Scholars Journal of Engineering and Technology (SJET) **ISSN 2321-435X (Online)**
Sch. J. Eng. Tech., 2015; 3(2B):170-174 **ISSN 2347-9523 (Print)**
©Scholars Academic and Scientific Publisher
(An International Publisher for Academic and Scientific Resources)
www.saspublisher.com

Research Article

Research on the network transmission of cooperative work in the logging platform of large data environment

Ruishan Du, Fuhua Shang
School of Computer and Information Technology, Northeast Petroleum University, Daqing, 163318, China

***Corresponding author**
Ruishan Du
Email: durs918@163.com

Abstract: Currently advanced methods of logging interpretation method such as acoustic-electric imaging, MRI, array acoustic have been widely applied. Data needed handling are large two-dimensional and even three-dimensional data. Meanwhile, as for the explanation and conclusion of logging processing and interpretation, it's necessary to share with other interpreters in a same project team. Cooperative work in logging processing and interpretation platform is a key technology to improve the interpretation efficiency of interpreter and that the function of large data transmission is a technological difficulty which needs tackling in cooperative work in logging processing and interpretation platform. Therefore, this paper proposes a network transmission method of cooperative work in the logging processing and interpretation platform of large data environment and studies real-time data of large data. In the paper, a prototype system of data transmission module is designed and realized, which realizes the point-to-point and point-to-plane functions of large data transmission. Besides, the paper also respectively tests P2P communication and the property of large data transmission in intranet. In the process of transmitting, oil field logging information, the security, holistic optimization and transmission efficiency are also taken into consideration in this paper.

Keywords: Cooperative work; Logging interpretation; Large data transmission.

INTRODUCTION

In a new market competition environment, modern scientific research is a collative process of cross-enterprises and inter-enterprises. It possesses features such as complementarity, timeliness, distributivity and isomerism because it needs groups of experts with various specialized knowledge in various departments and workplaces to cooperate with one another [1-3]. However, some works need various sections to cooperate to complete and share the results of them in the process of logging interpretation. When logging interpretation experts are not available to share logging data, it is necessary to analyze and explain quickly logging data to local users. The research based on network on cooperative work methods of logging platform plays significances in logging in order to better improve user's response speed and share excellent interpretation results. Cooperative work has been widely applied at home and abroad. In military application, there are diverse CI systems. Therefore, the paper proposes research on the network transmission of cooperative work in the logging platform of large data environment and studies problems involved in sharing the intermediate results of cooperative work of logging platform based on network and large data transmission of cooperative work.

TRANSMISSION PRINCIPLES OF COOPERATIVE WORK

Cooperative work is based on data transmission which includes real-time and off-line data transmission. The key problem of cooperative work of logging platform is real-time data transmission which demands both sides of communication to be online, and data can never be transmitted only when handshake is successfully made by the two parties. If the handshake fails, the sender will automatically record the data that needs transmitting at home. When the sender finds the receiver online, it will retransmit data to the receiver. The operating principles are presented as in the figure 1.

Real-time data transmission makes socket communication [4-5]. Online data transmission consists of three types: they are small data block transmission, large data block transmission and real-time data transmission. Text chatting, operational order, institutional framework and UI control hierarchy order belong to small data block transmission. Hard disk file, database of large data stream belong to large data block transmission. Because large data stream only serves one data stream interface to read underlying data such as transmitting file data. When the user needs to transmit files, he can just drag files and trigger the interface of online transmission, which simplifies the upper serialized operation and improve the efficiency of transmission.

