Future Generation Computer Systems 38 (2014) 47–60

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Granary: A sharing oriented distributed storage system[☆]

Hongliang Yu[a,c,b], Fan Zhang[a], Yongwei Wu[a,c,*]

[a] Computer Science Department, Tsinghua University, Beijing, China

[b] National Engineering Laboratory for Disaster Backup and Recovery, Beijing, China

[c] Research Institute of Tsinghua University in Shenzhen, 518057, China

ARTICLE INFO

Article history:
Received 26 December 2011
Received in revised form
22 December 2012
Accepted 2 August 2013
Available online 10 September 2013

Keywords:
Distributed storage
DHT
Key-value
File sharing
Reliable storage

ABSTRACT

Up to now, more and more people use Internet storage services as a new way of sharing. File sharing by a distributed storage system is quite different from a specific sharing application like BitTorrent. And as large file sharing becomes popular, the data transmission rate takes the place of the response delay to be the major factor influencing user experience. We present the design and implementation of a distributed storage system named as Granary in this paper, which provides reliable data storage and sharing service to cyber users. Granary uses a specific DHT(Distributed Hash Table) layer to store file meta-data and employs a raw data storage scheme to scatter large data. We introduce its adaptive DHT recovery algorithm in this paper which assures the availability and consistency of meta-data with small bandwidth consumption and improved throughout. The replication strategies which are used to accelerate file sharing with low bandwidth consumptions are further discussed. Experimental results show that these methods offer a reliable and efficient data storage and sharing speed with network bandwidth costs less than conventional policies.

1. Introduction

Information centered digital ecosystems are emerging which connect people all around the world through the use of modern information technologies. Email is used as a tool for making connections, but it is inconvenient for its notable delay and limitation on the sizes of data exchanged. As the improvement of computer hardware and network infrastructure which offers more bandwidth and storage spaces, the trend of making connections by sharing and exchanging data like business files, music files, home videos or picture albums, in digital world is emerging.

More and more people use Internet storage services as a new way of sharing [1] up to these days. For example, commercial online storage services like Amazon S3 [2], Dropbox, Facebook Haystack [3], Google Megastore [4], Microsoft Azure [5,6] are offering gigabytes storage space for different user applications or directly to end users. Sharing based on a storage system is quite different from specific file-sharing applications like BitTorrent and eMule. By uploading the file to storage servers, users do not need to keep online to share the data. The quality of service is also guaranteed by the infrastructure of the storage system, not end users. Further, as audio and video sharing becomes common and popular, the data transmission rate takes the place of the response delay to be the major factor influencing user experience. On the other hand, as the file size grows, bandwidth consumption becomes more critical than before. Besides, data reliability and availability are always problems brought by Internet-based online systems. How to make the data reliable and available while getting a good accessing performance is the main goal of system designs. Further, the scalability of such systems is also important for commonly faced large number of users in digital earth.

Such factors lead to different design choices such as that of data replication and caching schemes. Mainly designed for file storage instead of file exchange, state-of-art distributed storage systems like Dynamo [2], Comet [7], ElasTras [8], Depot [9], Silt [10], Oceanstore [11], Pond [12], Past [13], or SafeStore [14] do not tackle the performance problem of sharing large files sizing megabytes or even gigabytes. Therefore, it is still challenging to accelerate sharing speed with low bandwidth consumption in such systems.

In this paper, we introduce a distributed storage system named as *Granary* which provides reliable storage service to end users. The system is designed for connecting people in a scalable manner. Users store their files and share some of them with others. Granary is a roughly two layer storage system. It uses DHT to store meta-data and employs a raw data storage scheme to scatter large data. Our main contributions are as follows.

- We design the sharing oriented Granary storage system by using a layered architecture. It abstracts and separates the

[☆] This paper was supported by the National High Technology Research and Development Program (2012AA012600).

[*] Corresponding author at: Computer Science Department, Tsinghua University, Beijing, China.

E-mail address: wuyw@tsinghua.edu.cn (Y. Wu).

0167-739X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.future.2013.08.001

execution of data communication, meta-data storage, large data storage, and tree-like file system.

- We introduce a DHT-based meta-data storage and recovery strategy which uses an adaptive algorithm designed for bandwidth saving and throughout improving, which assures the availability and consistency of meta-data.
- We implement a data replication mechanism to solve the problems brought by file sharing. It makes copies of files according to their locality and popularity, and thus decreases unnecessary network traffic to improve system performance.
- We deploy the Granary system in 20 machines located on campus networks across 5 different cities. We evaluate the methods used in the system both from deployed system tracing and system simulation.

This paper presents the design, implementation and evaluation of sharing oriented Granary storage system. In Section 2, we present an overview of Granary architecture. We describe the design in detail in Sections 3 and 4. In Section 5, we evaluate Granary's ability to share data files, recovery from data loss, and discuss our experiences. The related works are included in Section 6 followed by discussions and conclusions in Section 7.

