of large-scale power grid topology data to meet the actual needs of power grid informationization.

2. Design of Power Grid Graph Data Management Platform

This paper designs the power grid graph data management platform architecture as shown in the figure. The platform stores and manages the power grid topology data in the form of graph data, and supports high-performance power grid topology correlation analysis, such as power grid topology search, power grid device connectivity analysis, power grid power outrage range analysis and so on. The entire power grid graph data management platform mainly includes the following layers:

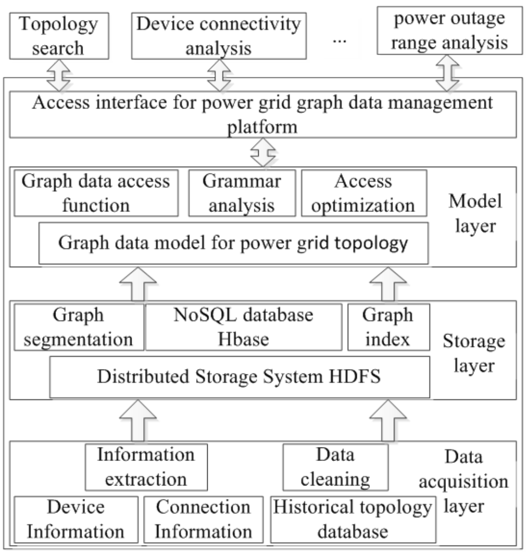

Figure 1. Architecture of power grid graph data management platform

(1) Data acquisition layer. The data acquisition layer mainly collects network topology data information, which includes not only physical device information such as busbars, substations, power stations, transformers, etc., but also physical connection information between these physical devices. At the same time, in order to support the power grid service analysis more comprehensively and effectively, it is also necessary to obtain device history information from the historical topology database, such as device model, device fault, device operation record, and so on.

(2) Storage layer. The storage layer is responsible for storing the graph data. In order to support the massive graph data, the platform is stored by the distributed NoSQL database Hbase. Hbase uses the distributed file system storage provided by HDFS to achieve reliable and massive data storage. At the

same time, the platform also achieves fast and efficient data access through graph segmentation and graph indexing.

(3) Model layer. The model layer mainly constructs the logic model of the graph data for the power grid data, and formats the power grid data according to the model. In addition, in order to support the access of the power grid topology analysis service to the graph data, the model layer also provides graph data access functions. At the same time, for more complex graph data access, such as various queries, updates, etc. of the graph data, the model layer also provides syntax parsing and optimizes the operations of these accesses.

In addition to the above three layers, the entire power grid graph data management platform also provides access to the power grid graph data management platform, various power grid topology business analysis. Power grid topology search, power grid device connectivity analysis, power grid power outage range analysis, etc. can access the power grid graph data stored in the platform by this interface.

3. Graph Model of Power Grid Topology

In order to apply the graph data to the power grid topology analysis, this paper abstracts the power grid topology into the graph structure as shown.

