Modified Architecture of Cloud Computing System

Prof. Dr. Subhash P. Rasal

Abstract: In recent years, the technology of cloud computing has been widely applied in e-business, e-education and etc. . Cloud computing platform is a set of Scalable large-scale data server clusters, it provide computing and storage services to customers. The cloud storage is a relatively basic and widely applied service which can provide users with stable, massive data storage space. The research shows that the architecture of current Cloud Computing System is central structured one; all the data nodes must be indexed by a master server which may become bottle neck of the system.

In this paper, new cloud storage architecture based on P2P and design a prototype system is proposed. The system based on the new architecture has better scalability and fault tolerance.

Keywords: cloud computing, P2P, storage.

I. INTRODUCTION

Cloud computing is Internet-based computing, whereby shared servers provide resources, software, and data to computers and other devices on demand, as with the electricity grid. Cloud computing is a natural evolution of the widespread adoption of virtualization, Service-oriented architecture and utility computing.

A cloud computing platform dynamically provisions, configures, reconfigures, and provisions servers as needed. Servers in the cloud can be physical machines or virtual machines. Advanced clouds typically include other computing resources such as storage area networks (SANs), network equipment, firewall and other security devices. This seminar will focus on the storage service from cloud. Some typical cloud systems such as GFS of Google, Blue Cloud of IBM, Elastic Cloud of Amazon, have a similar architecture for storage. In the system architecture there is a central entity to index or manage the distributed data storage entities. It is effective to simplify the design and maintenance of the system by a central managed architecture, but the central entity may become a bottleneck if the visiting to it is very frequent. Although systems in practice have used some technique as backup recovery to avoid the probably disaster from the central bottle neck, the flaw come from the architecture has not resolved essentially.

This seminar proposes a cloud computing architecture based on P2P which provide a pure distributed data storage environment without any central entity. The cloud based on the proposed architecture is self-organized and self-managed and has better scalability and fault tolerance.

Rest of the paper is organized as follows, in section 2 introduces some related work about cloud storage system and P2P storage system. In section 3 describes a typical scenario to explain the architecture of our proposed cloud computing storage environment. In section 4, there is an introduction on our prototype about the P2P cloud system. Section 5 is conclusion and proposal for future work.

II. RELATED WORKS

2.1 STORAGE ARCHITECTURE IN THE CLOUD

Storage architecture of the cloud includes the capabilities of the Google file system along with the benefits of a storage area network (SAN). Either technique can be used by itself, or both can be used together as needed.

Computing without data is as rare as data without computing. The combination of data and computer power is important. Computer power often is measured in the cycle speed of a processor. Computer speed also needs to account for the number of processors.

When looking at disk storage, the amount of space is often the primary measure. The number of gigabytes or terabytes of data needed is important. But access rates are often more important. Being able to only read sixty megabytes per second may limit your processing capabilities below your computer capabilities. Individual disks have limits on the rate at which they can process data. A single computer may have multiple disks, or with SAN file system be able to access data over the network. So data placement can be an important factor in achieving high data access rates. Spreading the data over multiple computer nodes may be desired, or having all the data reside on a single node may be required for optimal performance.

The Google file structure can be used in the cloud environment. When used, it uses the disks inside the machines, along with the network to provide a shared file system that is redundant. This can increase the total data processing speed when the data and processing power is spread out efficiently.

Prof. Dr Subhash P. Rasal is working as HOD in Department of Electronics in Mudhoji College, Phaltan (Maharashtra) Email:subhashprasal@gmail.com)

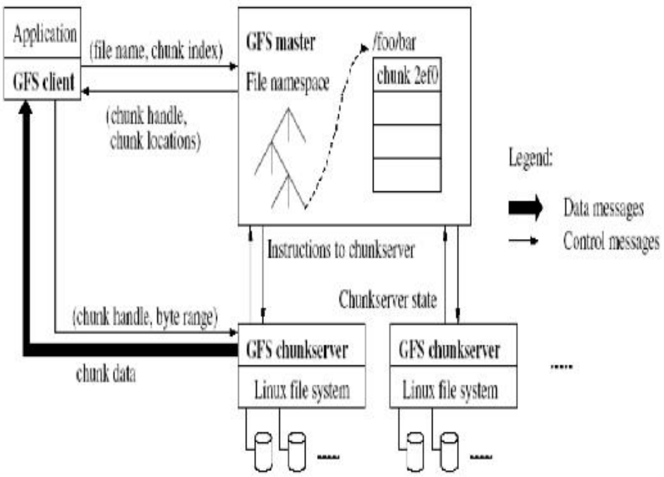


Figure 1: Architecture of Google File System

2.2 Google File System

The first to give prominence to the term cloud computing (and maybe to coin it) was Google's CEO Eric Schmidt, in late 2006.Google Inc. has a proprietary cloud computing platform which was first developed for the most important application of Google search service and now has extended to other applications. Google cloud computing infrastructure has four systems which are independent of and closely linked to each other. They are Google File System for distributed file storage, Map Reduce program model for parallel Google applications, Chubby for distributed lock mechanism and Big Table for Google large-scale distributed database.

Figure 1: Shows the architecture of Google file system. A GFS cluster consists of a single master and multiple chunk servers and is accessed by multiple clients. Chunk servers store chunks on local disks as Linux files and read or write chunk data specified by a chunk handle and byte range. The master maintains all file system metadata. This includes the namespace, access control information, the mapping from files to chunks, and the current locations of chunks. When a client wants to visit some data on a chunk server, it will first send a request to the Master, and the master then replies with the corresponding chunk handle and locations of the replicas. The client then sends a request to one of the replicas and fetches the data wanted.

