International Journal of Advanced Engineering and Technology Research
ISSN: 3106-8677 (Print), ISSN: 3106-8685 (Online) | Vol. 2, No. 1, 2026

Key Technologies and Applications of Geoscience Big Data

Yanbin Zhou

School of Resources and Environment, Henan Polytechnic University, Jiaozuo 454003, China

Abstract: Geoscience big data refers to massive, multi-source, heterogeneous, and spatiotemporally associated datasets generated in earth science research and engineering practice. It covers geological, geographical, remote sensing, meteorologi cal, hydrological, environmental, and resource-related data. With the development of remote sensing observation, GIS, Internet of Things monitoring, cloud computing, and artificial intelligence, geoscience data are increasing continuously in scale, type, and update frequency. Conventional stand-alone data processing methods have gradually become insufficient in terms of storage capacity, computing efficiency, spatial query, and integrated analysis. Based on a course report on geoscience big data, this paper reorganizes the concepts, technical framework, implementation process, and application scenario of geoscience big data in the form of an academic paper. First, the data sources, basic characteristics, and processing requirements of geoscience big data are analyzed. Then, a technical framework for geoscience data processing is constructed around cloud servers, Linux operating environments, Hadoop/MapReduce, MongoDB, PostgreSQL/PostGIS, and Python batch processing. On this basis, practical operations, including server connection, file management, MapReduce-based column statistics, and Java program compilation and execution, are summarized. The study indicates that geoscience big data technologies can improve the organization, storage, computation, and analysis efficiency of multi-source geoscience data and provide technical support for typical applications such as WebGIS-based disaster monitoring and early warning. However, further application still needs to address issues such as inconsistent data standards, insufficient data quality control, complex system integration, and data security protection.

Keywords: Geoscience big data, Hadoop, MapReduce, cloud server, Python, PostgreSQL, MongoDB, geoscience application.

1. Introduction

The rapid development of information technology has promoted society into a data-driven stage. Big data is no longer merely a concept denoting a large amount of data; it represents a complete technical system that includes data acquisition, storage, governance, computation, analysis, and service delivery. For earth science, geoscience data come from a wide range of sources, including traditional materials such as boreholes, geological maps, field surveys, sampling tests, and also new data types such as remote sensing images, unmanned aerial vehicle surveys, geophysical exploration data, real-time sensor monitoring, meteorological and hydrological observations, and three-dimensional modeling results. These datasets usually contain spatial locations, time-series records, attribute indices, and multi-scale representations. They are characterized by large volume, diverse formats, high dimensionality, and complex spatial and semantic relationships among different data sources.

In traditional geoscience research, data processing often relies on stand-alone software, manual data organization, and local analytical workflows. Such methods can satisfy basic requirements when the data scale is small and the data types are relatively simple. However, when dealing with wide-area, high-resolution, long-time-series, and multi-source integrated geoscience data, traditional methods are prone to problems such as limited storage capacity, low computational efficiency, delayed data updating, and poor reusability of processing workflows. Geoscience big data technologies provide new technical approaches for the efficient organization and integrated analysis of geoscience data through distributed storage, parallel computing, cloud-based management, database indexing, and automated scripting.

Previous studies have shown that geoscience big data is driving the research paradigm of earth science from empirical analysis and model inference toward data-intensive scientific discovery [1]. Related technologies can support rapid processing of massive geoscience data and serve practical problems such as natural resource evaluation, ecological and environmental monitoring, urban geological investigation, disaster monitoring and early warning, and mineral resource prediction [1-2]. Therefore, systematically summarizing geoscience big data technologies and analyzing their implementation based on server environments and programming practice have both theoretical significance and practical value.

This paper is organized around the logic of "conceptual characteristics - technical system - implementation process - application scenario - challenges and prospects". First, the connotation, data sources, and technical requirements of geoscience big data are analyzed. Second, the roles of cloud servers, Hadoop, MongoDB, PostgreSQL/PostGIS, Python, and MapReduce in geoscience big data processing are summarized. Third, based on course practice, Linux server connection, basic command-line operations, Python-MapReduce column mean calculation, and Java program compilation and execution are reorganized. Finally, WebGIS-based disaster monitoring and early warning is selected as a typical application case to discuss the application value and future development direction of geoscience big data technologies in multi-source data organization, spatial database management, model computation, and visualization.