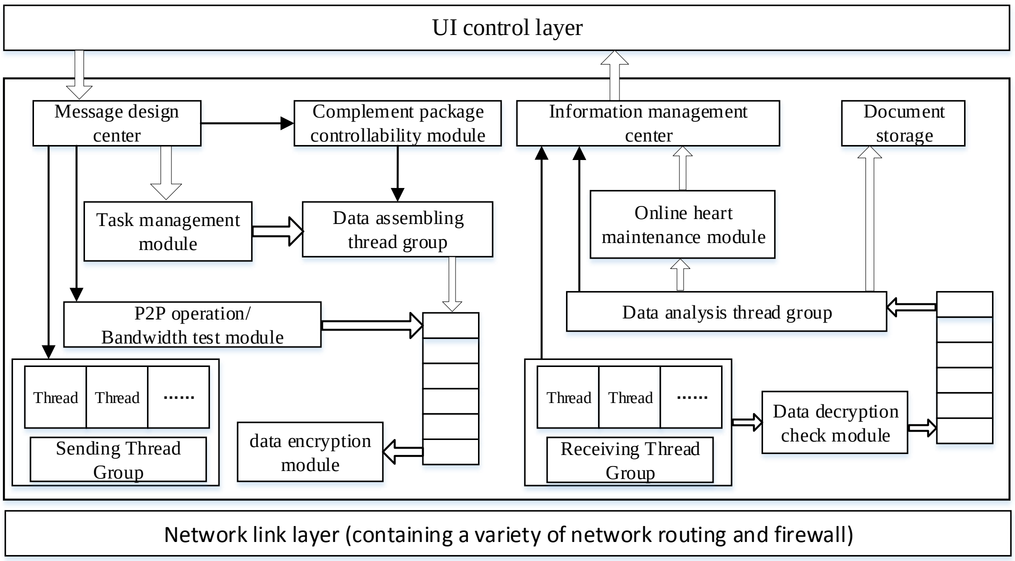
UI control layer
Message design center
Complement package controllability module
Information management center
Document storage
Task management module
Data assembling thread group
Online heart maintenance module
P2P operation/ Bandwidth test module
Data analysis thread group
Thread
Thread
......
Sending Thread Group
data encryption module
Thread
Thread
......
Receiving Thread Group
Data decryption check module
Network link layer (containing a variety of network routing and firewall)

Fig-1: Real-time data transmission

KEY TECHNOLOGIES OF LARGE DATA TRANSMISSION

The capacity of large data transmission is a main part to present the cooperative work of logging platform and responsible to stably and efficiently transmit large data. Key technologies of large data transmission mainly include the compression and un-compression, fragmentation and combination, multithread processing and breakpoint resume of large data [6-7].

The compression and decompression of large data

The compression and un-compression of large data in the paper adopt dynamic Huffman and combine it with LZW. The compression algorithm of LZW named after Lemple-Ziv-Welch who create it together is a novel compression method. It uses an advanced string list compression by putting the first string into a string list and referring a string with a number. The compressed file only store numbers not strings, which greatly improve the compression efficiency. To our surprise, the string list can also be set up correctly no matter in the process of compression or un-compression. When compression or un-compression is over, the string list will be abandoned.

In LZW algorithm, set up a string list first, and then put the first string into a string list and finally show it with a number which relates to the position of the string in the list. Then put the number into the compressed file. When the string reappears, we can replace it with the figure and put the figure into the file. The string list will be abandoned after the compression finishes. For example, the string list "print", we use number 266 to present it when compressing. Whenever the string "print" reappears, we can always represent it with number 266 and put string "print" into the string list. When we see number 266 in the process of decoding image, we can know that number 266 stands for string "print". When uncompressing, the string list can be rebuilt according to compressed data.

Under the circumstance of bad network environment and large data volume, it’d better to less occupy network bandwidth. Therefore, the compression and un-compression technologies are adopted to compress large data before it is transmitted to client. Finally, the compressed file is transmitted through the network [8].In that case, the time taken by compressing and un-compressing file will be counted in the time of document transmission, which will have some bad influence on capacity.

The fragmentation and combination of large data

However, due to the limited buffer of socket, only about 4K data can be transmitted at each time. Therefore, it is necessary to fragment the client when transmitting large data and then recombine it at destination. In fact, communication middleware have encapsulated the data and provided interface needed to send large data. As for single large data（>500mb）, it needs to fragment them and turn them into comparatively small files and send them. When receiving files, make those small files combine.

Buffer queue

Same to multithreading, buffer queue also aims to solve problems of large data volume of system. There are two application scenarios in the transmission system of large data, which are memory storage and waiting queue.

As we all know, the read-write speed of hard disk is slower than that of memory. Therefore, when it comes to application scenarios that demand higher speed and lower security of the circulation speed of data, memory storage can meet such need [9]. That is, in business process, all the messages are stored in memory according to some rules and all read-writes are also operated in memory, which can greatly speed circulation. The drawback of such application model lies in that hard disk needs to be highly stable. When the server fails and restarts, real-time data cannot be recovered.

When interaction is made among clients, waiting queue transmits corresponding information of clients by means of batch mode. Buffers are made and disposed one after another in clients, which can reduce the stress of client and timely free communication link to dispose other requests.