The GFS above is actually a central indexed distributed storage system. GFS master work as an index server which can provide the global information about each chunk server for clients. The flaw of central index architecture is that the GFS master may become bottle neck of the system since all the request to the target data chunk must be originated from the index server which burdens the master.

2.3 P2P Storage System

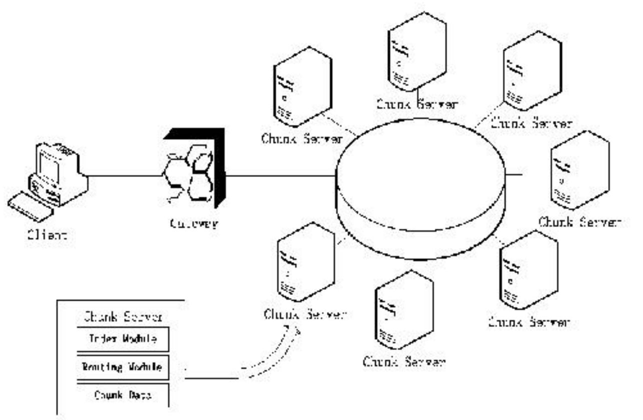
The distributed P2P network indexed by DHT arithmetic can resolve the problems of bottle neck come from central index system. Since the management is distributed equality to every peers in the network, there is no bottle neck any more, but the new problem is how to keep the consistency of the replica when read/write. Some P2P systems for distributed storage have been developed now, such as Ivy, Eliot, Oasis, Sigma, etc. . They keep the replica consistency in different way and index the data resource by DHT. In the following section, we will propose a cloud storage system based on P2P which can keep the consistency with an innovative method.

III Cloud Based on P2P

Architecture

We design a new system of P2P storage for cloud platform, which can take advantage of the P2P distribute architecture and do well in concurrent update.

Following figure 2 shows the architecture of the system:



Figrue 2 Cloud Storage Based on P2P

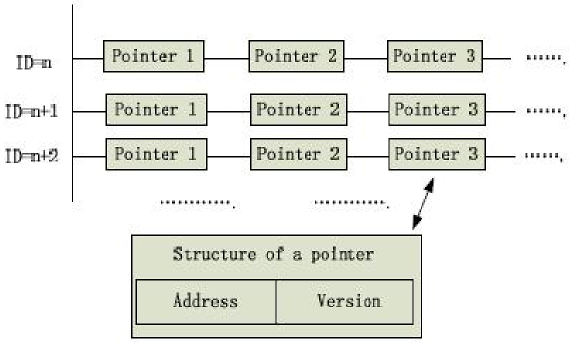
Roles involved in our architecture can be defined as follows and illustrated as below:

Client App: The client application which wants to get the data from the platform.

Gateway: The entity which can transfer the request or response between the Client App with the network and can lead the request to the nearest node in the network.

Chunk Server: The entity which is served as the data resource node and P2P node. Different with the function of pure data storage in GFS, the chunk server here has three function modules with separated interfaces. As shown in the figure above: Index Module, take charge of part of the global resource index which is assigned by DHT arithmetic such as Chord, Pastry and so on. Route Module; pass a lookup request by a next hop routing table which is also assigned by DHT. Data Module, provide the data resource stored in the local machine.

In the index module, as shown in the figure 3, there is a chain containing the data index information pointers to all of the data blocks with the same name ID will be linked in a sub chain. A pointer contains the address of a data block and the update version number of that block.



Figrue 3 Index Module Details

3.1 TYPICAL WORKFLOW

In this section, we will present the system working flow with a typical scenario.

At first, before a Client App can do its work, data blocks and the corresponding replica should be uploaded to the Chunk Servers. How to select the chunk servers for storage is the same with the traditional cloud computing platform. When a Client lookup data block, the work flow is different, as below:

1. Client App sends a request for a data block with logic identifier to Gateway.
2. Gateway analysis the request, parsing the identifier of the data block in the request, such as logic address, and change it to 128 bits logic ID by DHT algorithm which can be recognized by chunk server P2P network.
3. Gateway constructs a P2P search request data package including the logic ID, and sends the request to the chunk server P2P network
4. The P2P search request package routed among the chunk servers following the P2P search protocol such as Chord, Can, Pastry, Tapestry and so on. The chunk servers now act as a routing nodes of P2P and the routing interface will be taken used of.
5. The request reaches the server which contains the index information of the logic ID in searching.
6. The index includes all the pointers of the data replica with the same ID. The chunk server now acts as an index server and the index function interface will play its role. The chunk server will select a latest pointer by its version number, if there are more than one candidates, the server should select a nearest one by comparing the IP address of the client App and the data resource container, then return the best address to the client.
7. When the Client App gets the best address, it will then send its request to the address of the chunk server which contains the data block. Now the chunk server acts as a data provider as the traditional cloud storage platform does.

IV CONCLUSION AND FUTURE WORK

In this paper, we propose a new architecture of cloud computing system based on P2P protocol, which has two advantages: data access is faster when clients requesting for data simultaneously and no more relying on only one server and moreover it resolve the problems of bottle neck come from central structure.

In the future work, we will do some optimization about the throughput of the system by the technique such as pipelining read or write.

REFERENCES

[1] M. Zeller, R. Grossman, C. Lingenfelder, M. Berthold, E. Marcade, R. Pechter, M. Hoskins, and R. Holada, "Open standards and cloud computing: KDD-2009 panel report," in *KDD*, Paris, France, 2009

[2] Balachandra Reddy Kandukuri, Ramacrishna PaturiV, Atanu Rakshi, “Cloud Security Issues”, 2009 IEEE International Conference onServices Computing