Earthquake observation data grading and storage research

*Kun Zhu, Hui Liu, Yu Ji, Zehao Li

Changchun Institute of Technology, Jilin Province Key Laboratory for Earthquake Resistance & Hazard Mitigation of Civil Engineering, Jilin Province Research Center of Construction Technology & Engineering For Housing in Cold Regions, Changchun, Jilin, 130012, China

* Corresponding author’s e-mail: tm_zk@ccit.edu.cn

Abstract. With the constant development of contemporary science and technology and increasing popularization of the big data technology, the concept of “database” has been integrated into people’s daily work and life. The database system also plays an indispensable role in earthquake early reports. This paper briefly analyzes introducing the mature big data technology to earthquake observation and seeks graded storage of databases to ensure a faster and more accurate retrieval of data as well as timely reports of earthquakes.

1. Introduction

China has been a country suffering serious losses from earthquakes. Immediate pre-earthquake emergency reports can largely reduce losses resulted from earthquakes. The national management and requirements of earthquake forecasting, pre-warning, prevention and so on have been constantly improving. Implementation of pre-warning has a close bearing on basic data [1]. During the whole earthquake emergency pre-warning, data play a tremendous role, having been recognized as a focus of earthquake pre-warning. An issue of great concern for database construction is about how to construct precise databases to enable precise applications, timeliness and validity of data information, and to provide bases for commanding and decision-making [2].

The output of increasing earthquake observation data has put forward new requirements of data storage, management, etc. More and more information systems have been developed, leading to a soaring number of data, which has been found with one after another contradiction with the system’s limited storage. In order to address the contradiction and improve the storage management efficiency, timely optimization and update of storage space and performance have been necessary. Currently, the commonly-used database storage means lack diversity and scalability. This has been partially responsible for the large number and high complexity of data storage. It is stressful for a single database to process mass data. Against the backdrop, use of graded storage technology for management of storage space and improvement of earthquake report efficiency has been an irresistible development trend.

2. Graded storage subsystem

As one of core systems, graded storage subsystem plays a significant role in functions, such as real-time waveform access, earthquake phase automatic collection, earthquake phase auto-correlated positioning, automatic computing of the amplitude and earthquake magnitude, FDSN Web service interface, etc. This paper mainly analyzes the graded storage subsystem processing technology, that is, to construct the graded storage subsystem based on the Ignite distributed high-availability cluster, HBase distributed high-availability cluster, Hadoop distributed high-availability cluster, and MariaDB cluster.
Distributed storage is to store data, according to data importance, access frequency, retention time, capacity, performance and so on, in storage systems of different performances in different storage modes. The platform data graded storage is based on the locality of data access. Through graded storage management, automatic migration of data between different storage devices can be realized via graded storage management [3]. Through automatic migration and online data to the first storage layer, a large storage space can be unleased to obtain a high cost performance ratio for near-line data and offline data, thus improving the storage performance. In order to ensure the data storage and retrieval efficiency, and realize graded storage of waveform data, namely online data (<one-hour waveform data), near-line data (<30-day waveform data), offline data (waveform data over one to two years), data at these three layers of storage subsystems can be automatically circulated. Meanwhile, as the number of the data size keeps on increasing, the system should possess a favorable scalability. Being not just a traditional means which storages data in a database, graded storage actually can store data at the three layers of storage for respective retrieval and storage, respectively. So advantages of graded storage is self-evident. First, graded storage has guaranteed the system stability. The single database of graded storage can independently operate file catalog and retrieve data, which can alleviate the burden of some data operations on the data server, and largely reduce the frequency of database operation, which can contribute to the system stability. Second, compared with the traditional storage system, the graded storage system excels undoubtedly in the reading and writing speed.

3. Graded storage plan

Estimated by 15,000 stations, around 300G to 400G waveform data are generated every day. In order to meet the needs of real-time computing, quick retrieval and mass data storage, three-layer storage, chiefly online, near-line and offline storage, is adopted. As shown in Figure. 1, waveform data generated around one hour are stored in the Ignite distributed cache for high-concurrency and real-time retrieval. The waveform data generated over the 30 days are stored in the HBase distributed database for secs-level retrieval. Waveform data generated over the long term are stored in the HDFS, which can realize quick data retrieval based on the catalog rules and SQLite database index rules.