Breakpoint resume

Large data file always needs to be transmitted which require long-term connection. However, due to current network conditions, some inevitable jitters or the system halted of client or server will interrupt the transmission. In such cases, breakpoint resume is needed.

Breakpoint resume records the transfer status of large data through the description files of server's local-storage. It tries to not fail data by recording information by means of timing and fixed length. When there is interruption, the server will automatically search for information according to the information of client just by reconnecting the client. Then the transmission will continue from the last breakpoint position till it's finished.

ARCHITECTURE OF THE DATA TRANSMISSION SYSTEM

According to the open and interconnected reference model, the large data transmission system architecture of the cooperative work of logging platform in the paper mainly consists of network monitoring module, data transmission module, data management module and data processing module [10-11].

Data processing module takes charge of sending and receiving the interpreting data and operational data of logging. According to the different data types, security classification and corresponding rules of data transmission, operations such as encrypting, decrypting, compressing, uncompressing, coding, decoding, calibrating, fragmenting and combining are processed to data. The transmitted data are encapsulated into several internal modules of UDP data package according to the data classification (conversely, recombine several UDP data packages into block data of corresponding type.).

Data management module takes charge of UDP data package which has been processed by data needs to be transmitted or uniformly manage and distributes UDP data package which was received some time before. Meanwhile, it takes charge of managing the buffer, concurrence and integrity of data.

Data transmission module takes charge of data transmission. Transmission paths is determined by judging the environment of network link. The exact sending and receiving processes of data are also realized.

Network monitoring module takes charge of monitoring the connection timeout of local and target users, informing them to offline, managing local network anomaly as well as testing the network bandwidth of the two parties involved in data transmission.

EXPERIMENT OF REAL-TIME DATA TRANSMISSION

The experiment is based on the normal work of logging platform and applies the real-time data transmission of the cooperative work system of logging platform. The testing environment is general network. Different sizes of data packages have influence on large data transmission capacity, it is showed as table 1 and figure 2.

Table-1: The comparative of transmission speed

	2MB	10MB	30MB	50MB	100MB	200MB	500MB	900MB
10kb	1.532	4.593	13.932	23.468	49.896	105.229	239.780	446.316
20kb	0.996	2.492	7.498	12.276	26.246	53.430	117.487	217.268
50kb	0.792	1.312	3.999	6.088	15.176	24.498	58.068	113.665
100kb	0.723	1.201	3.156	5.812	11.103	22.388	51.426	87.421
1024kb	0.895	1.231	3.526	5.286	12.298	22.886	52.603	89.89

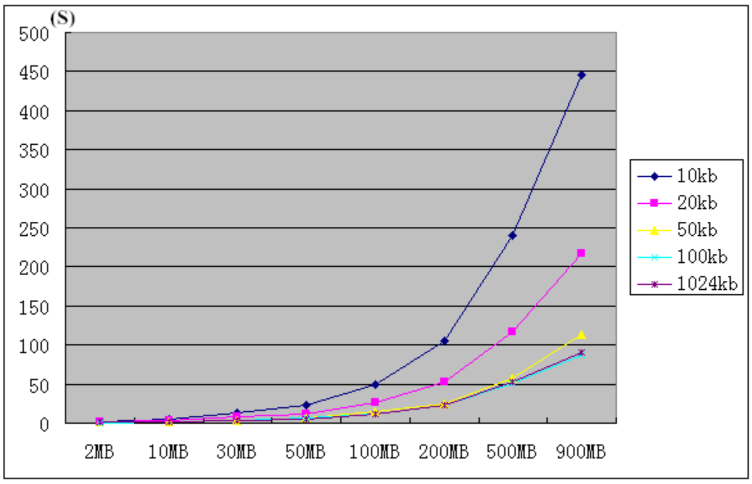

Fig-2: The comparative of transmission speed

From the result, it is known that bigger differences on the transmission rate of different data package settings appear as the transmission quantity of data increases. It indicates that the data transmission rate speeds with the increase of data package settings.

Different sizes of data packages have influence on large data transmission speed, network occupation and CPU occupation. As a result, it is necessary to adopt proper control mode of transmission process. The comparison test on the occupation of network and CPU and transmission rate when the sizes of data packages are different showed in table 2 and figure 3.