2. Connotation and Technical Characteristics of Geoscience Big Data

(1) Definition of Geoscience Big Data

Geoscience big data refers to massive datasets generated during earth science observation, investigation, experiments, simulation, production, and management. Compared with general business data, geoscience big data has obvious spatial, temporal, and multi-scale characteristics. Spatiality means that most geoscience data are related to coordinates, regions, strata, profiles, or three-dimensional locations. Temporality means that meteorological, hydrological, environmental monitoring, and remote sensing data are often updated continuously or periodically. Multi-scale characteristics are reflected in the fact that the research object may range from the global or regional scale to mining areas, cities, watersheds, engineering sites, or individual working faces.

In terms of data structure, geoscience big data includes structured data such as borehole tables, sample test tables, monitoring records, and attribute databases; semi-structured data such as XML, JSON, log files, and metadata files; and unstructured data such as remote sensing images, geological maps, profiles, reports, videos, and images. The multi-source and heterogeneous nature of these data determines that geoscience big data processing cannot rely on a single database or a single algorithm. Instead, it requires an integrated technical system consisting of data storage, spatial indexing, parallel computing, data cleaning, and visual presentation.

(2) Data Sources and Types

According to acquisition methods, geoscience big data can be divided into observation data, survey data, experimental test data, model computation data, and management service data. Different data types vary in accuracy, scale, update frequency, and data structure, so unified coding, format conversion, coordinate correction, and quality checking are required before data processing. Table 1 summarizes typical geoscience data sources and their characteristics.

Table 1. Typical sources of geoscience big data and processing requirements

Data source	Typical data	Main characteristics	Processing requirements
Remote sensing and aerial survey	Satellite images, UAV images, DEM	Large coverage, high resolution, large data volume	Image preprocessing, classification, time-series analysis
Geological survey	Geological maps, profiles, borehole columns	Strong professional semantics and complex spatial relationships	Spatial vectorization, attribute standardization, 3D modeling
Monitoring observation	Meteorological, hydrological, environmental, and disaster monitoring data	High update frequency and clear time-series patterns	Real-time access, anomaly recognition, trend analysis
Experimental testing	Sample tests, geotechnical parameters, mineral components	Relatively structured data and clear indicator systems	Data cleaning, statistical analysis, indicator correlation
Model results	3D models, raster fields, prediction results	Multi-scale and multidimensional results with large files	Index management, version control, visualization service

(3) Basic Characteristics and Technical Requirements

Geoscience big data generally exhibits large volume, strong diversity, high processing-speed requirements, and clear differences in value density. Large volume means that remote sensing images, monitoring sequences, and three-dimensional models may reach the GB, TB, or even PB level. Strong diversity means that a single research task often involves tables, images, vectors, rasters, texts, and model files at the same time. High speed requirements mean that scenarios such as disaster monitoring, environmental early warning, and production scheduling require near-real-time processing. Differences in value density mean that only part of the information contained in a large amount of raw data directly serves scientific interpretation or engineering decision-making.

Therefore, geoscience big data processing should meet at least the following requirements. First, the storage system should be scalable and support massive datasets and multi-format file management. Second, the computing system should support parallel processing to improve batch-analysis efficiency. Third, the database system should support spatial indexing and attribute queries to meet the requirements of geoscience object retrieval and spatial analysis. Fourth, the application system should support visualization and result services so that data analysis results can be understood and used by researchers and managers.

3. Key Technology System of Geoscience Big Data

(1) Cloud Servers and Linux Operating Environment

A cloud server is a virtualized server running in a cloud computing environment. It provides computing, storage, and application operation resources to users through a network. Compared with traditional physical servers, cloud servers are flexible in deployment, scalable in resources, convenient for access, and controllable in cost. They are suitable for geoscience big data course experiments, small and medium-sized data processing platforms, and phased computing tasks. For geoscience projects that need long-term operation, higher data security, or heavy computational loads, private cloud or hybrid cloud solutions may be adopted to balance security, performance, and cost.

