

Fault-tolerant Mechanism for Cloud Storage System with Integrity Verification Mechanism

Liu Hong qing¹, Huang Yan¹

¹. *Hunan Vocational College of Modern Logistics, Changsha Hunan, 410131*

Abstract

A self-adaption fault-tolerant mechanism based on access frequency (SFMAF) is proposed for the cloud storage platform, wherein this mechanism aims at adopting the approximate LRU (Least Recently Used) algorithm to maintain a file access frequency table in order to self-adaptively adjust the fault-tolerant mode. Specifically, the replica redundancy mechanism is adopted in SFMAF for the frequently accessed files and RS (Reed-Solomon) erasure code tolerant mechanism is adopted in SFMAF for the rarely accessed files. The experiment result shows: compared with the replica redundancy mechanism, SFMAF can reduce the transmission flow of the internal system data at the cost of increasing CPU and memory usage rates in an acceptable range, namely: the system storage space is reduced.

Keywords: *Cloud Storage; Erasure Code; File Access Frequency Table; Self-adaption; Fault-tolerant Mechanism*

1. Introduction

Along with the sharp increase of data volume, the storage system shall have higher reliability and higher throughput performance to meet the data storage requirement. Specifically, the distributed storage system can disperse the pressure of the data servers and obtain relatively high throughput capacity to provide high-reliability and high-performance storage system, thus becoming the storage technology frequently used by current service providers, wherein cloud storage is a typical application.

The replica redundancy mechanism is adopted as the fault-tolerant mechanism in most cloud storage platforms [1]-[2]. Although such system has the features of simple implementation and high throughput capacity as well as severer load balancing, yet the replica backup needs to occupy a large storage space. Some scholars have proposed the erasure code fault-tolerant mechanism as the fault-tolerant mechanism of the cloud storage system; although this method can reduce the storage space occupied for backup and provide higher reliability, yet the server load balancing capability is reduced. In practical production activities, not all files in the cloud storage system are frequently accessed, and quite a number of files are rarely accessed or regarded as the backup for further check. In allusion to such situation, the above two methods are combined in this article, and the file frequency table based on the approximate LRU algorithm is added to the file system. According to the file access frequency, the backup strategy of the corresponding file is self-adaptively switched between the replica redundancy mechanism and the erasure code fault-tolerant mechanism to reduce the system storage space at the cost of increasing the system resources in an acceptable range.

2. Introduction to Hadoop Distributed File System

2.1. Hadoop Distributed File System Architecture

Hadoop distributed file system, abbreviated as HDFS, is not only the storage foundation layer of Hadoop framework, but also a key part in Hadoop architecture for realizing such functions as input and output data storage, management and fault control in the framework. Moreover, HDFS has the features of high reliability, high expandability, high file throughput rate and strong portability, thus applicable to the distributed system on the large clusters of the low-cost common parts.

As the typical master-slave architecture, HDFS is mainly composed of three parts, namely: Name Node, Secondary Name Node and Data Node [4]-[6]. Specifically, the system usually includes only one name node which is responsible for recording the metadata of the file system and organizing the data nodes in a cluster as well as coordinating the client terminal to access relevant files. The secondary name node is used to back up the metadata in the name node and immediately substitute the name node when the name node is invalid, thus to ensure the normal operation of the system. The data node is used to store the file blocks for data management. HDFS cluster includes multiple data nodes and the cluster scale can be expanded according to actual needs; moreover, any host computer which can be connected to the Internet and runs in Linux operating system can be registered as a data node.

2.2. Fault-Tolerant Mechanism of Hadoop Distributed File System

At the beginning of the design, Hadoop framework is defined to run in the low-cost common hardware. In HDFS, hardware fault is regarded as a common state, and the corresponding fault mechanism is adopted to ensure the system robustness.