![Figure. 1 Graded distributed subsystem](image)

**3.1 Ignite distributed cache**

The online data are stored in the Ignite distributed cache. The data organization within Ignite features a high-performance, integrated and distributed storage platform, which can achieve real-time handling of tasks and computing in the big data. Compared with the traditional technology that is based on disc or flash memory, Ignite distributed cache has achieved conspicuous improvements for its performance. The platform constructs the client end based on Kafka Stream. From the Kafka information exchange subsystem, the waveform data can be read on a real-time basis. The data expiration period is set to be one hour. The meta-data and data packets are simultaneously stored in the Ignite distributed cache.

(1)Configuration information
Management of configuration information is to meet needs of quick inquiry and data persistence. At the same time, the update notice mechanism should be designed, and the latest configurations are adopted if not during the downtime. The internal memory data grid of Ignite is a complete task-based distributed key-value storage, which can seek horizontal expansion on the cluster of several hundred sets of servers. At the same time, Ignite can provide the complete SQL, DDL and DML support, and can use the pure SQL rather than write codes for interaction with Ignite.

When the system is launched, Ignite can read all configuration information from the MariaDB/MySQL database and cache the data in internal memory. Various business modules quickly retrieve data from Ignite via Key-Value or SQL. To update configuration information, the configuration information should first be updated in the MariaDB/MySQL database. At the same time, the cache in Ignite is update. Configuration update notice can be announced through the Kafka. After receiving the notice of update, business models can read the latest configuration information from Ignite.

(2) Intermediate results

Intermediate results (earthquake phase collection results, amplitude and magnitude, focus information, earthquake event information, etc.) generated by various business modules will first be posted in the Kafka information exchange subsystem. The independent client ends are built based on Kafka Stream, respectively, and used for internal storage cache and data persistence of intermediate results. The intermediate results are written in the Ignite distributed cache. Setting of the expiration date can prevent a large number of invalid intermediate results from influencing the performance. Meanwhile, persistence of intermediate results in the MariaDB/MySQL database is necessary. Concerning intermediate results with a large data size can be stored by the Sharding-JDBC according to the time submeter.

The Ignite distributed database, because of the highly flexible scalability brought thereby to the system, has gradually been a main development trend of the database storage technology in the big data era to the high flexible scalability of the Ignite distributed database. Apart from automatically controlling data zoning, the Ignite distributed database allows developers to insert the self-defined (relation) function and to merge part of data for efficiency improvement. In contrast to traditional databases, Ignite distributed storage can ensure the system stability, security, reading and writing speed. The overall data reading and writing speed is amount to the polymerization speed of multiple storage discs [4]

3.2 HBase database

HBase is a distributed data storage system, whose table structure is presented in Table 1. Different from the general relation database, it is a database suitable for non-structured data storage. Besides, the HBase model is based on the column rather than the row. Relying a high scalability and throughput, HBase resorts to Key/Value storage. Therefore, the inquiry performance will not be impaired because of the robust data growth. Being a column-style database, HBase can store different columns at different service cases. Particularly when the single table contains lots of fields, the above advantage of HBase over the traditional row-style database can help distribute the loading pressure [5]. The platform constructs the client end (micro-service) based on Kafka Stream, reads the waveform data from the Kafka information exchange subsystem, and then stores the meta-data of data packets together with data packets in HBase. In the process of preparing HBase table, data TTL is designated to be 30 days (60 years by default), so those whose TTL is beyond 30 days will be automatically eliminated.

Long.MAX-Start Timestamp is used to arrange waveform data in a reverse order by time, which can accelerate the retrieval of the latest data. The TTL is set to be 30 days upon table formulation, so HBase can automatically eliminate expired data.

RowKey example: The Network Code of a continuous waveform is IU; its Station Code is ANMO, Location Code is OO, and Channel is 0BHZ. The Start Time of the data packet is 10 o’clock 24min, 30s, 256ms on October 10, 2018, which means the timestamp is 1539138270256 and its RowKey is 00IUANMO00OO0BHZ:1539138270256.
Table 1. SEED Table Structure.

| Project | Explain |
|---------|---------|
| RowKey  | Rule:  \{Stream ID\}:\{Long.MAX-StartTimestamp\} network, station, location, channel less than four digits left complement 0 |

| ColumnFamily | Column                        |
|--------------|-------------------------------|
| meta         | size                          | byte array length             |
|              | times                         | record start timestamp        |
|              | timee                         | record end timestamp          |
|              | timea                         | data arrival timestamp        |
|              | net                           | network                       |
|              | sta                           | station                       |
|              | loc                           | location                      |
|              | chan                          | channel                       |
|              | num                           | number of samples             |
|              | rate                          | sample rate factor            |
| data         | seed                          | waveform data (Format for miniSEED) |