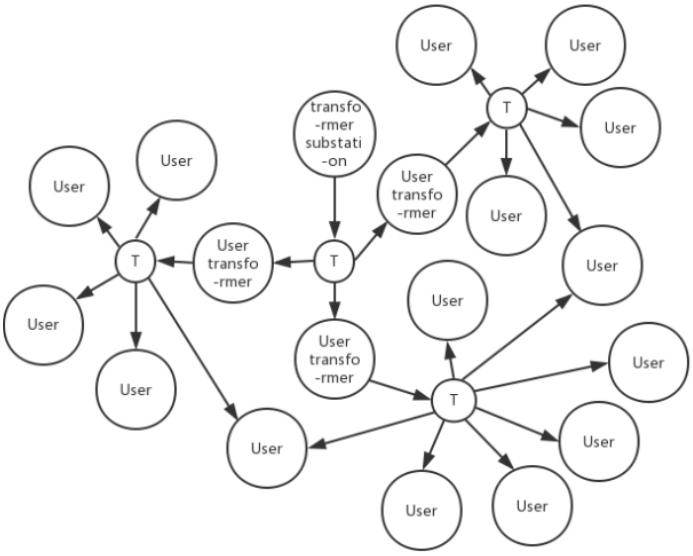

Figure 2. data model of power grid topology

As shown in Figure 2, the graph data model for the power grid topology is defined as an directed graph G = (V, E), where V is the vertex set, $E = V \times V$ is the edge set. The vertex set V contains two types of vertex, which are device type nodes and terminals type nodes. The device type nodes represent various power devices, such as BUSBAR, TRANSFORMER, Power Switch, PROTECT, TOWER, etc. Terminals type nodes represent abstractions for device connections. Edge set E contains multiple types of edges, such as HIGH_VOLTAGE_LINE, LOW_VOLTAGE_LINE, and so on.

In the graph data model, the vertices and edges also contain corresponding type attributes to indicate their characteristics. For example, the switch device has the device unique number as the main attribute, and also has general attributes such as switch status and voltage level. Edges usually have the attributes like voltage levels. In addition, in order to determine the true specific location of each device in the power grid, the general attributes of the node also include geographic location information, as well as real-time weather information based on geographic location. The attribute information on the vertices and edges enables the graph data model to fully express the complex structure and semantics of the power grid topology, while also enabling the graph database to support queries based on grid topology data.

4. Storage Design For Graph Data of Power Grid

The power grid graph data is stored in NoSQL database. The specific implementation uses Hbase database storage in the Hadoop platform. Hbase is a distributed NoSQL database based on Key Value for column families, which can be used to query the primary key in milliseconds. According to the graph data model for the power grid topology, combined with the common power grid topology analysis service, this paper uses the graph segmentation method to realize the multi-table storage of the power grid topology graph data. The partitioned power grid topology graph data storage mainly includes the following tables.

(1) Device vertex table

ROWKEY	Family	Column	Description
V_id	Cf1: Devinfo	V_id	Device number
		LID	Device type number
		SBZLX	Device properties
		SHAPE	Geographic location
		DYDJ	Voltage level
	Cf2: Change	Conn	Device connection terminal
		SS	switch status
		TS	Timestamp

The device vertex table mainly stores the basic information of the device. Considering that there are many access modes based on the unique number of the device, the device number name v_id is used as the Rowkey. The table is divided into two columns of Equipinfo and Change, and the column family Equipinfo stores information such as location, voltage level, and other information with nearly no change. Due to the modeless extensibility of Hbase, you can add it directly when you need to add any attribute fields later. This information does not change often and often read at the same time. Putting it under the same column family can greatly reduce the read and write IO latency. The column family Change stores information such as the terminal connected to the device, the switch status, and so on. This value is changed using the Hbase Increment function whenever data is inserted into the device vertex table. In this way, the primary key can be directly queried, which avoids the defect that the Hbase database needs to be extended in the full table scan in the statistical query. TimeStamp is listed to identify different versions of a row of data when update the new data.

(2) Device relationship table

ROWKEY	Family	Column	Description
MD5<v_id>-MD5<connection>	Cf:relation	V_id	Device vertex
		connection	terminal vertice

The device relationship table is used to store the connection relationship between the device and the terminals. When storing, it needs to accommodate hundreds of millions or even billions of connection relationships. The MD5 value of the device ID and the MD5 code of the terminal ID value are used to establish a composite ROWKEY, which can make sure the length of the ROWKEY value keep same, and the table creates a column cluster for storing device vertices and terminal vertices. When querying the device relationship, you only need to query the composite ROWKEY, and obtain the connection relationship of the device by matching the MD5 value of the terminal. The vertices of two or more devices connected to the same terminal are physically connected.

5. Performace Evaluation

Based on the previous design, this paper implements the power grid data management platform and conducted the related experiments. The experiment is based on three Huawei RH5885 rack servers (CPU:1.9GHz; memory: 1TB; operating system: CentOS 6.5, Hbase:hbase-2.0.0). The data used in the experiments is the real power grid topology data of a provincial power company. The entire data size is the graph data with 500 million vertices and 1.2 billion directed edges. At the same time, in order to compare with the graph database, this paper also carried out related experiments based on the Neo4j graph database.