The replica redundancy strategy is adopted as the fault-tolerant strategy of HDFS. Firstly, HDFS copies the replica of each data block, and the name node controls the data copying process according to the designated replica number, namely: the database is transmitted in a pipeline mode between the data nodes according to the replica number requirement to form the redundancy replica. Secondly, for the replica number maintenance, each data node will periodically send heartbeat information to the name node; if the name node detects that a certain data node fails to successfully send the heartbeat information, then the name node will mark this data node as an invalid data node and will not send IO request to this data node. In the above case, the data block which the invalid data node intends to associate with has been lost. Additionally, the name node will periodically check the replica number of the data block; if the name node detects during the check process that the replica number of some data blocks is less than the designated replica number, then the name node will select a suitable opportunity to send the file block copying command to the data node which still contains the corresponding data block, thus to ensure the consistency between the replica number and the parameter setting.

3. Reed-Solomon (RS) Erasure Code

In recent years, many researchers take the erasure code as the redundancy mechanism in the distributed system to improve service reliability. The basic thought of the erasure code is as follows [7]: k blocks of source data are encoded to generate n blocks of complied target codes ($n > k$), k blocks of codes optionally selected from n blocks of codes can be used to recover k blocks of source data. In n blocks of compiled codes, the loss of at most $(n-k)$ blocks of codes will not influence data recovery. We call such code as (n, k) erasure code.

RS erasure code, proposed by I.S. Reed and G.S. Solomon in 1960[8]-[9], can be divided into two types according to different matrixes generated thereby: Vandermonde RS code and Cauchy RS code. Therein, RS encoding method is the unique MDS (maximum distance separable) encoding method that can satisfy any data disk number n and redundancy disk number m ; the following aspects are mainly considered in R-S erasure code algorithm[10]: adopt Vandermonde matrix for relevant calculation to generate and maintain the check data; adopt Gaussian elimination to generate and maintain the check data in the faults; adopt Gaussian elimination to recover data from faults; adopt Galois field for the operation in a finite field.

1. Calculate the checksum: $X = (x_1, x_2, \dots, x_m)$ is defined as the metadata packet vector, $Y = (y_1, y_2, \dots, y_n)$ is defined as the checksum data packet vector, $G_{m \times n}$ is the linear erasure code generation matrix and $G_{i,j} = j^{i-1}$ is Vandermonde matrix.

$$G = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1^{m-1} & 2^{m-1} & 3^{m-1} & \dots & n^{m-1} \end{pmatrix} \quad (1)$$

Set $Y=XG$ to obtain the following formula:

$$\begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1^{m-1} & 2^{m-1} & 3^{m-1} & \dots & n^{m-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (2)$$

2. Maintain the checksum: when data X are changed as X' , the corresponding checksum data shall be also changed, namely: the original data shall be deducted from the changed data in order to recalculate the checksum, and the new checksum calculated thereby shall be added to the original checksum to obtain the final checksum, as shown in the following formula:

$$Y' = Y + (X' - X)G \quad (3)$$

3. Adopt the Gaussian elimination method to recover the invalid data: matrix $A = \begin{pmatrix} I \\ G \end{pmatrix}$ and

vector $E = \begin{pmatrix} X \\ Y \end{pmatrix}$ are defined, wherein matrix I is a unit matrix; then, set $AX = E$ to obtain the following formula:

$$\begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 2^{m-1} & 3^{m-1} & \dots & n^{m-1} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \quad (4)$$

When the data in vector E are lost, it is necessary to delete the corresponding line in matrix A to obtain new vector E' and new matrix A' , wherein $A'X = E'$. If m data in

vector E are lost, then the corresponding m lines in matrix A shall be deleted. Additionally, matrix G is a Vandermonde matrix and any two lines thereof are linearly independent of each other, so any two lines of the new matrix composed of p lines optionally selected from matrix A are also linearly independent of each other. Obviously, matrix A' is a nonsingular matrix, namely: matrix A' is an invertible matrix, and the value of vector X can be calculated and obtained through Gaussian elimination method according to $A'X=E'$. Therefore, when less than m data blocks are lost, the data can be effectively recovered through this method.

4. Adopt Galois field for the operation in a finite field in order to avoid the calculation result overflow. Galois field is closure to the arithmetic operation, and the addition/subtraction operation thereof is XOR operation. For multiplication/division operation, it is necessary to directly add or subtract the indexes of the generated elements and then implement the modular arithmetic thereto.

