Research and Software Implementation of Document Service System Based on Cloud Platform

Shuangyan Shen, Zebo Zhu, Jianming Liu, Fulin Zhao, Chun Liu
China Satellite Maritime Tracking and Control Department, Jiangyin, 214431, China

Abstract. With the explosive growth of various types of documents, these resources have been collated and utilized to become a concern of the industry. This paper builds a cloud computing platform based on OpenStack, and implements a document service system based on the built cloud computing platform. The administrator can perform state monitoring and management operations on the implemented document service system, and the system user can delete, search and classify the document. This research has greatly improved the use of documents and the level of scientific research.

1. Introduction
With the explosive development of information, the amount of data in various types of documents has also exploded, especially in government agencies and research institutes. The accumulated resources will become larger and larger over time, plus these resources. The heterogeneous relationship between them, how to organize and utilize these rich resources has become a concern of the industry. At present, some organizations have developed data storage platforms based on cloud computing platforms, which have improved the sharing and management efficiency of specific resources.

There are currently a number of deployable cloud computing platforms, such as: Amazon AWS, Rackspace, Google App Engine, Microsoft CRM, Salesforce.com and other platforms. However, these platforms are mostly computing platforms developed by large companies. The source code is not public, and there are security risks for independent use and application.

In order to realize the rational use of scientific research resources and improve its use efficiency, this paper builds a cloud computing platform based on OpenStack, and implements a document service system based on the built cloud computing platform. This research has found an effective solution for deep integration of various resources and improving the efficiency of the use of document information.

2. Construction of OpenStack cloud computing platform
The core architecture of the OpenStack cloud platform constructed in this paper is divided into six modules: Nova, Cinder, Glance, Keystone, Ceph, and Neutron. It can also include optional modules such as Horizon and ceilometer. The overall architecture topology is shown in Figure 1:
Figure 1. OpenStack cloud platform overall topology

This deployment uses 4 nodes, one control node and three compute nodes. Among them, the two nodes are the newly purchased ThinkServer RS260 server, the other two nodes are the original servers, and the memory of the control node and the computing node are both 8G. Each server uses three 1T SAS hard disks, one for system installation, and the other two for Ceph storage, which is used as a Ceph OSD node, as shown in Figure 2.

Figure 2. Node role assignment
Configuring the disk on the controller

Figure 3. Disk usage

Each server uses two 1.0Gbps NICs, which are connected by different Gigabit switches. On the first NIC, configure the Fuel Management Network (Admin PXE), the OpenStack virtual machine's private network, the OpenStack storage network, and the OpenStack management network, separated by Vlan. The specific Vlan number is shown in Table 1, and the second NIC is configured as OpenStack. The external network, different networks can be configured into different combinations by drag and drop.

The specific network configuration and network usage are shown in Table 1.

| Name     | Vlan        | Description                                | Password                  |
|----------|-------------|--------------------------------------------|---------------------------|
| Admin    | 10.20.0.0/24| Fuel deployment, management network        |                           |
| Private  | 1000-1030   | Private network between virtual machines   |                           |
| Storage  | 102         | OpenStack storage network                  |                           |
| Management| 101         | OpenStack Management Network               |                           |
| Public   | 129.23.55.0/24| OpenStack Management Network            |                           |

After the network configuration is complete, you can verify that the network configuration is correct by verifying the network.

3. The requirements of the documentation system are implemented.

Traditional document systems can only read and search documents by identifying the title of the document. The content of the test document is more complicated. Reading only the title of the document cannot extract valid information from the document. Therefore, the designed test document system needs to effectively identify the specific information in the document content.

The built system implements this requirement in the following ways:

a). Use Apache POI technology to read specific information in documents such as Word and Excel.

b). Use PDFBOX technology to read specific information in PDF files.

c). Use the Java Zip technology to read the documentation in the rar/zip archive.

d). Pictures, sav and other files are directly uploaded to the server by means of files. The detailed information of the files stored in the database, including size, upload date, uploader, storage path, etc., and download functions are provided in the system.

The information read from the document is stored in an RDF file manner, and various types of information stored in the file are stored in a distributed manner; for the data characteristics of the RDF data, the frequency of the keywords in the RDF graph is calculated by using a statistical cost model, thereby calculating The weight of each edge is divided, and the RDF data graph is divided according to the weight and the smallest edge cut set.

Using Hive technology to analyze and process the extracted data, Jena reads the RDF information, and uses Map/Reduce technology to analyze and obtain user information, optimize the ranking of
SPARQL, and calculate the degree and degree of each node according to the statistical cost model. And the data storage location, sorting the queries to minimize the intermediate results of each query. Use the SPARQL query to retrieve the data the user needs in RDF big data. The specific flow chart of user uploading documents and document processing is shown in Figure 4.

![Figure 4. User-uploaded data and specific flow chart processing](image)

4. Conclusion
This paper builds a cloud computing platform based on OpenStack and implements the document service system. The following conclusions are obtained:

a). The built OpenStack-based cloud computing platform can meet the needs of managers and users in the system.

b). Within the implemented document service system, the administrator can monitor the users of the cloud platform and their status, and can perform management operations on the user.

c). Users of the document service system can perform document uploading, modification, deletion, and searching, and can classify uploaded documents.

d). The implementation of the document service system based on the cloud platform greatly improves the efficiency of the use of documents and improves the level of scientific research and experimentation.

References
[1] [EB/01] https://it.wikipedia.org/wiki/Cloud_computing, 2016
[2] Kumar R, Gupta N, Charu S, et al. Open source solution for cloud computing platform using OpenStack [J]. International Journal of Computer Science and Mobile Computing, 2014, 3 (5): 89-98.
[3] Zhang Q, Hellerstein J, Boutaba R. Characterizing task usage shapes in Google’s compute clusters[C]. Proceedings of the 15th international conference on World Wide Web. LADIS, 2012: 122-125.
[4] Rossignieux F, Lefevre L, Gelas J P, et al. A generic and extensible framework for monitoring energy consumption of OpenStack clouds [C]/Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on. IEEE, 2014: 696-702.