

Design and Development of the Mass Image Storage Platform Based on Hadoop

Xiaoqing Zhou

Center of Computing
China West Normal University
Nanchong, China
e-mail: 369656820@qq.com

Zhiyong Zhou

Center of Computing
China West Normal University
Nanchong, China
e-mail: ZhouSun123@163.com

Jiangjing Gou

Center of Computing
China West Normal University
Nanchong, China
e-mail: 449735106@qq.com

Jiaxiu Sun

Colleges of Business
China West Normal University
Nanchong, China
e-mail: Zhouyu12@126.com

Abstract—With fast development and deep appliance of the Internet, the problem of mass image data storage stands out, so the problem of low management efficiency, low storage ability and high cost of traditional storage framework has appeared. The appearance of Hadoop provides a new thought. However, Hadoop itself is not suit for the handle of small files. This paper puts forward a storage framework of mass image files based on Hadoop, and solved the internal storage bottleneck of NameNode when small files are excessive through classification algorithm of preprocessing module and lead-in of high efficiency and first-level of index mechanism. The test manifests that the system is safe, easy to defend and has fine extension quality; as a result, it can reach to a fine effect.

Keywords—Massive Images; Hadoop; Distributed Calculation; Storage Framework

I. INTRODUCTION

With fast development and deep appliance of the Internet, it appears more and more large portal sites, e-business websites and net community. These websites have large picture resource and the number of pictures increases in high speed. Traditional technological framework cannot play an active role in dealing with massive images. How to build a cheap and efficient picture storage management system in the precondition of fulfilling high concurrent access has become a to-be-solved problem.

This paper puts forward a massive image storage module based on Hadoop through analysis of HDFS of Hadoop, research of MapReduce and business demand analysis of picture storage. This system has a framework of Master/Slave and based on HDFS distributed file system of Hadoop. It's hardware is constructed on the common Linux machine clusters. High error tolerance, high response and load balancing can be realized through indoor monitor and it provides outdoor service, what's more, it can also fulfill high concurrency and dependable appliance.

II. OVERALL PLATFORM DESIGN

The system adopts HA framework and smooth expansion which enable the availability and extension of the whole file system. Besides, the system uses flat data organization structure through efficient extended first-level index mechanism, in this way, which can place sequence file and offset position relationship of pictures and files quickly. Storage perfection and stability of every storage node is realized through load balancing and cache system design. The system employs MVC3 layer design after combining nature of different structure of massive image data, distribution and diversity and considering the realization of system organization. As a result, the structure is clearer and the system is easier to extend. This overall platform consists of three layers, including data resource layer, business logic layer, application interface layer. Data resource layer is the basis of whole platform and the fundamental part of cloud storage. Business logic layer can parallel process massive image data and manage the entire platform system configuration. It is the most important part of the cloud storage, but also has technical content. The main function is to coordinate a plurality of storage apparatus underlying business logic layer, to provide a uniform API interface to the upper layer application service and to shield storage apparatus on the lower layer.

III. PLATFORM FUNCTION DESIGN

A. Platform overall function module structure

Considering system's function, the whole system's overall function module structure is presented like Figure 1.

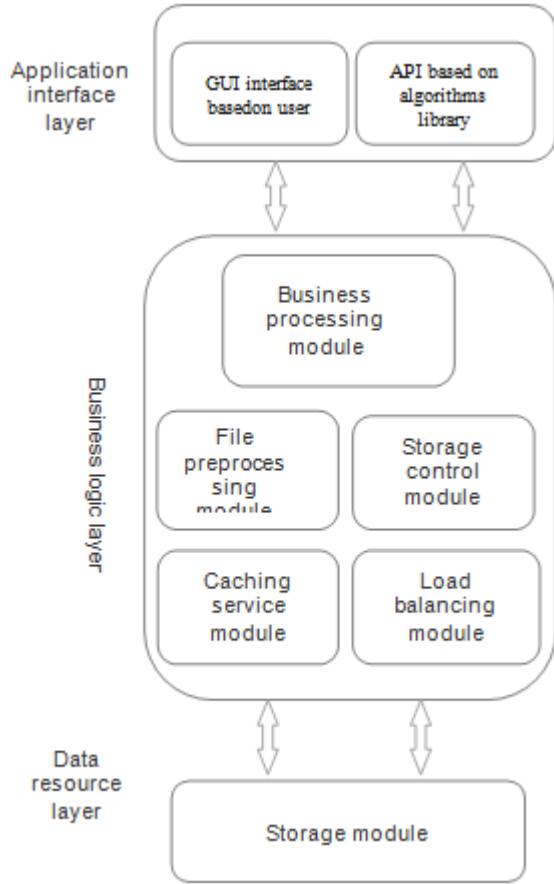


Figure 1. The overall function module structure of the platform

B. Stratified platform module structure analysis

1) Data resource layer module structure

At the bottom is the data storage layer, mainly composed of a storage module. For massive data storage, different data sources provided by different database need to be shielded by storage module to provide database access services, so that the system can meet the requirement of handling massive data storage, as a result, the system has better extensibility and completeness which is ease of management and deployment. A large number of low-cost machines are combined into a cluster through HDFS of Hadoop which provide massive storage capacity.