Linux systems are widely used in big data technical environments. Their command-line operations are convenient for remote management, script execution, and program deployment. Geoscience big data processing involves many operations, such as file copying, directory management, program execution, and log viewing. Mastering basic Linux commands can improve the controllability of data processing workflows. In course practice, Linux servers can be connected through Xshell, PuTTY, secureCRT, or terminals provided by cloud service vendors. File upload, directory management, Python program execution, and Java program compilation can then be completed through the command line.

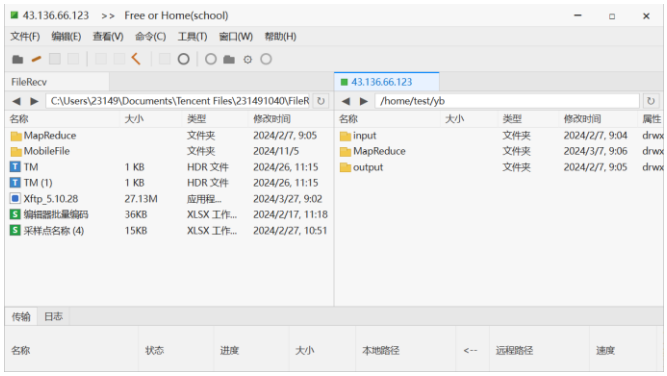

Figure 1. Example of Linux server file management interface

(2) Distributed Storage and Hadoop Technology

Hadoop is a typical distributed processing framework for big data and an important basic platform in the batch-processing technical system [3]. It mainly includes the Hadoop Distributed File System (HDFS) and the MapReduce distributed computing framework. Its technical ideas are closely related to early distributed system studies such as the Google File System, MapReduce, and Bigtable [4-6]. HDFS improves the storage capacity and fault tolerance of massive datasets by splitting large files into multiple data blocks and distributing them across different nodes. MapReduce divides computational tasks into several subtasks that can be executed in parallel. The Map stage performs local mapping and processing, while the Reduce stage aggregates intermediate results and obtains the final global result.

In geoscience data processing, Hadoop is suitable for tasks such as large-scale remote sensing image indexing, monitoring data statistics, spatial attribute batch processing, and text data cleaning. Its advantage lies in using cluster nodes to process large-scale data in parallel and reduce stand-alone computing pressure. Its limitation is that platform deployment and task scheduling are relatively complex, and it is not always the best choice for real-time interactive analysis or small-scale data processing. Therefore, in practical applications, a suitable technical route should be selected according to data scale, task type, and computation frequency.

(3) MongoDB and PostgreSQL/PostGIS Data Management

MongoDB is a document-oriented NoSQL database suitable for storing semi-structured data with flexible structures and frequently changing fields. For example, monitoring equipment logs, JSON-format data, acquisition records, and metadata descriptions can be organized as documents. MongoDB has advantages in flexible schema design and scalability, which makes it convenient for rapid access to multi-source data. However, in scenarios requiring complex relational queries and strong transactional consistency, it is not as stable as relational databases.

PostgreSQL is a relatively complete open-source relational database. With the PostGIS extension, it can support spatial geometric objects, spatial indexing, and spatial queries, making it suitable for managing geoscience spatial data such as borehole points, geological boundaries, sampling locations, administrative boundaries, grid cells, and model indices [7]. For geoscience applications requiring attribute filtering, spatial overlay, buffer analysis, and multi-table association, PostgreSQL/PostGIS has high practical value. Compared with MongoDB, PostgreSQL/PostGIS is more suitable for data management scenarios where the structure is clear, relationships are explicit, and spatial query constraints are required.

(4) Python and the MapReduce Computing Model

Python has the advantages of concise syntax, a rich library ecosystem, and strong cross-platform capability. In geoscience data processing, it is commonly used for data cleaning, file conversion, spatial analysis, statistical computation, and post-processing of model results. For small-scale data, Python can directly use libraries such as pandas and NumPy for computation. For larger datasets, Python scripts can be combined with the MapReduce idea, where data are divided into multiple subtasks for local computation and then intermediate results are summarized, reducing the memory pressure of a single computation.

The basic idea of MapReduce can be summarized as "divide and conquer". The Map stage is responsible for grouping, filtering, counting, or transforming input data and outputting intermediate key-value pairs or local results. The Reduce stage is responsible for merging, reducing, and globally computing the outputs of multiple Map tasks. For column statistics in geoscience attribute tables, segmented aggregation of monitoring data, and batch calculation of sample indicators, the MapReduce model has good interpretability and scalability. Figure 2 presents the Python-MapReduce column mean calculation process summarized in this paper.

