

# A Load Balancing Strategy Based on Multiple Metrics in Spatial Image File Access Process in Hadoop

Lina Yuan<sup>1</sup>

*South China Institute of Software Engineering.GU, Guangzhou University  
Guangzhouconghua 510990, P.R. China  
E-mail: 18971656@qq.com*

Yuanyong Feng<sup>a</sup>, Binbin Li, Fufang Li<sup>23b</sup>

*School of Computer Science and Educational Software, Guangzhou University  
Guangzhou 510006, P.R. China  
E-mail:<sup>a</sup>ComerFeng@gmail.com,<sup>b</sup>lffgz@163.com*

To solve the problem of load imbalance when accessing spatial image files in HDFS, we present a loading capacity model based on multiple attributes. In our model, five attributes, including disk space load capacity, CPU processing capacity, memory processing capacity, disk read-write processing capacity and bandwidth, are introduced to measure the load balancing of each node. Then we used the objective weighting method of variation coefficient method to process experiment data, calculated the weight coefficient of the five attributes and obtained a load capacity model. By this way, the nodes can calculate the actual load ability of their own, which will help improve the efficiency of the load balance when accessing spatial image files in HDFS. To validate the proposed model, a large number of spatial image files are adopted to be accessed in our experiments. Experiment results show that, when accessing the spatial image files in HDFS, , better load balance ability, system performance and efficiency can be obtained by using the strategy of the proposed multi-attribute load balance model.

*CENet2017  
22-23 July 2017  
Shanghai, China*

<sup>1</sup>Speaker

<sup>2</sup>Corresponding Author

<sup>3</sup>This work has been co-supported by: Natural Science Foundation of China under Grant No. 61472092; Guangdong Provincial Science and Technology Plan Fund with grant No. 2013B010401037; Natural Science Foundation of Guangdong Province under Grant No. S2011040003843; GuangZhou Municipal High School Science Research Fund under grant No. 1201421317; State Scholarship Fund by China Scholarship Council under Grant No. [2013]3018-201308440096; and Yuexiu District Science and Technology Plan Fund of GuangZhou City with grant No. 2013-CY-005.

### 1.Introduction

The rising wave of smart city information construction and the rapid development of big data technology bring about various heterogeneous spatial image files, which need to be uploaded to the big data center server for distributed storage management[1]. With rapid increase of such large amount of heterogeneous spatial image files, the load unbalance may be induced among the server nodes; but the default load balancing strategy only uses a single disk space utilization index to measure the load of each node. Someone puts forward three factors (including using CPU utilization, memory utilization and disk I/O access rates) to describe the node load pressure while others present several properties to measure the load situation of each node and adopt the analytic hierarchy process method (a part of subjective weighting method) to calculate the weight of each factor[2-6]. But the spacial image files have a large amount, demand high prcision, need more disk, CPU, memory, bandwidth and disk I/O access rates than other ordinary images in the access procedure, and also require more objective process from many aspects. Therefore, the default load balancing strategy or three factors (including using CPU utilization, memory utilization and disk I/O access rates) strategy can not accurately reflect the actual Workload, and there are still some problems (such as low cluster performance and nodes load imbalance). In addition, the analytic hierarchy process method is subjective so that the result will be easily influenced by human factors. In this sense, the method is not suitable for the spacial image file to calculate the weight of each index. To solve the problem, we put forward an improved HDFS load balancing strategy, in which we used five attribute indexes to measure the load capacity of each node and adopted variation coefficient method to confirm the weight of each attribute index. As a result, we can objectively reflect various nodes load balance ability, balance the node load and improve the efficiency of cluster.

### 2.HDFS Default Load Balancing Strategy

Hadoop Distributed File System (HDFS) is the foundation of data storage in the distributed storage architecture based on Hadoop. HDFS adopts the master/slave architecture to manage the whole distributed file system. A HDFS cluster is composed of a namenode and several datanodes[3]. The HDFS system structure in Hadoop2.x is shown as Figure 1.

