Design and Implementation of Universal Hydrological Data Sharing Model Based on WebService

Xingguo Qiu¹, Yuyang Xie¹,*
¹Xi’an University of Science & Technology, Xi’an, China
*Corresponding author e-mail: 2454888413@qq.com

Abstract. The sharing of scientific data on hydrology and water resources is an indispensable part of the national scientific data sharing process. This paper introduces several common data sharing methods, analyzes the basic architecture and design ideas of the common hydrological data sharing model based on WebService, and attempts to solve the large data volume transmission and data security in the process of hydrological data sharing by serialization and compression. The problem of the aspect is finally verified by the specific procedure, which proves the feasibility and practicability of the data sharing model.

1. Introduction

Hydrological and water resources scientific data is an important basic information for studying water issues. It is the basic data for scientific research and planning in the fields of national economic construction, flood control and drought reduction, water resources sustainable use and management, water environment and water ecological protection. The sharing of scientific data on hydrology and water resources is an indispensable part of the national scientific data sharing project[1].

However, although the current hydrological resource data sharing in China has made some progress, there are generally problems such as imperfect sharing mechanism, imperfect technical standards, and immature sharing technology. The reason is that due to the development time of the system or the development department, there are often multiple heterogeneous information systems running on different software and hardware platforms running at the same time. The data sources of these systems are independent and mutually closed, making the data It is difficult to communicate, share and integrate between systems, thus forming an “information island”.

For the sharing of hydrological data, the General Technical Requirements and Data Collection Standards for Online Monitoring and Networking of Coal Mine Safety Production Prepared by the State Administration of Safety Supervision and the State Administration of Coal Mine Safety, which provide the following technical solutions for monitoring data networking technologies[2]: (1) File sharing Using FTP, the data is read into a file, the file format is uploaded to the designated receiving server, and the desired data file can be downloaded from the server to complete data sharing. The advantage of this method is that in the case of a large amount of data, it can be transmitted through a file without timeout and without occupying network bandwidth. The solution is simple, avoiding the concept of network transmission and network protocol. The disadvantage is that it is not suitable for real-time business; there must be a common file server, and there is a risk in the file server. Because the file may be tampered with, deleted, or leaked, etc.; the format of the file data must be agreed upon. When changing the file format, each system needs to be modified synchronously. (2) WebService. WebService is a data transmission method that supports multiple programming languages and
cross-platform. Based on this method, there are two data sharing strategies\cite{3}. The first is that the client actively sends data to the server-specific interface. The second is the server. Request data from the client. The advantage of this method is: easy to program, java currently provides a variety of frameworks, shielding the underlying communication details and data transfer conversion details; easy to control permissions. Through the transport layer protocol https, the transmitted data is encrypted, which improves the security; the versatility is strong, and the client is a .net architecture, java, python are all possible, making the service universal. The disadvantage of this method is that the server and the client must work at the same time. When the server is unavailable, the entire data interaction is impossible\cite{4}; when the amount of data transmitted is relatively large, the network bandwidth is seriously occupied, which may cause the connection to time out\cite{5}. This makes the service unreliable when the amount of data interacts. (3) Open database method. Connect to the same database for data exchange. The advantage of this approach: compared to the file mode transfer, because the same database used, the interaction is more simple; because the database provides quite a few operations, such as updates, rollbacks, and so on. The interaction mode is flexible, and through the database transaction mechanism, reliable data exchange can be achieved. Disadvantages of this approach: When more and more systems are connected to the database, because the connection pool of the database is limited, the connection allocated by each system will not be much. When the system is more and more, it may lead to There is no database connection available; in general, systems from two different companies will not open their own databases to connect with each other, as this will have security implications.

In view of the above analysis, the WebService technology is used here to build a data networking sharing project for solving XX Coal Industry Group Corporation. What needs to be solved here is the performance and security issues when transferring large data volumes.

2. WebService architecture

A complete WebService includes the following processes:

2.1. Key Technologies

SOAP (Simple Object Access Protocol, soap is used to describe the format of the information passed. It is a simple XML-based protocol that allows applications to exchange information over HTTP\cite{6}. When WebService sends requests and receives results via HTTP protocol The sent request content and the result content are encapsulated in XML format, and some specific HTTP message headers are added to illustrate the content format of the HTTP message. These specific HTTP message headers and XML content formats are SOAP protocols. SOAP provides standards. The RPC method to call the Web Service.

WebService uses the HTTP protocol to transfer data, and encapsulates the data in XML format (that is, XML describes which method to call the remote service object, what parameters are passed, and what the return result of the service object is). XML is the format for representing data in the WebService platform. In addition to being easy to build and easy to analyze, the main advantage of XML is that it is platform-independent and vendor-independent.
WSDL (WebServices Description Language): A Web service description language that describes how to access a specific interface [7]. Describe the Web Service and its functions, parameters, and return values. It is a standard format that both WebService clients and servers can understand. The WSDL file is saved on the web server and accessed via a url address. Before the client wants to call a WebService service, it needs to know the address of the WSDL file of the service. The WebService service provider can expose its WSDL file address in two ways: by registering with the UDDI server so that it can be looked up by someone; directly telling the client caller.