2. System overview

Granary is a wide-area networked storage system, it is composed of dedicated nodes at a global scale and provides online storage services to end users. Users can upload their files onto the nodes of Granary and can download or modify these files. More important, users can use Granary to share their files with their friends. It contains a DHT storage layer which is also distributed on wide-area networks and stores meta-data into it. Granary uses replication to maintain the availability and reliability of data when sharing.

The architecture of the Granary system is shown in Fig. 1, it is mainly composed of 5 layers: basic service layer, structure overlay layer, basic storage utility layer, advanced storage utility layer and application layer.

- *Basic service layer*. It provides synchronous and asynchronous network communication, message handling, thread management and data encryption to layers above.
- *Structure overlay layer*. A DHT-based overlay network is constructed to organize the whole nodes in this layer.
- *Basic storage utility layer*. Two kinds of storage services are provided in the layer, including a meta-data or small data oriented DHT storage service and a large file oriented raw data storage service.
- *Advanced storage utility layer*. This layer provide a key–value-based file storage service, where the content of a file can be fetched by its name. An adaptive data replication management strategy is implemented too.
- *Application layer*. The layer provides a structured tree-based file storage service to make up for the plain file structure in key–value storage. It also optimize the storage of large data.

The DHT stores DHT objects. A DHT object consists of a key and a value and comes with a 128-bit hash value, which is the MD5 hash of its key. Every node is assigned a unique 128-bit key called nodeId, which can be the hash value of a per-machine signature such as MAC address. A DHT object is replicated and stored on the closest nodes in terms of nodeId, which are probably disperse in terms of physical location. This provides high availability to DHT objects.

There are three kinds of meta-data in Granary's DHT system. Its first kind of meta-data indicates which nodes of Granary are storing replicas of a certain file. Before users' files are located, Granary will fetch this kind of meta-data first to learn where users' files are located. Such meta-data may be updated by upper-applications of Granary because Granary may migrate the replicas adaptively in order to boost performance and balance loads of data sharing.

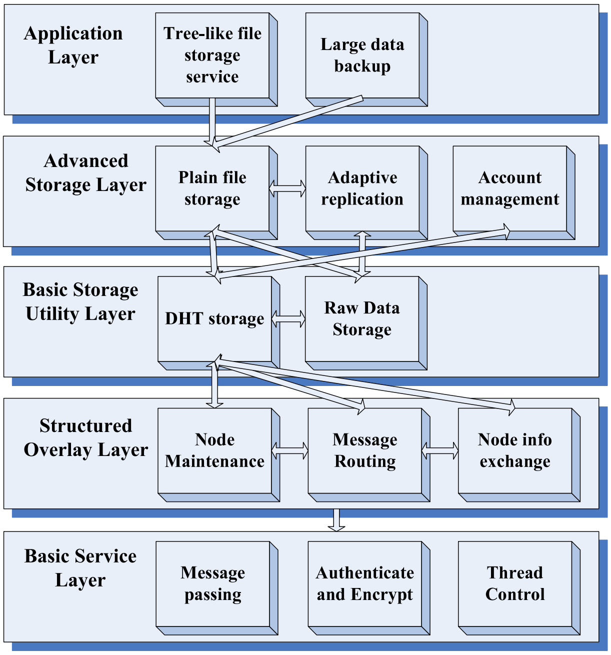

Fig. 1. The architecture of the Granary storage system.

The second kind of meta-data includes the information about Granary's users, including user names, passwords and quota information. Granary will fetch this kind of meta-data to check the user's authorization and whether the user has reached the upper limit of his/her quota. This kind of meta-data will be updated when users change their passwords or upload/delete their files.

The last kind of meta-data is users' catalog information, which represents structures of users' files. Granary must fetch this kind of meta-data when users want to list their files. The catalog meta-data are updated the most frequently, since Granary updates them whenever users make any changes to their catalog, e.g., adding a new file or deleting an old one.

Granary does not directly store files in DHT. Instead, it stores and replicates files in the upper layer and puts only the locations of replicas (which is called the *replica list*) and the meta-data of files in the DHT. The high availability of DHT guarantees that the meta-data can be accessible despite the network disconnections or node failures. Because the keys of the meta-data in DHT are generated by a secure hashing function, such as MD5, the hot meta-data are hardly converged to a single node. Moreover, the operations on meta-data are often based on short-term communications with small size data transmission. As a result, the loads of node with a few hot meta-data will not be extremely heavy. Therefore, we don't need to design a specific load balancing algorithm for the meta-data access. We focus on improving the performance of the raw data operations.

A *Granary* server acts not only as a DHT node, which routes messages for other nodes and stores DHT objects that are mapped to it, but also as an upper-level storage server which clients access directly for file uploading and downloading. This choice of design allows us to develop a more flexible replication algorithm that are not constrained by the binding of nodeIds and objectIds in the DHT. Since trivial operations like message transmission and meta-data read/write are all handled by the DHT layer, we are able to focus on the replication strategy itself to boost the performance of data sharing.