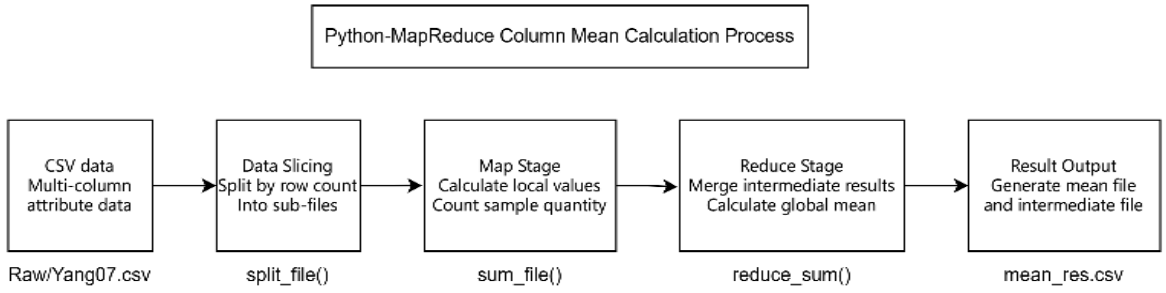

Figure 2. Workflow of Python-MapReduce-based column mean calculation

4. Technical Implementation and Experimental Organization

(1) Experimental Environment and Basic Operations

The practical part of this paper uses a Linux server as the operating environment and organizes the operations of remote connection, file transfer, command-line operation, Python script execution, and Java program compilation. Server remote connection establishes an interaction channel between a local computer and a cloud or remote Linux environment. File transfer moves raw data, script files, and output results between local and server environments. Command-line operations are used to switch directories, view files, run programs, and check results.

Table 2. Common commands used in Linux server practice

Command	Function	Typical usage
ls / ls -a / ls -l	View directory contents	Check current files, hidden files, and detailed information
mkdir	Create a directory	mkdir input; mkdir output
cd	Change directory	cd input; cd..
mv	Move or rename files/directories	mv old.csv new.csv
cp	Copy files or directories	cp data.csv backup.csv
rm	Delete files or directories	rm temp.csv
cat	View text content	cat MapReduce.py
vim / nano	Edit text files	vim MapReduce.py; nano DemoType.java
python	Run a Python script	python MapReduce.py
javac / java	Compile and run Java programs	javac HelloWorld.java; java HelloWorld

(2) Data Slicing and Loading Method

Under the MapReduce idea, raw data first need to be divided into several smaller data slices. The purpose of data slicing is not to change data content, but to decompose the whole computational task into multiple local tasks so that each subtask can complete part of the computation independently. In the course practice, the original CSV file is split by a specified number of rows, and the split sub-files are written into the input directory. Then, the program reads each sliced file, loads the specified columns, and converts them for subsequent computation.

In the implementation process, the split_file function controls the number of data rows contained in each slice file through the scope parameter, while the loadtext function reads the data within the specified column range. For geoscience attribute tables, similar methods can be used to split data by spatial partition, time segment, or file size, thereby providing input for subsequent parallel computation.

(3) Map Stage: Column Sum and Sample Count Statistics

The core task of the Map stage is to perform local statistics on each data slice. Taking column mean calculation as an example, it is necessary to first calculate the sum of specified columns in each slice and record the number of samples in that slice. Then, the Reduce stage can calculate the global mean based on the column sums and sample counts of all slices without loading all raw data into memory at one time.

(4) Reduce Stage: Global Mean Calculation and Result Output

The Reduce stage is used to summarize the local results output by each slice. In the initial implementation, the column sums of all slices were first added and then divided by the total number of samples. In the improved implementation, a sample-count field was added to the intermediate result so that the program could retain sample-size information during merging. It should be noted that if different slices have different sample sizes, directly averaging the slice means may lead to bias. A more robust method is to reduce the column sums and sample counts separately and finally divide the total column sum by the total sample count to obtain the global mean.