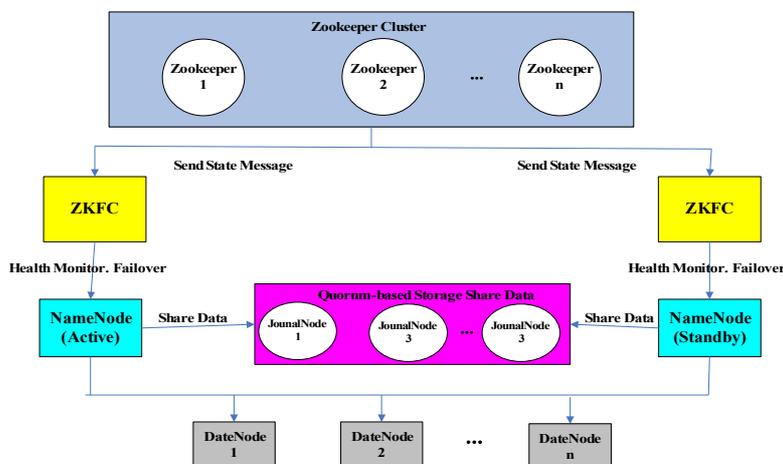
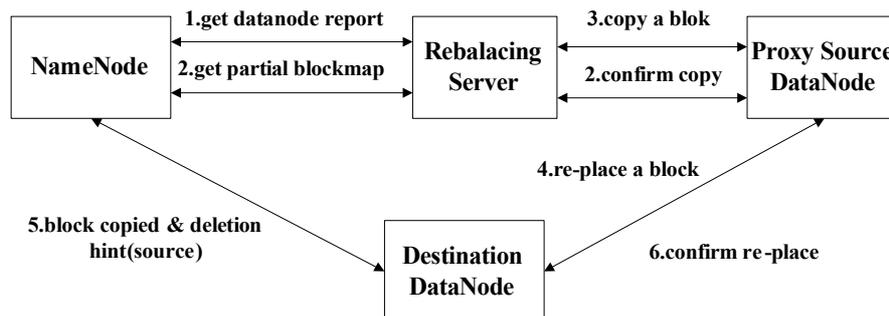


Figure 1:HDFS System Structure

### 3.Default Load Balancing Strategy of HDFS

In HDFS system, when a node is added or deleted in the cluster, or the hard disk storage in the machine at a certain node reaches the saturation value, the disk utilization unbalance between the machines may occur easily. In case of HDFS load unbalance, the data load balance adjustment should be performed on the HDFS, i.e. the data storage distribution of the machine at the nodes should be adjusted[4]. Therefore, the data can be evenly distributed on the DataNode, the IO performance can be balanced and the hotspots can be avoided. The HDFS provides a default load balancing strategy and its load balancing principle is shown in Figure 2.



**Figure 2:**Default Load Balancing Principle of HDFS

The Re-balance Server firstly requires the NameNode to produce the DataNode data distribution analysis report and acquire the disk using condition of each DataNode; the load balancer summarizes the data distribution condition to be moved and calculates a specific data block migration route map to ensure the shortest route within the network; the data block migration task starts, the proxy source datanode copies a data block to be moved on the target DataNode, and deletes the original data block; the target DataNode confirms with the proxy source datanode that the data block migration is completed; the proxy source datanode confirms with the load balancer that the data block migration is completed. Then the process is executed continuously till the cluster reaches the data balance standard.

However, the default load balance strategy does not have high adaptability in the spatial big data image file access process. The HDFS default load balancing strategy measures the load only via a single index of disk space usage rate, but the spatial big data image files are large in number and high in accuracy requirement, and the CPU, memory and bandwidth occupied in the access process are larger than those of common images. Therefore, the default load balancing strategy only adopts the disk space usage rate to measure the load and cannot accurately reflect the actual workload of the datanodes. The problems of cluster load unbalance and low performance still exist[5].