4. Self-Adaption Fault-Tolerant Mechanism based on Access Frequency (SFMAF) and the Implementation

4.1. Self-Adaption Fault-Tolerant Mechanism based on Access Frequency

According to the analysis of the file states in the storage system, the following strategies are adopted in SFMAF mechanism:

1. If a file is written in the system for the first time, then the default mode ---- replica redundancy mode is adopted in the system for data storage. For such storage mode, the system cannot judge whether the file will be frequently accessed after being written in the system, but the file may be much probably accessed after being written in the system for the first time. Therefore, under the condition that the file can be frequently accessed, the replica redundancy mechanism can effectively realize load balancing and improve access efficiency. Under the condition that the file is not frequently accessed after being written in the system, the fault-tolerant mode will be rapidly adjusted as the erasure code fault-tolerant mechanism according to the self-adaption adjustment mechanism of the system.
2. A file access frequency table is maintained in the system. When a file is accessed and the file name is included in the frequency table, the access frequency of the file will be increased by 1. When the file access frequency reaches to the designated number, the file will be regarded as a frequently accessed file and the fault-tolerant mode of the file will be changed as the replica redundancy fault-tolerant mechanism.
3. When a file is accessed but the file name is not included in the frequency table or the frequency table is fully filled, it is necessary to firstly compare the access time to frequency ratios of the files in the table to find the file record with maximum access time to frequency ratio (minimum access frequency) and remove it out of the frequency table, then add the information of the accessed file into the access frequency table, set the access frequency as 1 and record the time stamp of the file for entering the file table.
4. When a file is accessed but the file name is not included in the frequency table and the frequency table is not fully filled, it is necessary to add the information of the accessed file into the access frequency table, set the access frequency as 1 and record the time stamp of the file for entering the file table.
5. For the file removed out of the frequency table, in order to avoid the frequent calculation for converting the erasure code mechanism, it is necessary to judge whether the removed file enters the frequency table again during the next frequency table scanning process. If the removed file is not included in the frequency table, then the fault-tolerant mechanism shall be changed as the erasure code fault-tolerant

mechanism. If the removed file is included in the frequency table, the fault-tolerant mechanism shall not be changed.

6. The file frequency table shall be periodically managed. According to the time stamp order, the file with the access frequency more than 1 and less than the designated file access frequency shall be removed out of the access frequency table in order to control the capacity of the access frequency table. Firstly, it is necessary to traverse the frequency table and find the file with the access time in the frequency table exceeding the access time threshold value in the file information with the fault-tolerant mechanism still as the erasure code fault-tolerant mechanism; then, it is necessary to remove the eligible file information from the frequency table, wherein the access time threshold value can be set as two or three times of the frequency table clearing interval, namely: if the fault-tolerant mechanism of the file is still not changed after two ~ three clearing operations of the frequency table and the access frequency thereof is not high, then there is no need to statistically collect the access frequency information thereof and the file shall be removed.

4.2. Implementation of SFMAF Erasure code Fault-Tolerant Mechanism

Block Info information array field `checksum_blks` is added in `INodeFile` class in this article in order to realize the erasure code fault-tolerant mechanism. For finding the check data block, it is necessary to find the instance of the corresponding `BlockInfo` through the `checksum_blks` in `INodeFile` class so as to communicate with the selected data node to obtain relevant data.

One source data block can be used to conveniently find the set of all source data blocks and check data blocks through the reverse researching capability of `BlockInfo`. The modified `INode` class diagram is as shown in Figure 2.

Firstly, the file is divided into k source data blocks and SFMAF also includes k check data blocks. After the file is successfully written in the system, when the data blocks are invalid, the system has three data block recovery conditions:

1. The source data blocks of the file are discovered to have invalid blocks but the check data blocks contain no invalid blocks. In such case, the check data blocks are not lost, and the “decoding” algorithm can be adopted for the erasure code to recalculate the source data blocks. Specifically, after invalidation confirmation, the name node obtains the data node map of the corresponding k check data blocks according to “data block - data node list” map and then encapsulates the map in the “decoding” command to be sent to a certain data node. After receiving the command, the data node will decode the lost blocks to recover the invalid data blocks.
2. The source data blocks of the file contain no invalid blocks but the check data blocks are discovered to have invalid blocks. In such case, k data blocks are not lost and the “decoding” algorithm can be directly implemented for the erasure code to recalculate the check data blocks. Specifically, after invalidation confirmation, the name node obtains the data node map of the corresponding k source data blocks according to “data block - data node list” map and then encapsulates the map in the “decoding” command to be sent to a certain data node. After receiving the command, the data node will decode the lost blocks to recover the invalid data blocks.
3. The source data blocks and the check data blocks of the file are discovered to have invalid blocks. In such case, the precondition of the data recovery is that the sum of the number of the valid source data blocks and the valid check data blocks is equal to or more than the number of the source data blocks of the file (namely k). Under the precondition that the above condition is met, the name node will select k data blocks from the valid source data blocks and the valid check data blocks, then obtain the data node map of the corresponding k source data blocks according to “data block - data node list” map and then encapsulate the map in the “decoding” command to be

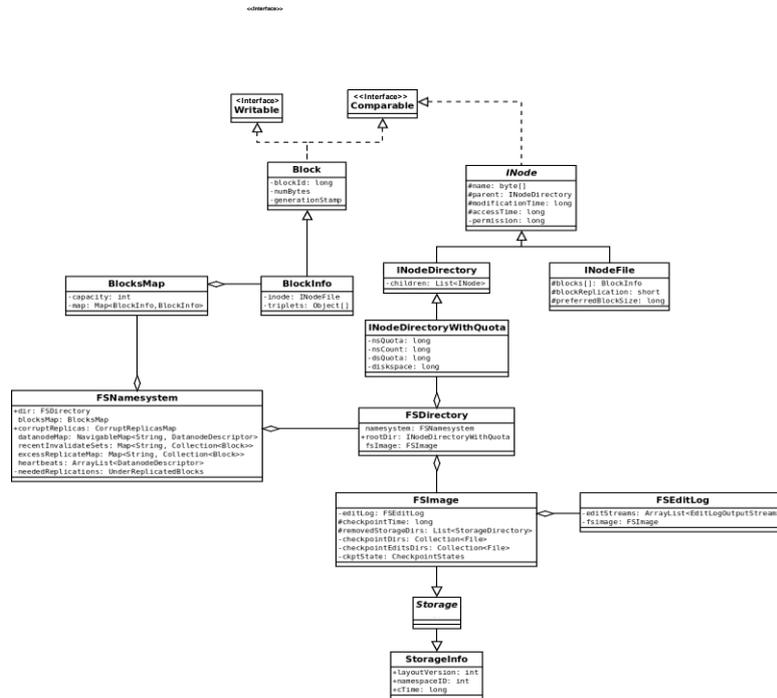


Figure 3. FSNamesystem Class Diagram

5. Experiment and Analysis

The network topology is as shown in Figure 4. The experiment needs 8 servers, 2 interchangers and 1 notebook computer, wherein 2 servers are taken as the name node and the secondary name node, other 6 servers are taken as the cluster data nodes, and the notebook computer is taken as the client terminal for system access and operation. The configuration of the name node server is as follows: CPU dominant frequency 2.4GHz, memory 2GB, hard disk 60GB; the configuration of the secondary name node server is as follows: CPU dominant frequency 2GHz, memory 1GB, hard disk 120GB.

24h continuous file access and writing operations are executed in the cloud storage systems respectively adopting SFMAF and complete replica redundancy mechanism for simulation and comparison. In SFMAF, the parameter is set as num=5, the capacity of the file access frequency table is set as 1,000 records, and the file replica number is set as 3; in the complete replica redundancy mechanism, the file replica number is set as 3. Meanwhile, OSS (Open Source Software) Ganglia is adopted to monitor the memory usage of the servers, CPU occupation rate and network throughput. Specifically, the performance index comparison results are as shown in Figures 5~8, the memory usage rate comparison diagram is as shown in Figures 5~6, CPU occupation rate comparison diagram is as shown in Figure 7, and the comparison of the transmission flow of the internal system data is as shown in Figure 8.