The practical running results show that the program can complete batch statistics for multiple columns and output the mean value of each indicator. According to the original report, one calculation produced mean values for 13 statistical fields, as shown in Table 3. Because the original field names were not provided in the report, they are temporarily represented as Indicator 1 to Indicator 13 in this paper.

Table 3. Example of MapReduce multi-column statistical calculation results

Indicator	Mean	Indicator	Mean	Indicator	Mean
Indicator 1	477.1917	Indicator 2	7.3445	Indicator 3	8.8310
Indicator 4	15.2394	Indicator 5	11.9296	Indicator 6	77.1690
Indicator 7	442.1831	Indicator 8	17.1408	Indicator 9	32.5775
Indicator 10	3.1972	Indicator 11	18.0845	Indicator 12	2.4366
Indicator 13	5.6338				

```
4260.96,25.3,61.0,89.0,67.0,487.0,2202.0,94.0,349.0,17.0,147.0,16.0,71.0
4224.95,58.26,82.0,206.0,172.0,615.0,6279.0,221.0,314.0,31.0,173.0,18.0,27.0
4263.25,193.90000000000003,101.0,212.0,143.0,873.0,3296.0,311.0,331.0,20.0,265.0,86.0,31.0
4370.969999999999,20.5,82.0,112.0,89.0,625.0,2446.0,115.0,210.0,40.0,115.0,9.0,33.0
4313.92,45.3,72.0,144.0,111.0,954.0,2647.0,141.0,308.0,29.0,172.0,15.0,52.0
4369.18,17.3,71.0,71.0,87.0,624.0,7169.0,67.0,242.0,19.0,128.0,9.0,42.0
4340.91,56.400000000000006,73.0,123.0,98.0,657.0,6152.0,126.0,303.0,29.0,146.0,11.0,58.0
3736.4700000000003,104.5,85.0,125.0,80.0,644.0,1204.0,142.0,256.0,42.0,138.0,9.0,86.0
```

Figure 3. Example of output result from the Map function

```
4260.96,25.3,61.0,89.0,67.0,487.0,2202.0,94.0,349.0,17.0,147.0,16.0,71.0,9
4224.95,58.26,82.0,206.0,172.0,615.0,6279.0,221.0,314.0,31.0,173.0,18.0,27.0,9
4263.25,193.90000000000003,101.0,212.0,143.0,873.0,3296.0,311.0,331.0,20.0,265.0,86.0,31.0,9
4370.969999999999,20.5,82.0,112.0,89.0,625.0,2446.0,115.0,210.0,40.0,115.0,9.0,33.0,9
4313.92,45.3,72.0,144.0,111.0,954.0,2647.0,141.0,308.0,29.0,172.0,15.0,52.0,9
4369.18,17.3,71.0,71.0,87.0,624.0,7169.0,67.0,242.0,19.0,128.0,9.0,42.0,9
4340.91,56.400000000000006,73.0,123.0,98.0,657.0,6152.0,126.0,303.0,29.0,146.0,11.0,58.0,9
3736.4700000000003,104.5,85.0,125.0,80.0,644.0,1204.0,142.0,256.0,42.0,138.0,9.0,86.0,8
```

Figure 4. Example of aggregated result from the Reduce function

(5) Java Program Compilation and Execution Practice

In addition to Python scripts, the course practice also involves the compilation and execution of Java programs on a Linux server. Java is a common development language in the Hadoop ecosystem. Mastering basic JDK environment configuration and Java program execution helps understand the underlying task submission and program execution mechanisms of Hadoop. The practice process mainly includes installing JRE/JDK, writing Java source files, using the javac command to compile class files, and using the java command to run compiled programs.

Taking the HelloWorld program as an example, the command "javac HelloWorld.java" is used to compile the program, and "java HelloWorld" is then used to run the program and output the result. Similar methods can also be applied to basic experimental programs related to data types, class inheritance, inner classes, anonymous classes, file I/O, and ArrayList. Although these Java examples are not geoscience data processing algorithms themselves, they provide basic training for understanding program compilation, operation, and debugging in big data platforms.