### 4.Load Balancing Strategy based on Multiple Attributes

The spatial big data image files are large in number, high in accuracy and frequent in access, therefore, the I/O access frequency is relatively high in the access process. At meanwhile, in the spatial big data image file access process, a small part of image files frequently accessed should be stored in a local cache and memory, and the CPU and the internal memory are also required. In addition, the spatial big data image files have high requirement on accuracy, and more bandwidth is required during the network transmission. However, the HDFS default load balancing strategy only adopts single index of disk space usage rate to measure the

node load. In order to further improve HDFS cluster load balancing effect in the spatial big data image file access process and fully consider the datanode performance, this article proposed a load balancing strategy based on multiple attributes for the shortcoming of the HDFS default load balancing strategy. In addition to the disk space usage rate, the strategy further considers the CPU processing capacity of the nodes, the disk read-write speed, the memory processing capacity and the network bandwidth.

#### 4.1 Performance Measuring Index

By analyzing the load performance of the Datanode, the article selected the optimum node to save the image files according to the weight obtained in analysis to further achieve load balance of the cluster [6]. There are mainly five load performance indexes to measure the nodes as follows:

##### (1) Disk space load capacity

The larger the node disk space surplus is, the higher the surplus ratio and the node disk space load capacity will be. The system firstly saves the data in the datanode with higher disk space load capacity optimally to enable the integral disk load of the cluster to be balanced. The disk space surplus ratio is shown by  $P_{disk}$ ,  $P_{disk} = 1 - disk_{use}$ .

##### (2) CPU load capacity

The lower the node CPU usage rate is, the lower the CPU load will be and the higher the node CPU load capacity will be. The system will give preference to the datanode with higher CPU load capacity to save the data. CPU load capacity  $P_{cpu} = 1 - CPU_{use}$ , and  $CPU_{use}$  is the ratio of the CPU execution system non-idle thread time in CPU total operation time.

##### (3) Memory load capacity

The lower the node memory occupancy is, the lower the node memory load will be and the higher the memory load capacity will be. The system will give preference to the datanode with higher memory load capacity to save data. The memory load capacity can be expressed as  $P_{men} = 1 - MEM_{use}$ , in which,  $MEM_{use}$  is the percentage of the memory used by the datanode operation system thread and the datanode total physical memory.

##### (4) Disk read-write capacity

The disk read-write capacity has direct effects on the task execution time, and the system will give preference to the datanode with high read-write capacity. The disk read-write capacity is an important index for the data placement strategy, shown by the parameter T;  $T_w$  is the disk write speed,  $T_r$  is the disk read speed, and  $\alpha$  is the ratio of the write operation in the read-write operation. The disk read-write capacity can be expressed as:  $P_{IO} = \alpha T_w - (1 - \alpha) T_r$ .

##### (5) Bandwidth surplus ratio

The network bandwidth will also affect the cluster load balance, and the bandwidth usage rate is the ratio of the actually used cluster operation bandwidth of all nodes in the cluster and the total bandwidth. The lower the usage rate is, the more free the cluster will be; the higher the usage rate is, the busier the cluster will be. Taking two time points M1 and M2 randomly, and the total data flow in the period can be calculated,  $S = D_i * (M_2 - M_1) (i \in [1, n])$ , wherein  $D_i$  is the network bandwidth of  $i$  node and  $n$  is the number of datanodes in the cluster. In a

certain period, the network bandwidth of datanode  $i$  surplus ratio  $P_{Bandw} = 1 - S_{dj}/S$ , and  $S_{dj}$  is the actual data flow.