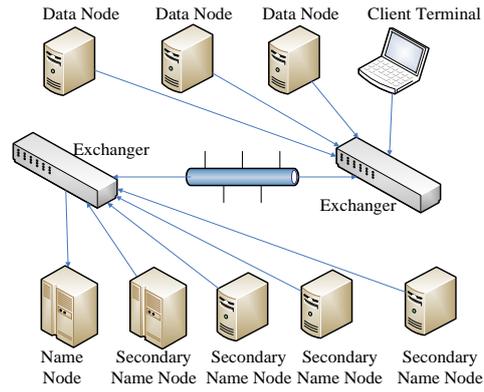


Figure 4. Network Topology

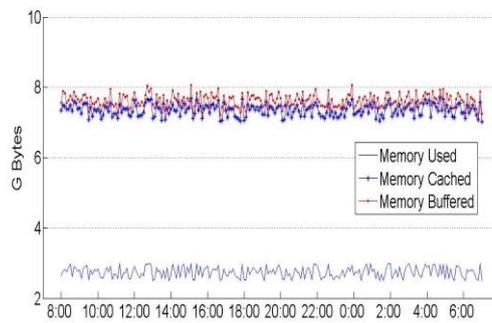


Figure 5. Memory Usage in Complete Replica Redundancy Mechanism

Figure 6. Memory usage in SFMAF Mechanism

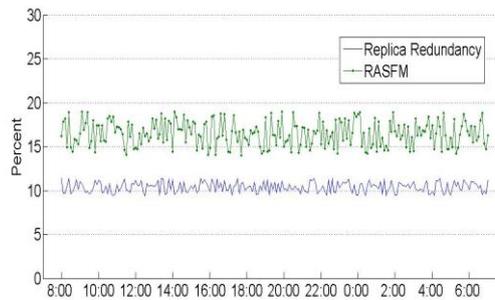


Figure 7. CPU Occupation Rate Comparison Diagram

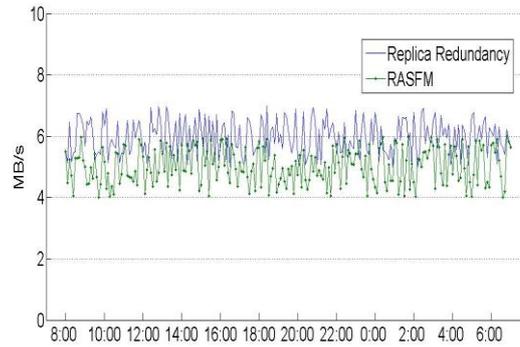


Figure 8. Comparison of Transmission Flow of Internal System Data

According to Figures 5 and 6, the memory usage of the system adopting SFMAF mechanism, especially the memory cache usage, is higher than that of the system adopting the replica redundancy mechanism. According to Figure 7, the CPU occupation rate of the file system adopting SFMAF mechanism is slightly higher than that of the file system adopting the replica redundancy mechanism. According to Figure 8, the transmission flow of the internal data of the file system adopting SFMAF mechanism is less than that of the system adopting the replica redundancy mechanism.

According to the above figures, compared with replica redundancy mechanism, SFMAF mechanism has following differences: ① the system needs more cache resources to store the matrix data used for erasure code calculation. ② In SFMAF, the check data blocks of the erasure code will be calculated after the file is written into the file system, and CPU occupation rate is accordingly increased along with the increase of the calculation quantity. ③ The transmission flow of the internal data of the file system adopting SFMAF mechanism is less than that of the file system adopting the replica redundancy mechanism, thus indicating that the replica quantity of the system adopting SFMAF mechanism is less than that of the system adopting replica redundancy mechanism.

In conclusion, when CPU occupation rate and memory usage are increased in an acceptance range, the replica quantity of the system adopting SFMAF mechanism is less than that of the system adopting the replica redundancy mechanism, thus reducing the transmission flow of the internal system data.

6. Conclusion

For the proposed algorithm SFMAF, the replica redundancy mechanism is adopted for the frequently accessed files and RS erasure code fault-tolerant mechanism is adopted for the rarely accessed files. SFMAF mechanism has its own advantages, but also has some defects, and the further research shall be mainly focused on the following two aspects: 1. In different environments, it is necessary to test the influence of the capacity of the file access frequency table and the parameter num on the algorithm in order to set the optimum frequency table capacity for the algorithm. 2. The erasure code algorithm needs to consume more calculation resources and memory resources, so it is necessary to improve the erasure code algorithm to improve the performance of the proposed algorithm.