Figure 3 shows the results of experiments which are imported data on the power grid graph data management platform implemented in this paper. In the experiment, the power grid topology with 1 million, 2 million, 5 million vertices (and associated edges) was imported with 10, 20, 50, 100 threads respectively. Figure 3 shows the average import rate under the corresponding conditions. It can be seen that the number of threads increases, or the scale of the graph becomes larger, the import speed can be increased, and the scalability is better. This is mainly because the power grid graph data management platform is based on Hbase and HDFS, which are both scalable and have good support for concurrent access. This enables the graph data management platform to make full use of multi-threading technology for high-performance data processing[13,14,15]

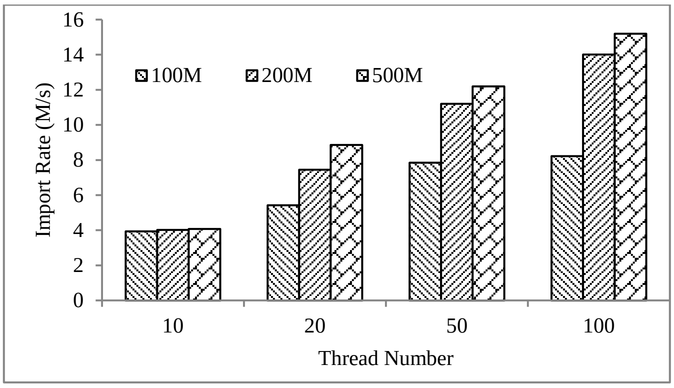

Figure 3. Data Import Performance Test

Figure 4 shows the results of the power grid topology search experiment. It gives a comparison of the time spent on the topological search of different vertex numbers on the graph database (Neo4j) and the graph data management platform (Hbase) implemented in this paper. It can be seen that, the platform proposed in this paper takes less time and has better scalability. This is mainly because Hbase itself used by the graph data management platform has a better concurrency support for massive data management design.

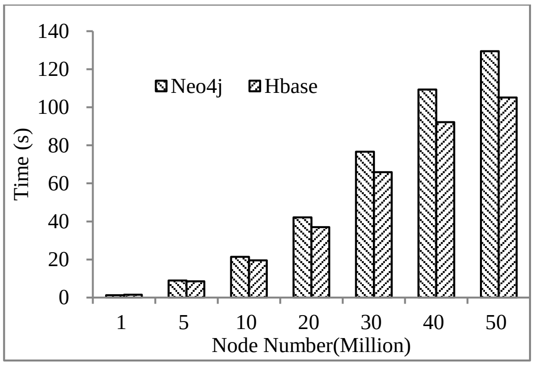

Figure 4. Node Topology Search Time

Figure 5 shows the time comparison of the power outage range analysis experiments on the graph database (Neo4j) and the graph data management platform (Hbase) implemented in this paper. It can be seen that, during the process of analysis, with the increase of concurrent users, the time spent gradually increases. But the performance of the data management platform (Hbase) proposed in this paper is significantly better. At the same time, when the number of users reaches 2000 and 3000, the increase of time-consuming is more obvious.

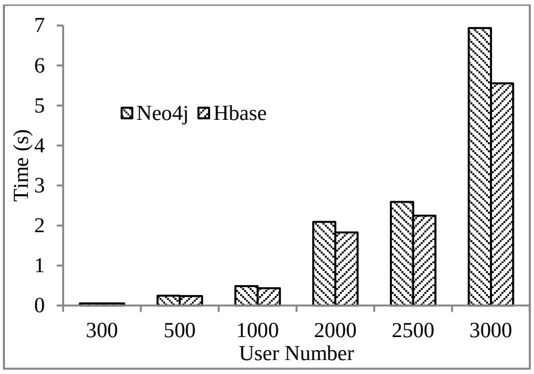

Figure 5 Analysis of power outage range

6. Conclusion

The construction of smart grid has led to a rapid increase in power grid device, and related management services such as marketing, power distribution, and dispatching have been gradually completed and refined. How to effectively store and manage the rapidly growing power grid topology data, and support the marketing, power distribution, scheduling and other services by analyzing the power grid topology data has become an important development direction of the smart grid. This paper designs and implements a power grid graph data management platform, which organizes power grid topology data in the form of graph data, and stores the power grid topology data based on the distributed NoSQL database HBase provided by Hadoop. Experiments on power grid topology data show that the graph data management platform proposed in this paper only takes nearly 80% of the time based on the graph database Neo4j when the scale is large enough, which can realize the storage and analysis of large-scale power grid topology data, and meet the needs of power grid informationization.