#### 4.2 Load Function Model

The linear weighting method was used to perform load capacity function modeling. Due to the fact that the measurement indexes are of importance on the total target to different degrees, the coefficients can be set for the measures respectively according to the importance degree, and the measurement index values with coefficients can be added to obtain the value of the total target[7]. Therefore, the improved load capacity function based on performance index can be described as follows:

$$P_i = A_1 * P_{disk} + A_2 * P_{cpu} + A_3 * P_{mem} + A_4 * P_{I/O} + A_5 * P_{Bandw} \quad (\sum A_i = 1) \quad (4.1)$$

In the above formula,  $P_i$  is the load comprehensive surplus capacity,  $A_1$ ,  $A_2$ ,  $A_3$ ,  $A_4$  and  $A_5$  are respectively represent full-value coefficients of five measurement indexes for showing importance degree of the load measurement indexes on the datanode load surplus; the function weight vector  $\omega = (A_1, A_2, A_3, A_4, A_5)$ .

#### 4.3 Weight vector calculation

According to different weight production methods, the multi-index comprehensive evaluation method can be divided into two main classes of subjective weight evaluation method and objective weight evaluation method, wherein the subjective weight evaluation method is characterized in that the expert performs subjective judgment based on experiences in a qualitative method to acquire the weights and then performs comprehensive evaluation on the indexes, such as analytic hierarchy process method, comprehensive evaluation method, fuzzy evaluation method, index weighting method and efficiency coefficient method. The method is random and subjective to some extent and high in interference by human factors so that it is difficult to acquire accurate evaluation when there are too many evaluation indexes[8]. The objective weight evaluation method determines the weight for comprehensive evaluation according to the relation between the indexes or the variation relation of the indexes, such as entropy evaluation method, neural network analysis, principal component analysis and variable coefficient method. The method fully considers mutual relation between the indexes and determines the weights according to the initial information of the indexes, and accurate value of the evaluation result can be reached. The article mainly adopted the variable coefficient method to calculate the initial value of the weight vector.

The variable coefficient method is an objective weight method that directly utilizes information contained by the information to acquire the weights of the indexes by calculation [9]. The method is basically characterized in that it is more difficult to achieve the index with higher value difference in the index evaluation system, and such index can better reflect the difference of the evaluated unit[10]. Due to the fact that the indexes in the variable index system have different dimensions, it is inappropriate to compare the difference degree directly. To eliminate the effect of different dimensions of the evaluation indexes, the variable coefficient of the indexes should be adopted to measure the difference degree of the index values. The variable coefficient formulas of the various indexes are as follows:

$$V_i = \frac{\sigma_i}{\bar{X}_i} (i=1,2,\dots,n) \tag{4.2}$$

Wherein  $V_i$  is the variation coefficient of the  $i$ th index, also called standard deviation coefficient;  $\sigma_i$  is the standard deviation of the  $i$ th index and  $\bar{X}_i$  is the average of the  $i$ th index. The weights of the indexes are:

$$W_i = \frac{V_i}{\sum_{i=1}^n X_n} \tag{4.3}$$

Calculate the weight coefficient of simulation experiment platform by adopting the Hadoop cluster, in total, including 16 PCs units, the specific hardware parameters as Article 4.1 experimental environment, and use 10G, 20G spatial image file respectively upload, download and other various operations in HDFS. As there is too much data, there is no detailed list, just a list of the steps and the results. In the process of operation, check and record every operation of each machine in the disk usage, CPU usage, memory usage, bandwidth usage, etc., and then according to Article 3.1 in respect of the performance measure as referred to in the disk space, CPU load capacity formula, CPU load capacity formula, calculate the disk load capacity, CPU processing capacity, memory processing capacity and disk read-write capacity, Bandwidth surplus ratio and so on five performance indicators in each operation. Figure out the mean and the standard deviation, to obtain the data in Lines 1 and 2 of Table 1:

Then Based on the above average and the standard deviation data, the variable coefficients of the indexes were calculated via the variable coefficient formula(  $V_i = \frac{\sigma_i}{\bar{X}_i}$  ) according to the above average and standard deviation data value, and all the variable coefficients were summed to obtain the data in the third line of Table 1. Finally the weights of the indexes were acquired

via the variable coefficient weight calculation formula(  $W_i = \frac{V_i}{\sum_{i=1}^n X_n}$  ) to obtain data in the

fourth line of Table 1.