References

- [1] J Hu and Z Gao. "Distinction immune genes of hepatitis-induced hepatocellular carcinoma[J]". *Bioinformatics*, vol. 28, no. 24, (2012), pp. 3191-3194.
- [2] J Yang, B Chen, J Zhou, "A Low-Power and Portable Biomedical Device for Respiratory Monitoring with a Stable Power Source[J]". *Sensors*, vol. 15, no. 8, (2015), pp.19618-19632.

- [3] G Bao, L Mi, Y Geng, K Pahlavan, "A computer vision based speed estimation technique for localizing the wireless capsule endoscope inside small intestine", 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Aug. (2014).
- [4] X Song, Y Geng, "Distributed community detection optimization algorithm for complex networks", Journal of Networks, vol. 9, no. 10, Jan. (2014), pp. 2758-2765.
- [5] D Jiang, X Ying, Y Han. "Collaborative multi-hop routing in cognitive wireless networks [J]". Wireless Personal Communications, (2015), pp. 1-23.
- [6] Z Lv, A Halawani, S Feng. "Multimodal hand and foot gesture interaction for handheld devices[J]". ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 11, no. 1s, (2014), p. 10.
- [7] G Liu, Y Geng, K Pahlavan, "Effects of calibration RFID tags on performance of inertial navigation in indoor environment", 2015 International Conference on Computing, Networking and Communications (ICNC), Feb. (2015).
- [8] J He, Y Geng, Y Wan, S Li, K Pahlavan, "A cyber physical test-bed for virtualization of RF access environment for body sensor network", IEEE Sensor Journal, vol. 13, no. 10, Oct. (2013). pp. 3826-3836.
- [9] W Huang, Y Geng, "Identification Method of Attack Path Based on Immune Intrusion Detection", Journal of Networks, vol. 9, no. 4, Jan. (2014), pp. 964-971.
- [10] X Li, Z Lv, J Hu. "XEARTH: A 3D GIS Platform for managing massive city information[C]". Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2015 IEEE International Conference on. IEEE, (2015), pp. 1-6.
- [11] J He, Y Geng, F Liu, C Xu, "CC-KF: Enhanced TOA Performance in Multipath and NLOS Indoor Extreme Environment", IEEE Sensor Journal, vol. 14, no. 11, Nov. (2014), pp. 3766-3774.
- [12] N Lu, C Lu, Z Yang, Y Geng, "Modeling Framework for Mining Lifecycle Management", Journal of Networks, vol. 9, no. 3, Jan. (2014), pp. 719-725.
- [13] Y Geng, K Pahlavan, "On the accuracy of rf and image processing based hybrid localization for wireless capsule endoscopy", IEEE Wireless Communications and Networking Conference (WCNC), Mar. (2015).
- [14] X Li, Z Lv, J Hu, "Traffic management and forecasting system based on 3d gis[J]", Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on, (2015), pp. 991-998.
- [15] S Zhang, H Jing. "Fast log-Gabor-based nonlocal means image denoising methods[C]". Image Processing (ICIP), 2014 IEEE International Conference on. IEEE, (2014), pp. 2724-2728.
- [16] D Jiang, Z Xu, Z Chen, "Joint time-frequency sparse estimation of large-scale network traffic[J]. Computer Networks, 2011, 55(15): 3533-3547. Jinyu Hu, Zhiwei Gao and Weisen Pan. Multiangle Social Network Recommendation Algorithms and Similarity Network Evaluation[J]. Journal of Applied Mathematics, (2013).
- [17] J Hu and Z Gao. "Modules identification in gene positive networks of hepatocellular carcinoma using Pearson agglomerative method and Pearson cohesion coupling modularity[J]". Journal of Applied Mathematics, 2012 (2012).
- [18] Z Lv, A Tek, F Da Silva. "Game on, science-how video game technology may help biologists tackle visualization challenges[J]". PloS one, vol. 8, no. 3, (2013), p. 57990.
- [19] T Su, W Wang, Z Lv, "Rapid Delaunay triangulation for randomly distributed point cloud data using adaptive Hilbert curve[J]". Computers & Graphics, no. 54, (2016), pp. 65-74.
- [20] S Zhou, L Mi, H Chen, Y Geng, "Building detection in Digital surface model", 2013 IEEE International Conference on Imaging Systems and Techniques (IST), Oct. (2012).
- [21] J He, Y Geng, K Pahlavan, "Toward Accurate Human Tracking: Modeling Time-of-Arrival for Wireless Wearable Sensors in Multipath Environment", IEEE Sensor Journal, vol. 14, no. 11, Nov. (2014), pp. 3996-4006.
- [22] Z Lv, A Halawani, S Fen, "Touch-less Interactive Augmented Reality Game on Vision Based Wearable Device[J]", Personal and Ubiquitous Computing, vol. 19, no. 3, (2015), pp. 551-567.
- [23] G Bao, L Mi, Y Geng, M Zhou, K Pahlavan, "A video-based speed estimation technique for localizing the wireless capsule endoscope inside gastrointestinal tract", 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Aug. (2014).
- [24] Di Zeng, Y Geng, "Content distribution mechanism in mobile P2P network", Journal of Networks, 9(5), 1229-1236, Jan. (2014).
- [25] W Gu, Z Lv, M Hao. "Change detection method for remote sensing images based on an improved Markov random field[J]". Multimedia Tools and Applications, (2015), 1-16.
- [26] Z Chen, W Huang, Z Lv, "Towards a face recognition method based on uncorrelated discriminant sparse preserving projection[J]". Multimedia Tools and Applications, (2015), pp. 1-15.
- [27] M Zhou, G Bao, Y Geng, B Alkandari, X Li, "Polyp detection and radius measurement in small intestine using video capsule endoscopy", 2014 7th International Conference on Biomedical Engineering and Informatics (BMEI), Oct. (2014).
- [28] G Yan, Y Lv, Q Wang, Y Geng, "Routing algorithm based on delay rate in wireless cognitive radio network", Journal of Networks, vol. 9, no. 4, Jan. (2014), pp. 948-955.