(6) WebGIS-Based Disaster Monitoring and Early Warning Application of Geoscience Big Data

In geoscience big data applications, disaster monitoring and early warning is a typical scenario with multiple data sources, high timeliness requirements, and strong demands for spatial representation. Instead of separately discussing environmental monitoring, resource management, and geological exploration, this paper selects a relatively complete application case for explanation. Therefore, this section focuses on WebGIS-based disaster monitoring and early warning. By combining a previous flood monitoring and prediction system with GIS disaster data visualization practice, it illustrates the overall application process of geoscience big data from data acquisition, data storage, and model computation to spatial visualization and publication.

From the perspective of business objects, flood disasters, geological hazards, forest fires, and other natural disasters or emergency events differ in type, but they all have clear spatial location attributes, dynamic evolution characteristics, and emergency response requirements. Rainfall, terrain and landform, surface cover, historical disaster points, real-time reported information, and monitoring station data can all serve as data sources for risk identification and early warning analysis. Through geoscience big data technologies, these heterogeneous datasets can be organized into databases and file management systems, and model results can be expressed in the form of maps, charts, and warning information.

At the system implementation level, "data acquisition - data storage - model computation - spatial visualization - warning publication" can be regarded as a general workflow for disaster-oriented geoscience big data applications. The front end is responsible for map interaction, layer control, chart display, and warning prompts. The back end is responsible for interface services, model invocation, and data writing. PostgreSQL/PostGIS manages spatial objects and attribute data, while remote sensing images, meteorological data, historical disaster data, and user-reported data jointly constitute model inputs. This workflow can effectively demonstrate the integrated application value of geoscience big data technologies in practical business scenarios.

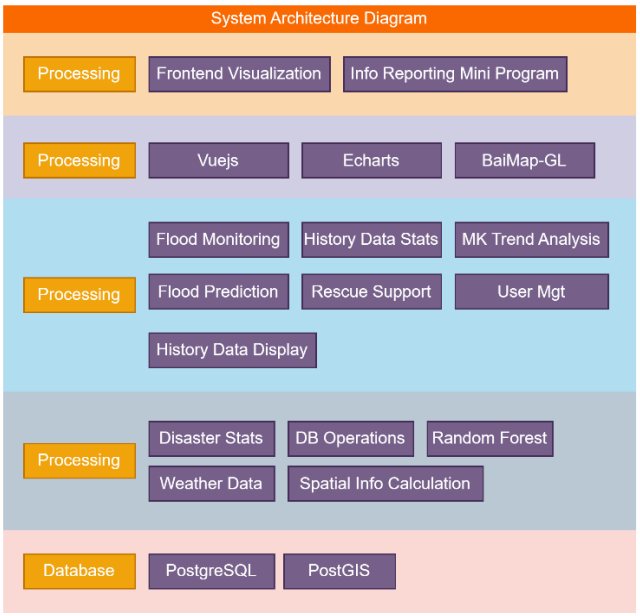

Figure 5. Example of the overall architecture of a WebGIS disaster monitoring and prediction system

System functions can be organized around user management, historical data statistics, disaster monitoring and prediction, auxiliary rescue, and visualization display. This functional structure corresponds to the big data technical system discussed above. Linux or cloud servers provide the deployment environment, databases manage structured data and spatial objects, Python or other languages undertake data cleaning and model computation tasks, and the WebGIS interface transforms analytical results into interactive spatial information.

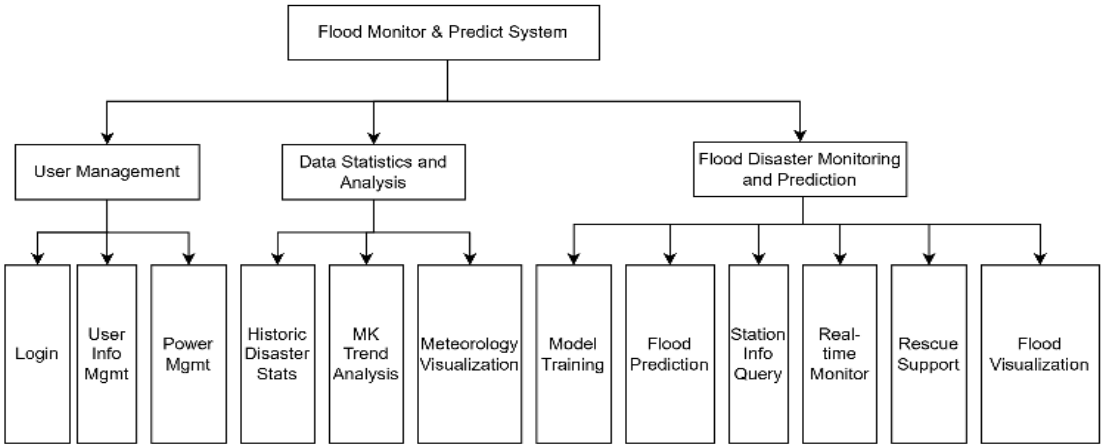

Figure 6. Functional composition of a WebGIS flood monitoring and prediction system