Thus, the final load comprehensive surplus capacity evaluation function could be obtained as follows:

$$P_i = 0.36 * P_{disk} + 0.18 * P_{cpu} + 0.12 * P_{mem} + 0.3 * P_{I/O} + 0.04 * P_{Bandw} \tag{4.4}$$

Index	Disk load capacity	CPU processing capacity	Memory processing capacity	Disk read-write capacity	Bandwidth surplus ratio	Sum
Average	0.55	0.52	0.49	0.59	0.71	-
Standard deviation	0.22	0.11	0.06	0.20	0.03	-
Variable coefficient	0.40	0.21	0.13	0.34	0.04	1.12
Weight	0.36	0.18	0.12	0.30	0.04	1.00

**Table 1:** Index Average and Standard Deviation

### 5.Experiment Results and Analysis

This article proposed a load balancing strategy based on multiple metrics in Spatial Image File Access Process[11], verified the improved algorithm and compared it with the default algorithm to test the actual effect of the improved strategy.

#### 5.1Experiment Environment

The Hadoop cluster set on the simulation experiment platform includes 3 racks: Rack1, Rack2 and Rack3. Each rack includes 5 Datanodes. The simulation Hadoop experiment platform version is Hadoop2.6.5, and JDK version is jdk1.7.0\_45. The experiment uses the upload and download of the spatial image files on HDFS to show the performance and efficiency.

The Hadoop cluster of the simulation experiment platform includes 16 PCs in total, with the network topology as shown in Figure 3.

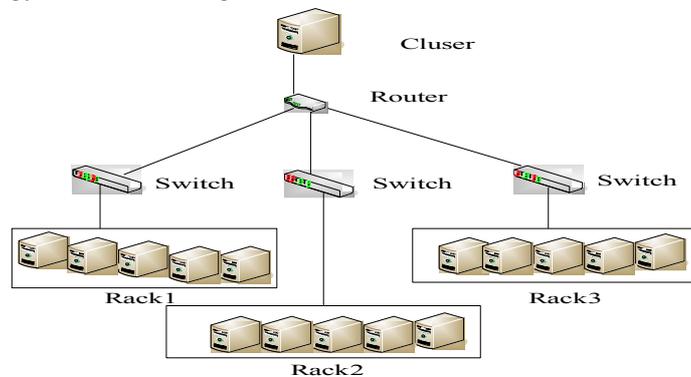


Figure 3:Network Topology

The main hardware parameter indexes of the 16 sets of datanodes are shown as Table 2. In Table 2, all the datanode bandwidth is 100M.

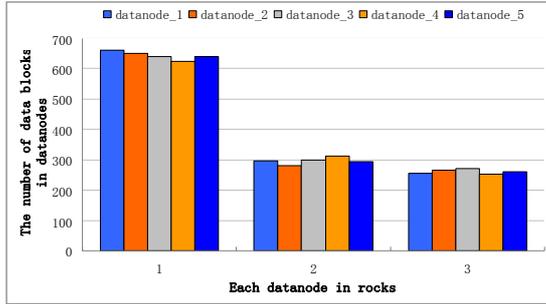
	Node	CPU	Memory	Hard drive capacity	I/O
	Namenode	i5	8G	1T	7200
Rack1	Datanode1	i5	8G	1T	7200
	Datanode2	i5	8G	1T	7200
	Datanode3	i3	4G	500G	7200
	Datanode4	i3	4G	500G	7200
	Datanode5	i3	2G	500G	5400
Rack2	Datanode1	i5	4G	1T	7200
	Datanode2	i5	4G	1T	7200
	Datanode3	i3	4G	500G	7200
	Datanode4	i3	4G	500G	7200
	Datanode5	i3	2G	500G	5400
Rack3	Datanode1	i3	2G	500G	7200
	Datanode2	i5	4G	1T	7200
	Datanode3	i3	4G	500G	7200
	Datanode4	i3	4G	500G	7200
	Datanode5	i3	2G	500G	5400