- [29] Y Lin, J Yang, Z Lv. "A Self-Assessment Stereo Capture Model Applicable to the Internet of Things[J]". *Sensors*, vol. 15, no. 8, (2015), pp. 20925-20944.
- [30] K Wang, X Zhou, T Li, "Optimizing load balancing and data-locality with data-aware scheduling[C]". *Big Data (Big Data)*, 2014 IEEE International Conference on. IEEE, (2014), pp. 119-128.
- [31] L Zhang, B He, J Sun. "Double Image Multi-Encryption Algorithm Based on Fractional Chaotic Time Series [J]". *Journal of Computational and Theoretical Nanoscience*, no. 12, (2015), pp. 1-7.
- [32] T Su, Z Lv, S Gao. "3d seabed: 3d modeling and visualization platform for the seabed[C]". *Multimedia and Expo Workshops (ICMEW)*, 2014 IEEE International Conference on. IEEE, (2014), pp. 1-6.
- [33] Y Geng, J Chen, R Fu, G Bao, K Pahlavan, "Enlighten wearable physiological monitoring systems: On-body rf characteristics based human motion classification using a support vector machine", *IEEE transactions on mobile computing*, vol. 1, no. 1, Apr. (2015), pp. 1-15.

Authors



Liu Hongqing. Liu Hongqing, he was born in Shaoyang, Hunan, China, 1973-05. He obtained a master's degree in computer science and technology, Chongqing University. He is a lecturer in the logistics information system of Hunan Vocational College of Modern Logistics. His research interests are mainly in the field of computer software, computer network and so on. He has published several academic papers in the field of research, and has written several books on computer science.



Huang Yan. Huang Yan, she was born in Changsha, Hunan, China, 1974-09. He received a bachelor's degree in computer science and technology, Xiangtan University. She is a lecturer in the logistics information system of Hunan Vocational College of Modern Logistics. Her research interests are mainly in the field of computer software, Internet of things, and so on. She has published several academic papers in the field of research.

