

Project Report
Group 17

Collaborative File Retrieval Network

Group details:

Contribution

1. Pranavdeep Singh	21119036	Server and Peer
2. Priyansh Mawal	21114076	Server and Peer
3. Piyush Arya	21114074	Receiver
4. Rajat Raj Singh	21114079	Receiver

Project Description:

The developed network allows a system to request a file, and receive parts of the file in a concurrent manner from different peers and reconstruct it.

This network consists of 2 physical entities:

1. **Server**: It is responsible for handling the requests from receivers and assigning tasks to peers in order to complete file transfer.
2. **Client**: The client is divided into 2 logical entities:
 - **Receiver**: Responsible for requesting the file and reconstructing the concurrently received parts of the file
 - **Peer**: Responsible for sharing particular portions of the file to the receiver as instructed by the server

Technologies used:

- Socket API
- TCP Connections
- fork(): To create child processes to enable concurrency

Working:

Server and Peer: The moment a new system connects to the server, the peer sends a registration message to the server and the server acknowledges the registration. The peer will send periodic file system updates to the server in order to notify the server of the existing files on the system. The server maintains a datastructure storing the list of systems who currently have the file. The server is capable of serving multiple requests concurrently. The moment a request for a file arrives at the server, the server sends the list of peers which can possibly send the data to the receiver. The receiver chooses which and how many peers to receive the file from and sends this message to the server. The server sends concurrent messages to each peer to send a particular part of the file, by using load balancing, to the respective receiver. The server waits for an acknowledgement from the receiver regarding successful transmission of the file. If it receives a negative acknowledgement, it reassigns peers for the failed tasks.

Receiver: The receiver mode is always running concurrently along with the peer mode. Whenever a system wants to request a file, it sends a message to the server requesting it to assign the peers from which to accept from. The server replies with a list of peers which are ready to send the requested file. Since every system has a maximum concurrency limit, the receiver replies back with the list of the peers it is ready to receive from. The receiver then create multiple concurrent processes on which multiple peers send different portions of the file. The connection is accepted only when the assigned peer asks to connect to its port. After receiving all the data parts concurrently, the complete file is constructed. The receiver waits a certain amount of time to receive the data and if the peer fails to send the data, the receiver send a negative acknowledgement to the server and tries to receive again from reassigned peers.