Table 2: Hardware Parameter List

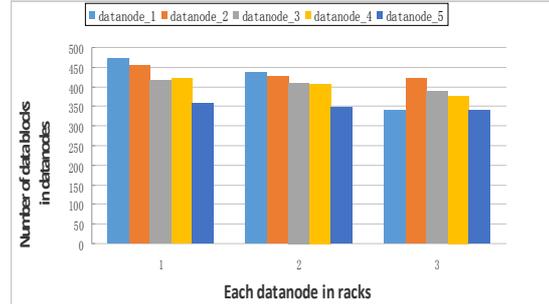
#### 5.2Experiment Result Analysis

In the simulation experiment, the system default load balancing strategy and the load balancing strategy based on multiple attributes were compared through the client. Image files of 6000 data blocks were submitted to the server through the client. According to statistics, the data block distribution condition is shown in Figure 4 when adopting the system default load

balancing strategy. The data block distribution condition is shown as Figure 5 when the load balancing strategy based on multiple attributes is adopted.



**Figure 4:**Data Distribution Condition of Default strategy

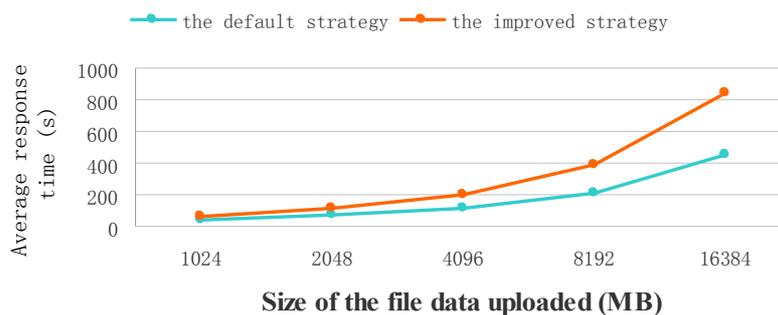


**Figure 5:**Data Distribution Condition of Load Balancing Strategy based on Multiple Attributes

The results in the Figure 4 showed that if the HDFS adopted the default load balancing strategy, most data were stored in Rack 1 and the average storage data block reached 600, but in Rack2 and 3, the average storage data block was less than 300. In this sense, the datanode performance and load difference were not considered when the spatial big data image files were accessed and the files were evenly placed on the datanodes of the racks.

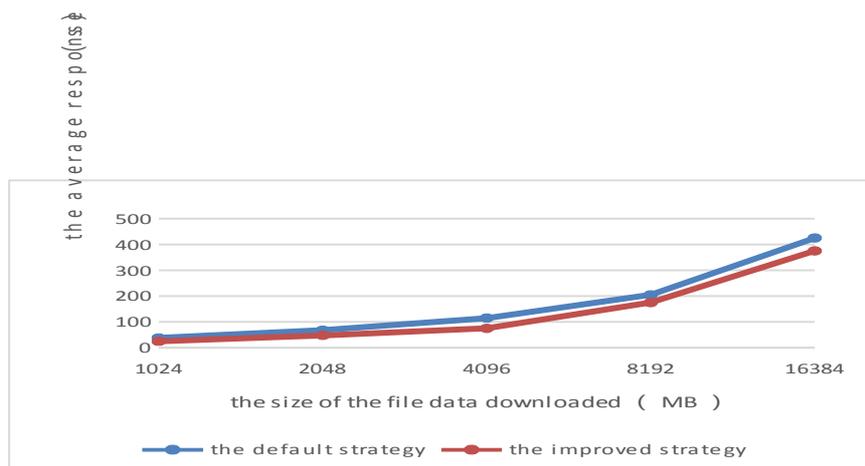
Figure 5 shows that using the improved strategy, no more data is stored in Rack 1, but according to the performance load index in each datanode of each rack, relative datanode load ability of the slightly store more data, relatively weak load node stores less data. It showed that the improved load balancing strategy based on multiple attributes better achieved the system load balance.