2.2. WebService Applicable Scenes and Advantages
Make up for the incompatibility of various platforms. The HTTP protocol is used for data transmission, and each host accessing the Web uses the HTTP protocol for communication. WebService supports multiple languages. Basically, each language supports XML [8]. The server and client can be programmed in different languages. In addition, WebService also supports communication across firewalls. If the application has thousands of users and is distributed around the world, communication between the client and the server will be a tricky issue [9]. Because there is usually a firewall or proxy server between the client and the server, using WebService technology will effectively solve these problems.

3. Hydrological data integration sharing
This paper is mainly to develop a WebService-based data sharing platform for a coal industry group company, including the server-side software of the group company and the client-side software of its subordinate mines. The main function of the server is to publish the shared data interface and receive the data pushed from the subordinate mine through the WebService technology; the main function of the client is to push the real-time data to the server and request the data shared by the server.

The overall architecture of the software is shown below.

Figure 2. The Architecture of WebService Server.

Figure 3. The Architecture of WebService Client.

The following two parts are designed from the server side and the client side, focusing on analyzing and solving the performance of large data volume transmission and data security.
3.1. server
The main function of the server is to publish the data interface, which is to share the data to the client and to receive the data pushed from the client. In the case of the published data interface, in principle, no matter which party knows the interface WSDL information, the shared data can be obtained from it, so the security of the data cannot be guaranteed. Therefore, the data security problem needs to be considered here.

![Figure 4. The IPO picture.](image)

For the data security problem, there are two solutions. One is that for the limitation of the SOAP Header, the user needs to have an identity verification, and the client must provide the correct identity information verification before proceeding to the next operation; the second is In order to encrypt the transmitted data, encapsulate the data information, such as using data set encapsulation, binary byte array encapsulation, etc., this needs to lead to the next transmission performance problem.

For the transmission efficiency problem, large data volume transmission often takes the form of data sets. Based on this form, there are the following solutions: (1) Return the DataSet object directly. The advantage of this method is that the code is reduced, easy to handle, and the small data volume is processed faster; the disadvantage is that the transmission of large data volume is slow and consumes network resources; (2) Returns the byte array of the DataSet object serialized with Binary. The advantage of this method is that it is easy to handle, and the Chinese content can be encrypted; the disadvantage is that the transmission of large data volume is slow and consumes network resources; (3) Returns the byte array of the DataSetSurrogate object serialized with Binary. The advantage of this method is that it is easy to handle, and the Chinese content can be encrypted; the disadvantage is that the transmission of large data volume is slow and consumes network resources; (4) Return the byte array of the DataSetSurrogate object serialized with Binary and Zip compressed. The advantage of this method is that when the amount of data is large, the performance improvement effect is obvious, and the compression ratio is large;

In view of this, comprehensive large data volume transmission performance and security issues, this paper chooses the fourth scheme, using serialization plus data compression, the data is returned to the client with the compressed byte array, and the client receives the data. Then decompress, so that it can solve the performance and security problems of large data volume transmission, and on this basis, compared with other three schemes.

3.2. client
The functions of the client mainly include two parts: push data and request data, respectively: the client periodically pushes data to the server; the client accesses the WebService interface server, and requests data provided by the server. There are two types of data push: configuration data and real-time data. For the configuration data, because the frequency of change is low, it is not necessary to push to the server at any time. It only needs to be pushed for the first time. It can be pushed at the specified time interval or compared with the first time, and then changed and pushed. For real-time data, it is constantly changing, and it needs to be pushed to the server at regular timing. At this time, it is necessary to consider the problem of data transmission, push frequency, data loss, and data push abnormality.

For the client, the prerequisite for all functions is to access the WebService server and design according to the interface information provided by it. This is more limited. Once the server changes, the client needs to be recompiled. Therefore, consider the general purpose here. Sex, for the
WebService server to design a special proxy class, use it to dynamically access the WebService interface. The main design idea of this class is to reflect the loader through the assembly path, parse the WebService server to provide WSDL information, load the class information, and call the method according to the provided method name. The proxy class can also flexibly respond to data sets returned by the server, single data, arrays, etc., dynamic matching of parameters, and so on.

Table 1. The definition of WebServiceProxy class.

| Attribute/method/event | Description |
|------------------------|-------------|
| url                    | interface address |
| Methodname             | Method name |
| param                  | By passing parameters through the object[] array, which can ignore the difference between the number of parameters and the type. |
| ExecuteQuery(string methodName, object[] param) | Calling interface, has a return value |
| Operate()              | Solve the problem of server-side disconnection in the process of pushing data |

The dynamic access of the WebService interface is solved, and the remaining problems are security and transmission problems. Here, since the server-side design adopts the form of data compression, the client directly calls the decompression program. The transmitted data is then processed. For the function that the latter client pushes to the server, the principle is the same as that the server side shares data with the client, and can also transmit in the form of data compression, so it will not be described here.

