Group 7 Design Document: P2P Filesystem

Members: Siddhanth Sharma, Ajay Raj

Github Link: https://github.com/ajayraj/179F_proj

Project:

Our goal is to implement a Peer-to-Peer filesystem in Clear Linux using FUSE. We aim to utilize a key-value-store to implement the file system abstraction. As a general assumption, we take that a node can join and leave the system, but not too frequently. Implementing logging per peer to manage write conflicts and using CHORD to organize the p2p network will allow us to create a working read-write system.

Components and Justifications:

Docker

- Easily sharing a configured project environment with the required dependencies included. Docker Hub repo: https://hub.docker.com/repository/docker/ajayraj/179f_proj
- Allows for the creation of multiple virtual workspaces and network topologies without the need for physical computers

Clear Linux

- New open-source Linux distribution by Intel optimized for performance, cloud support, and quick updates. Said to work well with Docker and Kubernetes.

Libfuse

- Can use libfuse as a protocol (CHORD distributed hash table algorithm) and DHT translator
    - Read/write desired info in P2P file system and track the most recent changes locally and in the network
- Easy-to-use open source filesystem that is particularly useful for virtual filesystems (allows for a user to access both the p2p network filesystem and their local filesystem transparently without seeing a difference)

Chord

- Analysis of file usage can be done to determine which files are more commonly accessed than others
- Cache commonly used files with peers that are located more closely based on CHORD to speed up file updates/distribution across P2P system
- Implementing a Logged File System (like Ivy).
    - Collect metadata for changes made by each peer, store changes in DHT to avoid write conflicts and keep track of the most recent changes
    - Each peer maintains its own log of changes and a private snapshot of logs of other peers

Timeline of Objectives:

1. Week 5
    a. Working ClearLinux Docker Image
    b. Working FUSE filesystem installation in Docker Image
2. Week 6
    a. Basic Filesystem with example files/directories
    b. Virtual network of connected Docker Images
3. Week 7
    a. Implement Distributed file system
    b. Implement Distributed Hash Table/Chord mapping
4. Week 8
    a. Handle P2P edge cases:
        i. People leaving/joining p2p server
        ii. Locating data via each user
        iii. Load balancing
    b. Start implementing logging of changes per peer
5. Week 9
    a. Caching of “commonly” accessed files
6. Week 10
    a. TBD