Operating System Theory Project Proposal
Optimized Distributed File System with Multi-device Rescue Measures

Yi Qi yq97
Yuhang Zhou yz853

Introduction

Our project is to implement a distributed file system mainly based on GFS architecture. And we will also make some optimizations in multi-device rescue measures when the servers are not able to be accessed.
The distributed file system is easy for users to expand their storage without any hardware upgrade. Users can also transfer files between different devices with this system. The security of files is guaranteed by the backup in servers and authority management of the system.
Our optimization is mainly for multi-device rescuing measures when the server is down. For example, a user has multiple devices accessing the system. A file in his desktop has a copy on the server so that the user can access this file in the server with his laptop in traveling. While the server may down or the file may be lost during that time. So our system should be able to check all his online devices with this file backup and build a simple instant server in that device so that his laptop can access this file from this device directly.

Analysis

There are already some existing distributed file systems. And now developers have some issues to pay attention to such as scalability, reliability, flexibility, transparency, fault tolerance, security, architecture and so on.
NFS: It's a basic distributed file system, with one server and multiple clients. Each client can issue a system call to access the files on the server.
AFS: It's a stateful file system, with scalability as a major goal. Since with states, it doesn't need to check whether the cached contents are changed, which can influence the performance.
GFS: The google file system which can tolerate some faults. A GFS cluster consists of a single master and multiple chunk servers and is accessed by multiple clients. The chunk server keeps some replica if one of these replicas corrupts, it can use another as a backup.

So most of these file system assumes that the clients have a very stable with the server, but if the server is down, even for a few seconds, clients have a big penalty.

We choose the GFS as our system architecture because of the reliable combination of the master server and chunk servers. And we get the idea from the google file system that we can use our own devices as some chunk servers and keep some files just in case the client can’t access the data server. And the master server can help to schedule during the rescue time.

Deliverables

The deliverables during this semester is a workable distributed file system.
The server will be implemented in GFS architecture, including clients, master server and chunk/data servers.
The client of our system will be implemented as a web application so that different platforms can using it with a browser.
Functions to implement:

1. Upload file to the data server
2. Download file from the data server
3. Synchronize a folder of a device and the data server
4. Access to the system with multiple devices
5. Multi-device rescue measures as described in the first part

Evaluations:

1. Data throughput
2. Upload and download speed
3. Load test
4. Rescue timing with our optimization