3.3. Testing
Mine hydrology data is relatively complicated for the data to be pushed. The data table of real-time data is as follows.

Table 2. The table of real-time data.

| elements        | Description                        | Length |
|-----------------|------------------------------------|--------|
| String[0]       | Sensor number                       | 15     |
| String[1]       | Monitoring point type number        | =<2    |
| String[2]       | The type of monitoring point        | =<10   |
| String[3]       | Units of measurement               | =<20   |
| String[4]       | The name of monitoring point        | =<225  |
| String[5]       | The location of sensor             | =<225  |
| String[6]       | Aquifer                            | =<20   |
| String[7]       | Whether to use the map (1 for use, 0 for unused) | 1     |
| String[8]       | The X coordinate of mine map        | =<10   |
| String[9]       | The Y coordinate of mine map        | =<10   |
| String[10]      | The elevation of orifice            | =<10   |
| String[11]      | Alarm red high ceiling              | =<10   |
| String[12]      | Alarm orange high ceiling           | =<10   |
| String[13]      | Alarm yellow high ceiling           | =<10   |
| String[14]      | Alarm red high lower limit          | =<10   |
| String[15]      | Alarm orange high lower limit       | =<10   |
| String[16]      | Alarm yellow high lower limit       | =<10   |
| String[17]      | Whether to use the alarm (1 is used, 0 is unused) | 1     |
| String[18]      | The depth of final hole             | =<10   |
| String[19]      | The status of sensor                | 1      |
| String[20]      | Upper limit of measurement value change | =<10 |
The WebService-based hydrological data sharing platform is debugged in the Visual Studio 2010 development environment and released using IIS7. Here we use two test methods for software performance, native test and cross-machine test, consider testing under the same network and host environment conditions, and compare the results, so the performance difference is mainly determined by different serialization and compression methods. The following is the final test result of the entire process of receiving a DataSet that returns 100,000 records directly from the WebService to the client:

Table 3. The result of testing.

| Test condition | Description | Test methods | Object | DataSet | DataSet+Binary | DataSetSurrogate+Binary | DataSetSurrogate+Binary+Zip |
|----------------|-------------|--------------|--------|---------|----------------|-------------------------|-----------------------------|
| The environment of Server | CPU:2.5Hz, Storage:4G | Native interface test, server and client are on the same computer | Local machine | 55s | 18s | 15s | 10s |
| The environment of Client | CPU:2.5Hz, Storage:4G | Remote interface test, the server and the client are in the LAN of the same network segment | Cross machine | 58s | 19s | 16s | 11s |

The test results show that the use of DataSetSurrogate + Binary + Zip can greatly reduce the amount of data, solve the performance problems in the transmission process, and at the same time largely solve the data security problem.

4. Summary
The general hydrological data sharing model based on WebService proposed in this paper is a good solution for the inconsistency of data between the relatively independent application management systems in large enterprises and the difficulty in realizing sharing in time. In hydrological data integration, the data sharing model integrates hydrological data from heterogeneous systems, and each department can still use its own independent application management system, but store the shared data that each department needs to use in the group's database. In this way, when one of the departments needs to use the information data of other departments, it can be obtained directly from the WebService interface server, without having to go through the cumbersome internal process to obtain the desired data. From a global perspective, this sharing model effectively promotes the integration and sharing of data.

Moreover, the paper also proposes a solution to the performance and security problems in the process of large data volume transmission, and through the program to verify, can meet the needs of users, due to time and space limitations, the text does not mention multi-user concurrent data request problems, The follow-up can be studied from the high concurrency side, and the data sharing model is continuously improved.

References
[1] Hua Chen, Jian Xu, Zhiyuan Xiao, Jiawei Yang, Jie Chen, Shenglian Guo, Chongyu Xu. Research and Design of Hydrology Data Sharing Platform [J] Water Resources Research, 2018, 7 (01): 10-18.
[2] Xiaoguang Tang. Research and design of data sharing platform based on subscription mechanism [D]. Northeast Petroleum University, 2013.
[3] Ping Li. Design and Implementation of Data Sharing and Exchange Platform Based on WebService[J]. Journal of Computer, 2016:17.
[4] Honglan Yu. Application of WebService in Campus Information System [J]. Computer Knowledge and Technology, 2018.
[5] Xiaoqin Xu. Implementation of SAP Interface Based on Webservice Technology [J]. Computer Knowledge and Technology, 2018, 14(14): 30-31+34.

[6] Deqing Li. Design and Implementation of Agent Management System Based on WebService Technology [D]. Qingdao University of Science and Technology, 2018.

[7] Xiaoyan Zhang. Application of WebService Technology in WeChat Development in Colleges and Universities[J]. Computer Programming Skills & Maintenance, 2018(10):63-64+87.

[8] Yang Zhao. Research and Application of WebService Push Technology [D]. Yanshan University, 2017.

[9] Weiqing Wan. Application and Implementation of WebService Technology in Distributed Development [J]. Software Development and Application, 2017:18.