In addition, the image files with different data volumes were uploaded from the client to the server under the above two strategies, and the average response time comparison was shown in Figure 6.



**Figure 6:**Uploaded File Average Time Comparison

Then the image files with different data volumes were downloaded from the server to the client by means of the above two strategies, and the average response time comparison was shown as Figure 7.



**Figure 7:**Downloaded File Average Time Comparison

In Figure 6 and Figure 7, the load balancing strategy based on multiple strategies has reduced the file uploading and downloading time obviously and improved the cluster working execution efficiency.

## 6. Conclusion

For the shortcoming of the default load balancing strategy which only adopts the disk space usage rate to measure the load balance, the problem of using subjective empowerment evaluation method of analytic hierarchy process method to calculate the weight of each factor in a literature has been improved with the improvement strategy by such five indicators as disk space load capacity, CPU processing capacity, memory processing capacity, disk read-write processing capacity and bandwidth, etc. to measure the load balancing of various nodes, each node can more objectively reflect the actual load capacity. Considering the objectivity that the spatial image file operations need more variable coefficient method and the experimental data, calculate the above five measure of the relative weight coefficient, and obtain the load function model. The experiment results have confirmed the improved strategy based on multiple attributes is better than the default policy in realizing the load balance of each node, and improving work efficiency to a certain extent, which makes the spacial image files in HDFS access operation to achieve better load balance of each node.

## References

- [1]S. Y. Liu. ETc. *HDFS load balancing in ophthalmic medical image file access*[J]. Computer Engineering and Applications, 2017, 53 (2) : 253-259. (In Chinese)
- [2]S. J. Li. ETc. *Geo-spatial Big Data Storage Based on NoSql Database*[J]. Geomatics and Information Science of Wuhan University, 2017, 23 (2) : 163-169. (In Chinese)
- [3]Z. X. Sun. ETc. *Massive Data Storage Technology in Cloud Computing* [J]. Journal of Nanjing University of Posts and Telecommunications( Natural Science), 2014, 34 (4) : 13-19. (In Chinese)
- [4]H. R. Wang. ETc. *Design of Massive Data Storage System Based on Hadoop* [J]. BULLETIN OF SCIENCE AND TECHNOLOGY, 2014, 30(9):127-130. (In Chinese)
- [5]B. H. Zhang. ETc. *Scheme of high space efficiency in distributed storage* [J]. Application Research of Computers, 2015, 32(5):1504-1511. (In Chinese)

- [6]C. K. Kang. ETc. *A multi metrics based load balancing algorithm for HDFS* [J]. Journal of Sichuan University (Natural Science Edition), 2014, 51(6):1163-1169. (In Chinese)
- [7]M. L. Lu. ETc. *Load Balancing Strategy Based on CMM Model in HDFS* [J]. Journal of Beijing University of Posts and Telecommunications, 2014, 37(5):20-25. (In Chinese)
- [8]A. P. Xiong. ETc. *A kind of HDFS small files storage strategy based on hybrid index* [J]. Journal of Chongqing University of Posts and Telecommunications ( Natural Science Edition), 2015, 27(1):97-102. (In Chinese)
- [9]H. W. Zhang. ETc. *Research on Data Distribution Strategy in Cloud Storage System Based on Multi-objective Optimization* [J]. Computer Science, 2015, 42(4):44-50. (In Chinese)
- [10]Nihtianov S. *Measuring in the Subnanometer Range: Capacitive and Eddy Current Nanodisplacement Sensors* [J].IEEE Industrial Electronics Magazine, 2014,8(1):6-15
- [11]X. C. Huang. ETc. *HDFS Load Balancer for Video Cloud Storage Service* [J]. Journal of Chinese Computer Systems, 2017, 38(2):293-298. (In Chinese)