At the data layer, remote sensing images, DEM, NDVI, land-use type, and meteorological tables need to be transformed into a unified spatial indexing system. Taking flood prediction as an example, NDVI and land-use type can reflect surface cover and underlying surface differences, DEM can describe topographic elevation conditions, and meteorological data can reflect rainfall and temperature changes. After these data are extracted and associated according to grid stations or spatial units, a feature matrix required for machine learning or statistical analysis can be formed.

Figure 7. Example of spatial sample organization based on grid stations

At the application layer, model results need to be presented intuitively through a map interface. Prediction results can be displayed as point distributions, heat maps, and warning pop-ups. Emergency dispatch functions can further overlay spatial analysis results such as buffers, nearest facility selection, and route navigation. Therefore, geoscience big data is not only a problem of background data storage and batch computation; it also requires WebGIS to transform analytical results into business information that is understandable, operable, and usable for decision support.

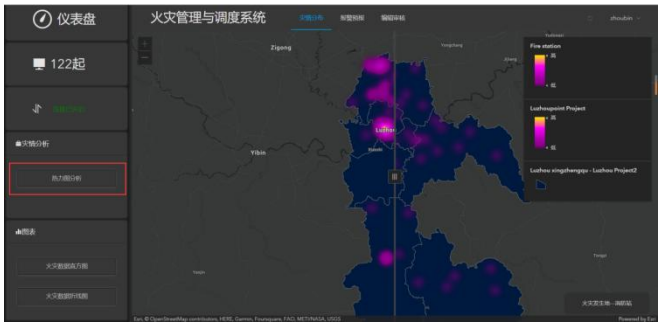

Figure 8. Heat map analysis display example of GIS disaster events

Overall, the disaster monitoring and early warning case can centrally reflect the technical chain of geoscience big data applications. First, multi-source data must be cleaned, converted, and spatially registered before entering a unified analysis workflow. Second, spatial databases and indexing mechanisms are important foundations for supporting rapid query, layer overlay, and model invocation. Third, model computation results must serve practical business through map visualization, chart statistics, and warning prompts. Therefore, using WebGIS-based disaster monitoring and early warning as an application case can better highlight the comprehensiveness and practicality of geoscience big data technologies than simply listing multiple application directions.

5. Conclusion

Based on a course report on geoscience big data, this paper reorganizes and deepens the related technologies and applications of geoscience big data in the structure of an academic paper. The study shows that geoscience big data has the characteristics of multi-source heterogeneity, strong spatial association, clear temporal variation, and large processing scale. Its effective utilization requires the joint support of multiple technologies, including cloud servers, Linux environments, distributed computing, database management, and automated scripts. Hadoop and MapReduce embody the basic idea of big data batch processing and are suitable for distributed statistics and aggregation of massive datasets. MongoDB is suitable for flexibly organizing semi-structured data, PostgreSQL or PostGIS is suitable for managing structured and spatial geoscience data, and Python can automate workflows such as data cleaning, slicing, statistics, and result output.

Through practical operations such as Linux server connection, basic command-line operation, Python-MapReduce column mean calculation, and Java program compilation and execution, the technical process of geoscience big data processing can be understood more intuitively. The practice shows that after raw data are organized by slicing, local statistics can be completed in the Map stage and then merged in the Reduce stage to realize batch processing of multi-column attribute data. This provides a methodological basis for larger-scale geoscience data analysis.

In terms of application, this paper takes WebGIS-based disaster monitoring and early warning as an example to illustrate that geoscience big data technologies can integrate multi-source spatial data, attribute data, model computation, and visual expression to serve disaster risk identification, dynamic monitoring, and decision support. This case indicates that the key to geoscience big data application does not lie in a single technical tool, but in the coordination among data organization, spatial computation, model analysis, and business presentation. Therefore, a complete geoscience big data application should not only emphasize data storage and batch computation, but also focus on spatial database organization, model integration, visualization, and the transformation of analysis results into practical decision-support information.

