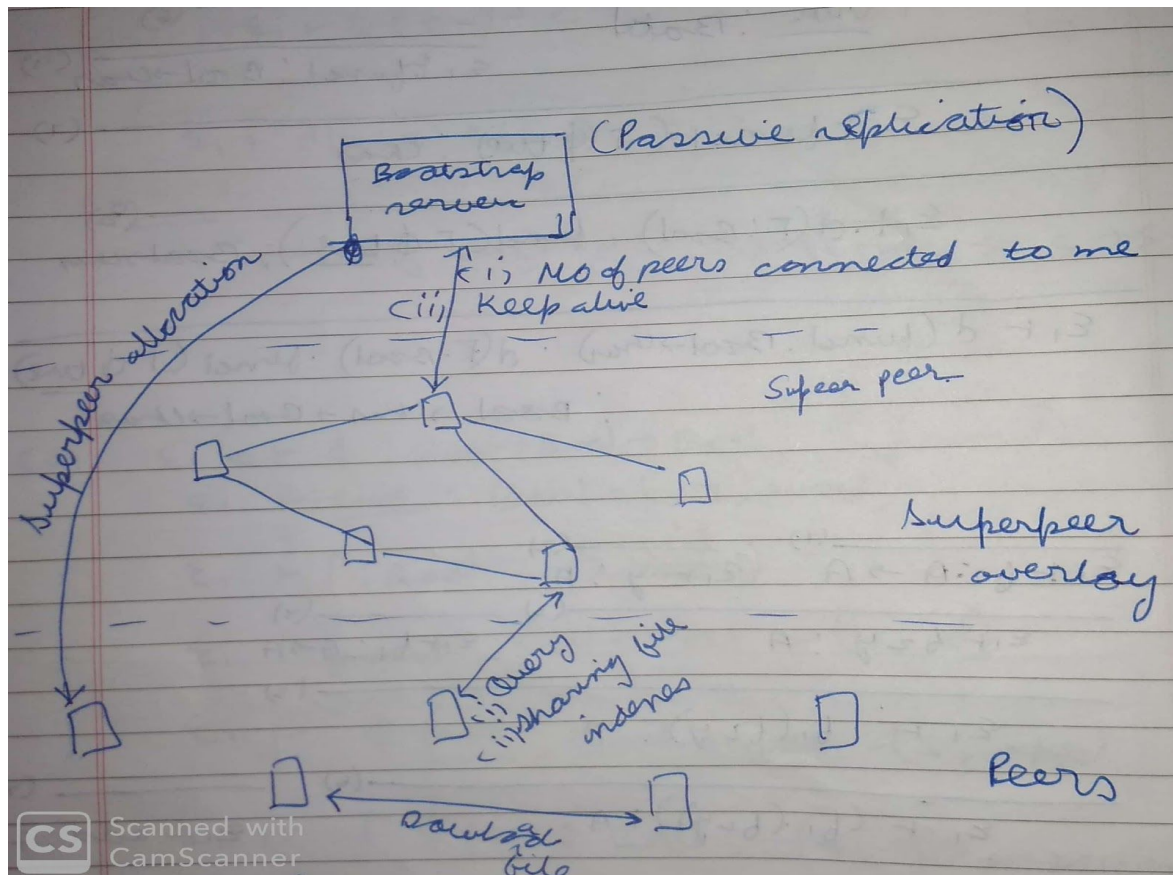


Offered featureset -

- Bootstrapping (Joining the network)
- Query the system for desired files
- Downloading the file from available peers
- Scalable and fault tolerance against node failures.

Architecture -



- Bootstrap server - Stores the list of available super peers and count of peers connected to each . Each new peer contacts the bootstrap server to ask for superpeer allocation. Helps in distributing load over all super peers. To provide reliability to the system, this will be passive replicated. While replying to neighbour queries from

superpeer, ensures that resultant superpeer based overlay is connected graph.

- Superpeer - Each superpeer contains file information about a dynamic subset of all peers in the network. A peer connecting to the superpeer allocated by bootstrap server will provide its file list. For query lookups broadcast based search in superpeer based overlay.
 - Superpeer additionally maintains inverted index(apart from file to peer mappings) to facilitate keyword based searching.
- Peer- Each peer in the system holds data resources to be shared and sends the indexes of these resources to its assigned super peers. Each query response will result in download from hosting peer of the file.

Protocols -

1. Bootstrapping the system -

- Superpeer joining the network -
 - Superpeer P asks the bootstrap server for its neighbours. Server assigns a fixed number of neighbours to P based on latest keep alive messages. Bootstrap server constructs the graph for superpeer based overlay and ensure its connectivity while assigning neighbours to prevent network partitioning.
- Peer joining the network
 - Fixed address of bootstrap server known to the peer P (hardcoded or DNS based hostname). The server assigns super peers(fixed in number) S to P. This ensures P is assigned to at least one superpeer in S if another fails. Each peer regularly check with its assigned super peers to inform about its availability.

2. Peer P to superpeer X connection

- Initial addition to X's hosted peers

- P sends its file lists to X along with their statistics (hash ,number of parts, downloaded part).
 - X adds P to its hosted peer set.
 - Updates to file list
 - On completion of download P sends X the new entry in its listed files.
 - Query based on keyword K to X from P
 - P sends request to one of X on which P is hosted.
 - X does a broadcast based search for K on the overlay.
 - Each superpeer on request searches the local inverted index and replies to P
 - Query based on file identifier I to X from P
 - Based on keyword based replies P choses a file identifier I associated with K (in last step) and queries X.
 - X does a broadcast based search for I.
 - Each peer containing I replies P with associated peers (IP addr:port).
3. Superpeer S to Superpeer X connection (Searching)
- S on receiving request for keyword lookup checks its local inverted index, replies to peer who requested the query if available and forwards the query to its neighbours X.
4. Downloading Peer D to Uploading Peer U connection (Downloading)
- After receiving peer set associated with a file identifier, D connects to each of U. Each U sends a bitset indicating available parts of the file on U. The associated swarm of peer with this file uploads and downloads to each requesting peer and serving peer. Each peer after completion of download of piece check its hash against the published hash of the piece.
 - Ensuring availability
 1. Rarest piece first
 - a. Each D for each part of the file sends request for least available part.

■ Ensuring fairness

1. Choking algo

- a. Initially each connected peer to U is choked.
A peer interested in downloading asks U to unchoke it. U unchokes all the peers which provides highest download rate to U (U may also be downloading some part from the interested peer).

2. Optimistic unchoke

- U randomly chooses an interested peer at regular interval to unchoke (To provide fairness to peer with no parts available which U is interested in).