Future work will further support the power grid topology analysis of other services based on the power grid topology realized in this paper, such as multi-power loop inspection, station detection, low-voltage station analysis, and power supply analysis. At the same time, the entire power grid topology graph will be segmented to expand the power grid topology parallel analysis on each of the divided subgraphs for the further improvement of performance.

7. Acknowledgments

This work is supported State Grid Company Research Project under grant 5455HJ170005.

8. References

[1] NING Bai-feng.Design and Application of Unifi ed Power Grid Resource Cloud Service Platform. Electric Power Information and Communication Technology. 2016, 14(2):143-146.

[2] CHEN Hong-bo, LIU Qing, YIN Ying-qi. Research on information resource virtualization of power enterprise using cloud computing[J]. Electric Power Information and Communication Technology, 2015, 13(11): 72-76.

[3] ZHAO Yuehui, PENG Yuying, GU Quan, CHEN Genjun. Real-time topology tracing for active power distribution network with distributed energy. Power Systems and big data. 2018, 21(6):44-50.

[4] ZHOU Buxiang, LIU Xinyu. Network graph-based power distribution network topology analysis and its application. Automaton of Electric Power Systems, 2003, 27(16): 67-70.

[5] Hu mu, Li yong, Kong zhen, Liu shijin. Data topology and its application in processing and analysis of power grid data. Automation of Electric Power Systems. 2013, 37(3):83-86.

[6] LONG Qifeng, CHEN Gang, DING Xiaoqun, et al. New method of power network topology analysis based on object-oriented technology[J]. Proceedings of the CSU-EPSA, 2005, 17(1): 73-77.

[7] Qiu Hongbin, Zhou Aihua, Hu Bin. Chai Bo, Pan Sen, Yang Pei. TAnalyzer: A Graph Database Based Topology Analysis Tool for Power Grid. proceedings of The 2nd International Conference on Computer Science and Application Engineering, 2018.

[8] LANG Yansheng, LI Jing, LUO Yadi, WU Lingyun, LI Qiang, ZHAO Jun, WANG Shunjiang. Large power grid topology analysis based on graph partitioning. Power System Protection and Control, 2017, 45(23):108-115.

[9] W. Dharmawan, R. Sarno. Book Recommendation Using Neo4j Graph Database in BibTeX Book Metadata. In Proceeding of the 3rd International Conference on Science in Information Technology (ICSITech), 2017, 47-52.

[10] E. Mahmoud, R. Mohamed, A. Mostafa, and -B.M.S Abdel. Large –Scale Ontology Storage and Query Using Graph Database – Oriented Approach: The Case of Freebase. In Proceeding of IEEE Seventh International Conference on Intelligent Computing and Information System (ICICIS'15), 2015.

[11] The Neo4j Graph Platform, https://neo4j.com/

[12] Apache HBase, http://hbase.apache.org/

[13] Basel A. Mahafzah. Performance evaluation of parallel multithreaded A* heuristic search algorithm. Journal of Information Science, 40(3), pp. 363-375, June 2014.

[14] Basel A. Mahafzah.Parallel multithreaded IDA* heuristic search: Algorithm design and performance evaluation. International Journal of Parallel, Emergent and Distributed Systems, 26(1), pp. 61-82, February 2011.

[15] Basel A. Mahafzah. Performance assessment of multithreaded quicksort algorithm on simultaneous multithreaded architecture. Journal of Supercomputing, 66(1), pp. 339-363, October 2013.

7