Table 2. The comparative of three performances

	10kb	20kb	50kb	100kb	1024kb
Occupation of network (%)	18	36	84	90	90
Occupation of CPU (%)	6	15	36	48	48
Transmission rate (m/s)	2	4	9	10	10

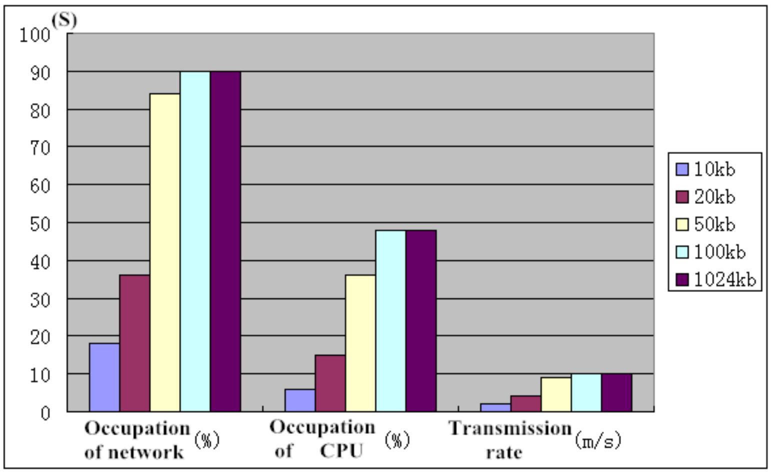

Fig-3: The comparative of three performances

Form the result, it is known that when the data package setting is larger, the resource consumption of the system and network occupation are also higher. Therefore, when choosing the size of data package, the resource allocation of system resources in the production environment should also be taken into consideration in order not to excessively occupy system resources and affect the operation of the whole system.

CONCLUSION

The paper proposes the network transmission research on the cooperative work of logging platform in large data environment develops relevant transmission modules and makes a teat on them. As the test shows, the transmission efficiency of the real-time data transfer mode is quite high and the transmission is stable. What’s more, the interior programming can be improved greatly in theory because it is made by hand.

What is going to do is to perfect the detailed design of service and other service function of data transmission.

Acknowledgment

This paper is supported by Youth Foundation of Northeast Petroleum University (2013NQ120, NEPUQN2014-18).

REFERENCES

1. Schmidt K, Bannon L; Constructing CSCW: The First Quarter Century . Computer Supported Cooperative Work (CSCW), 2013; 22(4-6):345-372.
2. Martin Hilbert, Priscila Lopez; The world’s technological capacity to store, communicate, and compute information.Science, 2011; 332(6025):60-65.
3. Lavrishcheva M; E. Software engineering as a scientific and engineering discipline. Cybernetics and Systems Analysis, 2008; 44(3):324-332.
4. Tawil ARH, Taweel A, Naeem U, Montebello M, Bashroush R, Al-Nemrat A; Integration operators for generating RDF/OWL-based user defined mediator views in a grid environment. Journal of Intelligent Information Systems, 2014; 43(1):1-32.
5. Alencar JMU, Andrade R, Viana W, Schulze B; P2PScheMe: a P2P scheduling mechanism for workflows in grid computing. Concurrency and Computation: Practice and Experience, 2012; 24(13):1478–1496.
6. Sakr S, Liu A, Fayoumi AG.; The Family of MapReduce and Large Scale Data Processing Systems. eprint arXiv:1302.2966, 2013; 46(1):28-36.
7. Leiserson CE, Schardl TB, Sukha J; Deterministic Parallel Random-Number Generation for Dynamic-Multithreading Platforms. ACM SIGPLAN NOTICES, 2012; 47(8):193-204.
8. Yewang Chen, Jinshan Yu; Generic Programming and Design. Computer Science, 2006; 33(04):253-257.
9. Wenxiao Li, Xiaohu Yang; Storage Model Based on Distributed Cache for Message Oriented Middleware. Computer Engineering, 2010, 36(13):93-95.
10. Tsai W, Wei X, Chen Y, Paul R, Chung R, Zhang D, Tsai W, Wei X et al; Data provenance in SOA: security, reliability, and integrity. Service Oriented Computing and Applications, 2007; 1(4):223-247.
11. Tu M, Ma H, Xiao L, Yen IL, Bastani F, Xu D; Data Placement in P2P Data Grids Considering the Availability, Security, Access Performance and Load Balancing. Journal of Grid Computing, 2013; 11(1):103-127.