ECE 6102 Project Proposal

Sunil Kavalan, Banji Jolaoso, Brett Wilson, Vinay Bharadwaj

October 5, 2012

Goal Secure distributed dependable ftp like service.

0.1 Overview

We propose to create a fault tolerant distributed ftp like service. This service will be immune to Byzantine faults on any single server, which will by definition provide strong guarantees of data reliability and availability. We will also provide that communication with the service is secure and will prove by nature of the service immunity to data poisoning on any individual server. We propose to satisfy this with an f-masking system to prevent faults. We will secure the channel and provide a CA server to verify authenticity of the parties.

0.2 Functionality

In order to implement this we will need to create a client side application and a server side application. We will assume for the initial design that the clients are aware of all the IP address of the servers in the network.

0.2.1 Client

From a user perspective the service will be act like a standard sftp service, providing the basic high level transactions that an ftp service would provide. The service will organize users into groups each group having read/write privalages on a single file structure. Underneath the UI the service will query a subset of servers for data(known as a quorum. The client will retrieve the data from a quorum of servers and compare the information. By intelligently picking the size of the quorum we can guarantee that all operations are secure and fault tolerant. This quorum also guarantees that all write operations can be propagated to any other requesting system without inter server communication. In order to make this claim all write operations will have to have a unique timestamp that will provide information to clients as to the most recent faultless version of the data. To check integrity the data, a hash computation is saved at client before sending to the proxy server. On every read operation, the hash of retrieved data is computed again to match with the saved one.

$$|Q| = \lceil \frac{n + 2f + 1}{2} \rceil \tag{1}$$

n is the size of the cluster and f is the number of acceptable failed servers and $n > 4f$.

0.2.2 Server

The server will store all data in an encrypted format. Decryption will occur when a channel is opened by an authenticated user. The server will store the files and a directory tree with all the timestamps. Attempted transactions will require a password authentication which will be used to decrypt the data and transmit it.

The server will store a list of users and passwords all encrypted by the user's password. These will be combined with master passwords to allow group level access to files on the server. The CA is an independent server that validates the authenticity of the server to the client and vice versa.

0.3 Facilities

- OpenSSL for distribution and authentication of parties using CA server
- Built-in security packages for preforming the encryption
- VMware and batch jobs on the ECE cluster to simulate server clusters

0.4 Workload Breakdown

0.4.1 Client

- ftp- "like" protocol – Sunil
- network traffic –Brett
- Quorum resolution - Vinay
- Channel encryption – Banji

0.4.2 Server

- ftp- "like" protocol – Sunil
- network traffic –Brett
- Encryption – Banji
- CA Server – Banji
- Failure Emulation – Sunil

0.5 Benchmarks

Expected Load The format of this solution provides interesting load distribution, we will describe the expected load and prove it to be true under several operations

$$\text{Load}_{System} = n^{-1}|Q| \tag{2}$$

Meeting Reliability The system should provide data correctly under a variety of read and write operation and a verity of faults including full server failure and arbitrary data faults.

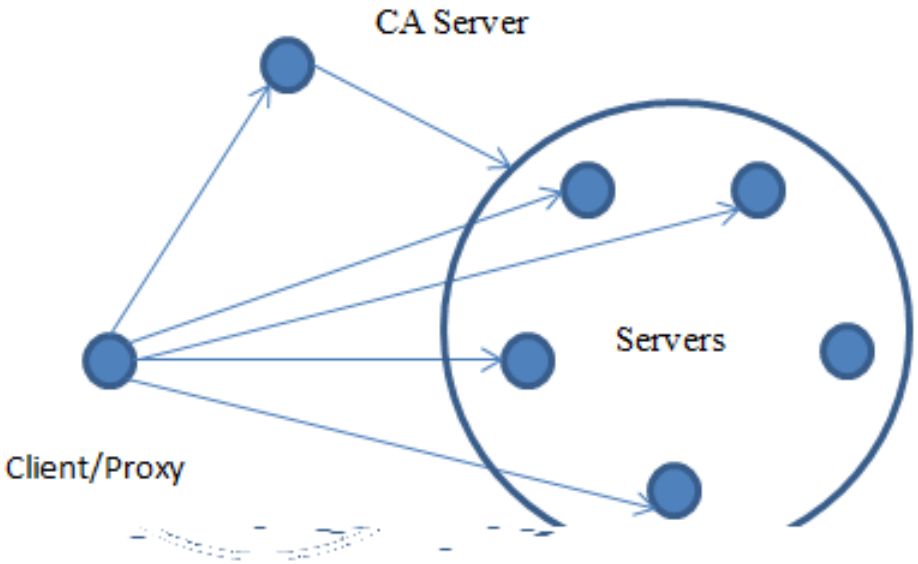
0.6 If Time Allows

Dynamic Sizing of Server Cluster Currently clients require the full list of servers precluding any fault correction for faults that appear as server downtime. If we can extend this system to allow for dynamic addition of servers, this solution can be converted to a decentralized network, potentially p2p.

Extend Group Organization Control Allow for modifiable read and write privileges on group's data and more complex group level accessibility to files.

Support Proxy Server Allow for a proxy server that will move load on the client to a external machine.

Proxy In order to offload client workload, the service will provide optional offloading of read/write quorum operations to a external server.


CA Server
Servers
Client/Proxy