3.3 HDFS
HDFS is a distributed file system which is designed to be compatible with commonly-used hardware. In addition to sharing characteristics of the existing distributed file systems, First, HDFS differs itself from traditional systems in the following aspects. HDFS is a system with a high fault tolerance. When the system breaks down in part of components (one or several), HDFS can still operate normally, which is suitable to machines with a low performance [6]. Second, HDFS allows data access of a high throughput, which is extremely fit for applications on large-scale datasets. The platform constructs independent client ends (micro-services) to periodically retrieve data (including meta-data and data packets) from HBase (1~n days, according to the daily waveform data size). Index files can be generated via meta-data, and after filing, data files and index files are simultaneously stored in the designated catalog of HDFS. The table design is demonstrated in Table 2.

Table 2. SEED Table Design.

| Column name | Explain                          | Other          |
|-------------|----------------------------------|----------------|
| times       | record start timestamp           | primary key    |
| timee       | record end timestamp             |                |
| timea       | data arrival timestamp           |                |
| offset      | the offset of the packet in the archive file |                |
| size        | byte array length                |                |
| net         | network                          |                |
| sta         | station                          |                |
| loc         | location                         |                |
| chan        | channel                          |                |
| num         | number of samples                |                |
| rate        | sample rate factor               |                |
Catalog design:
Referring to the catalog design rules of slarchive module in SeisComP3, the author names the waveform data file and index file. The naming rules are shown below:

Catalog naming:
```
<dir>/ Year/ NET/ STA/ CHAN./ TYPE/
```

File naming:
Index file: NET./ STA./ LOC./ CHAN./ TYPE./ DAY./ index
Data file: NET./ STA./ LOC./ CHAN./ TYPE./ DAY./ mseed

E.g.
```
<dir>/2018/HE/HNS/BHZ.D/HE.HNS.00.BHZ.D.2018.001.index (This file name indicates a file index of the waveform data from the Hebei Hongshan Seismic Station at the vertical direction on January 1, 2018.)
```
```
<dir>/2018/HE/HNS/BHZ.D/HE.HNS.00.BHZ.D.2018.004.mseed (This file name indicates a data file of the waveform data from the Hebei Hongshan Seismic Station at the vertical direction on January 4, 2018.)
```

When it comes to filing of waveform data which are generated for more than one day, the file naming resorts to the data of the first data packet.

The index file adopts SQLite database for data storage.

4. Conclusions
With the advent of the big data era, database construction will improve the earthquake monitoring and forecast level, ensuring the earthquake information publication to be timelier and more accurate. A complete database can help improve the government’s ability to cope with emergency, provide timely pre-warning, and alleviate losses caused by earthquakes. Meanwhile, through construction of the database graded storage subsystem, three-level storage, chiefly online, near-line and offline data storage, will be achieved for a more systematic management of the database as well as faster and more accurate inquiry of mass data. In the future, the database graded storage technology will find applications in more extensive scenarios.

Acknowledgments
This article is one of the phased achievements of the Jilin Province Science and Technology Development Plan funded project "Application Research of Big Data Artificial Intelligence in Earthquake Monitoring"

References
[1] Xu B. Design and research of risk pre-assessment database for earthquake disaster[J]. Wireless Internet Technology, 2021,18(6):56-57,60..
[2] Liu B. Y . Importance and improvement measures of earthquake emergency basic database construction[J]. Technology and Market, 2017,24(4):323-324.
[3] Zhang Y . & Pan H. F. Applications of intelligent graded storage system to Inner Mongolia Grid’s email system [C]. // Paper Collection of the 2015 Electricity Industry Informationalization Annual Meeting. 2015:367-370.
[4] Yang W. J. Study on distributed parallel cluster storage technology oriented towards big data [J]. Electronic Test, 2021(9):88-89.
[5] “Big Data Wall Facer” (Weibo Blog Account).(2021)Advantages and disadvantages of HBase. Retrieved from https://blog.csdn.net/weixin_42796403/article/details/112790756.
[6] Jin G. D., Bian H. Q., Chen Y. G., et al. Survey on Storage and Optimization Techniques of HDFS[J].