2) Business logic layer module structure

Business logic layer is the core of the whole system, but also key elements of the system design and development. It uses a distributed database technology and Linux cluster technology, providing the main function of the massive data parallel load storage. The processed data is stored at distributed database of the system, through the massive parallel processing of data in this layer, while also providing management support services to ensure system normal operation. This layer consists of five functional modules.

a) Image file pre-processing module

Image file preprocessing module is to preprocess the image file, file name design and picture metadata

management. Preprocessing module combines the strong correlation files into Sequence File by using the classification algorithm, which greatly reduces the number of files in HDFS. Through the design of the image file name, the image can be easily read.

b) Storage control module

Storage control module's main function is to provide storage management web interface and a unified command, realizing the management of the storage layer nodes via web interface. Hadoop provides opened port 50070 for viewing NameNode information. You can view the current normal Node, Node failures, NameNode log information through this interface.

c) Caching service module

The main purpose of caching service module is to build buffer. The client requests sent to here by cache buffer screening, where we use Redis to build caching system. Redis is similar to traditional Memcached. They both are a <Key/Value> type storage system. Value type of storage it supports is relatively more, including String, List, Set and zset. Push/pop, add\ remove and intersection, union set, difference Set, operations or richer operations are supported by these data types, and these operations are atomic.

d) Business processing module

Business processing module is mainly to solve image processing tasks. As usual, uploaded Internet application pictures need to be processed such as, upload picture as generate thumbnails, upload portrait with Image Segmentation. In the case of bulk upload pictures, if the picture processing focus on the application server, the application server will take up CPU resources, affecting the quality of the application service. Traditional approach is a machine alone does image processing. Such an approach can be decoupled from the application. However, after the emergence of MapReduce, we can process pictures based on MapReduce. Image processing is distributed on the cheap storage machine of image storage and stored directly after processing which not only save hardware resources, but also deal with stress distribution to each node.

e) Load balancing module

Load balancing module is mainly used to solve the stability under high concurrent use of the system. We deploy load balancing for the entire system, using HAProxy of RoundRobin load balancing algorithm, dividing front-end user requests pressure to each Web image server. HAProxy provides high availability, load balancing, and proxy TCP and HTTP-based applications. It is a free, fast and reliable solution. Our requests are sent to different servers through distinction of read and write. Picture read requests are sent to the image server. On the one hand it reads the picture metadata information through cache, on the other hand it hosts image through NameNode.

3) Application interface layer structure

This layer is made up by GUI interface module based on users and API interface module based on calculating ways. This interface module faces towards users for whom providing different kinds of operating applied tools. So users have easy access to huge data storage process. We can provide compiling applied system and assign API in

calculating storage according to some high-level users having higher and more requirements, which can realize needed applied function.

IV. SYSTEM REALIZATION AND RESULT ANALYSIS

A. Software preparation

Operation system: Ubuntu 9.04

Distributed file system: Hadoop0.20.2. The motion of Hadoop needs JDK environment and this paper chooses JDK1.6.0_31, including a NameNode server, a JobTracker server and four DataNodeservers.

Image server: Nginx-0.9.6

Caching software: Redis

Load balancing software: HAProxy

Java environment: JRE6

Java develop instrument: eclipse3.2

The main install process of Hadoop is as follows: first dispose host file, create new users and catalogue, install JDK and dispose environment index; then dispose SSH enrollment without a code, install Hadoop and its disposing.

B. Research result analysis

We first adopts applied server complication line to do “store and take” operation. We choose pictures randomly to analyze, finding the size of pictures is between 2K and 10M, and the effect is like Figure 2.

Storage system constructed by Hadoop can read TPS (transmission number of operations per second) and it increases with threads, but its growth gradually slowed down to 70 when the thread reached to the first peak, and then add read threads, and TPS is no longer growing stably. Traditional NFS storage systems require multiple IO when read pictures, so when the number of threads have reached to the 40, the system arrive to its load limit. While NFS storage systems using Hadoop’s new storage system reduces the number of IO operations, under the same machine configuration, you can achieve more concurrent read operations.

We then analyzed using random writes pictures. The picture size distribution is in the 2K-10M, including small pictures and big pictures. After the client concurrent requests, HDFS will store pictures. The effect is shown in Figure 3.