However, the further application of geoscience big data still faces several challenges. First, data standards and quality control remain fundamental issues. Differences in data acquisition standards, coordinate systems, field naming, classification systems, and accuracy requirements among different regions, departments, and projects increase the difficulty of data integration. Missing values, outliers,

duplicate records, coordinate offsets, inconsistent units, and semantic inconsistencies may also affect the reliability of data-driven analysis. Therefore, unified data standards, metadata specifications, quality-checking mechanisms, and traceable data processing records should be strengthened in future platform construction.

Second, data security and privacy protection should be given sufficient attention. Geoscience data may involve mineral resources, infrastructure, environmentally sensitive areas, engineering construction, and production management information, some of which may have confidentiality or commercial sensitivity. In cloud servers and big data platform environments, security mechanisms such as account permission management, network access control, data backup, log auditing, encrypted transmission, and hierarchical authorization are necessary to prevent data leakage, accidental deletion, and illegal access. For data involving people, enterprises, or production activities, desensitization, aggregated statistics, and authorized access should be adopted to balance data value utilization and security protection.

Third, technology integration and application implementation remain important practical problems. The geoscience big data technical system involves servers, databases, programming languages, distributed frameworks, spatial analysis, visualization, and business systems. The more technical components involved, the higher the complexity of system integration and the greater the requirements for developers, maintainers, and users. Therefore, an appropriate technical route should be selected according to data scale and business needs. For small and medium-sized projects, a lightweight solution such as "cloud server + PostgreSQL/PostGIS + Python scripts + visualization tools" can be adopted. For massive imagery, long-term monitoring, and regional platforms, Hadoop, Spark, and distributed storage systems can be further introduced.

In the future, geoscience big data will develop toward more standardized, intelligent, dynamic, and collaborative directions. Data governance capabilities will be further enhanced, and data standards, quality control, and metadata management will become key tasks in platform construction. Cloud computing, artificial intelligence, and geoscience models will be more deeply integrated, promoting the transformation from data management to intelligent analysis. Three-dimensional geoscience models, digital twins, and real-time monitoring data will also be combined to form more dynamic geoscience information representations. Meanwhile, interdisciplinary collaboration will continue to strengthen, and geoscience data will be integrated with ecological, urban, energy, transportation, and socioeconomic data to serve comprehensive decision-making. Therefore, under the constraints of geoscience expertise, the deep integration of big data technologies with artificial intelligence, cloud computing, spatial databases, and WebGIS platforms should be promoted to improve the scientific value and practical service capability of geoscience big data.

Acknowledgements

This manuscript was prepared based on the course report on geoscience big data and supplemented with relevant figures from a previous WebGIS flood monitoring and prediction system and a GIS development competition project. The author appreciates the practical environment and materials provided during the course learning process.

References

[1] Li Z B, Chen J P, Wang D C. Development and research status of geoscience big data technologies in China and the United States [J]. Journal of Geology, 2020, 44(04): 345-355. (in Chinese)

[2] Peng J B, Li Z H. Can geoscience big data contribute to geological hazard prediction? [J]. Earth Science, 2022, 47(10): 3900-3901. (in Chinese)

[3] White T. Hadoop: The Definitive Guide[M]. 4th ed. Sebastopol: O'Reilly Media, 2015.

[4] Dean J, Ghemawat S. MapReduce: Simplified Data Processing on Large Clusters [C]//Proceedings of the 6th Symposium on Operating Systems Design and Implementation. 2004: 137-150.

[5] Ghemawat S, Gobioff H, Leung S T. The Google File System [C]//Proceedings of the 19th ACM Symposium on Operating Systems Principles. 2003: 29-43.

[6] Chang F, Dean J, Ghemawat S, et al. Bigtable: A Distributed Storage System for Structured Data [J]. ACM Transactions on Computer Systems, 2008, 26(2): 1-26.

[7] Obe R O, Hsu L S. PostGIS in Action [M]. 2nd ed. Shelter Island: Manning Publications, 2015.