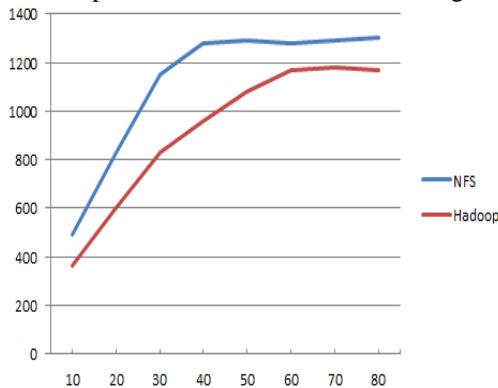


Figure 2. Image read analysis

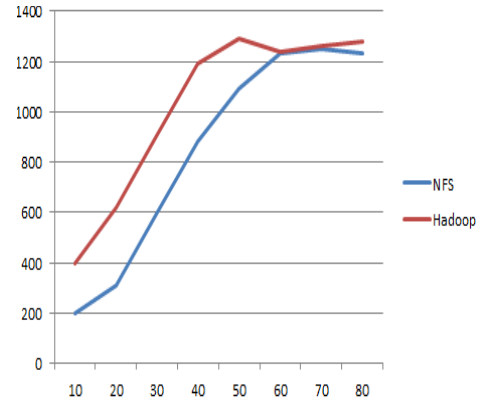


Figure 3. Image written analysis

Written TPS’s thread count peaked at 60, and then an additional written thread is involved, TPS is no longer growing stably at this moment. Written performance of the new storage system of Hadoop is superior to NFS written performance which ensure the high throughput of the system.

By analyzing renderings of concurrent read and write multi-thread, you can find a new storage system uses Hadoop in optimizing the storage procedure, reducing the number of IO access process and improving the carrying capacity of the system. In a word, the overall deposit is better than NFS systems. Especially in the face of high concurrent read and write operations, the new storage system HDFS guarantee the stability of the system and the security of the data. So we can be sure that the new storage systems of Hadoop have the ability to solve the problem of massive storage ability.

V. CONCLUSIONS

This paper designs and develops massive pictures and data storage platform based on Hadoop, adopting the technology of Linux data base and parallel distributed database technology. This paper employs a series of pictures and data management ways such as: HDFS distributed file system, Map/Reduce parallel calculation module and HBase data base technology. Besides, the paper also adopts Redis and HAProxy to construct buffer and load balancing, in this way, the whole system reach to a stable and healthy condition. If this platform is built on many cheap and common computers, the requirement of high efficiency and pictures and data management can be reached. The result of platform module realization manifests: this system has fine extension quality and is easy to defend, so the technological route and design method adopted by the system is effective and feasible.

ACKNOWLEDGMENT

China West Normal University Foundation (17YC178, CXTD2017-6)

REFERENCES

- [1] Zhao Xiao-yong, Yang Yang, Sun Li-li, Chen Yu. Hadoop-based storage architecture for mass MP3 files. *Journal of Computer Applications*[J]. 2012, 32(6) : 1724-1726.
- [2] Li Lin, Design and Analysis of the Mass Image Storage Model Based on Hadoop[D], 2011, 11
- [3] Cui Jie, Li Taoshen, Lan Hongxing. Design and Development of the Mass Data Storage Platform Based on Hadoop. *Journal of Computer Research and Development* [J]. 2012, 49:12-18
- [4] Liu Jun. Design and Research of the Mass Teaching Resources Storage Platform Based on Hadoop. *Computer & Telecommunication* [J], 2013, 07
- [5] Tai Jian-hua. Research of Mass Data Storage Technology Based on Hadoop Platform[D]. 2012, 06
- [6] Tome White. *Hadoop: The Definitive Guide* [M]. O'REILLY PRESS, 2009:9-13
- [7] Dean J'Ghemawat S. MapReduce: Simplified data processing on large clusters[C]//Proc. of the 6th Symp. on Operating System Design and Implementation, 2004:137-150
- [8] Zhang Jian-xun, Gu Zhi-min, Zheng Chao. Survey of research progress on cloud computing. *Application Research of Computers*. 2010, 27(2):429-433
- [9] Li Jun, Li Yong. Research on Integrated Frame and Implementation of Federated Heterogeneous Database Application Systems. *Application Research of Computers* [J]. 2001, 18(4):19-22
- [10] Ding Xiang-Wu, Zhang Dong-hui Safety Reinforcement for Hadoop Platform. *Computer Engineering*[J]. 2017, 12(4):5-9
- [11] Saleh E, Alsa'deh A, Kayed A, et al. Processing Over Encrypted Data: Between Theory and Practice[J]. *ACM SIGMOD Record*, 2016, 45(3): 5-16.
- [12] Tang Yan, Liu Ren-quan, Wang Ping. Design and implementation of college and university big data platform based on Hadoop[J]. *Information Technology*, 2017, 12:105-110