Computer Networking Scribing for Nov 8th, 2021

Peer-to-peer (P2P)

Web 2.0 vs Web 3.0
Web 2.0 is current client-server model of web communication,
what if communication was direct between who we currently view as clients? This would be “Web 3.0”, or the potential future of the web, decentralized.

In Web 3.0, nodes communicate with each other.

File distribution time
For file with size F to N peers, with d being the download speed and u being the upload speed.
Client-server

time to distribute F to N clients using client-server approach: $D_{c\text{-}s} \geq \max\{NF/u_s, F/d_{min}\}$

increases linearly in N

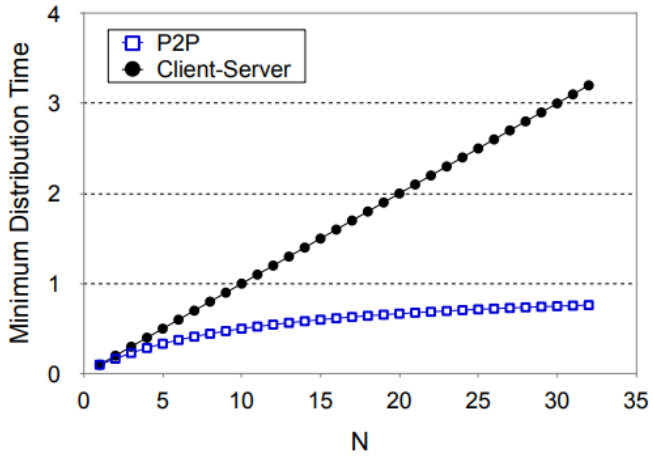
P2P

time to distribute F to N clients using P2P approach: $D_{P2P} \geq \max\{F/u_s, F/d_{min}, NF/(u_s + \Sigma u_i)\}$

increases linearly in N …
… but so does this, as each peer brings service capacity

Example of client-server vs P2P in regards to minimum distribution time as the number of clients grows (N)

client upload rate = u, F/u = 1 hour, $u_s = 10u$, d_{min}



BitTorrent
Follows a P2P file distribution, with a file (such as a movie) divided into 256Kb chunks distributed among various peers. In order to know who has what part of the movie, there exist **trackers**, servers with databases of where the chunks of file X exist (which IP addresses).
Problem? What if federal agents take down a given tracker server? Too centralized, single point of failure.
Solutions

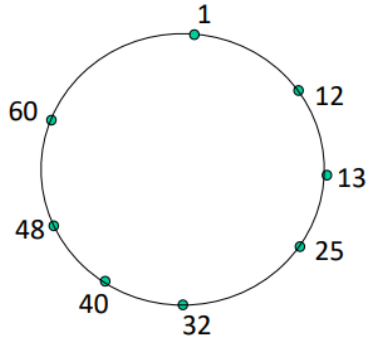
- Distributed Hash Tables (DHT)
- Peer Exchange (PEX)
- Magnet links

Distributed Hash Table (DHT)
Original key: movie title, key: hash(original key), value: IP address,
A hashed key is used because it allows for more convenient store and search of files, as well as confirmation that the downloaded file hashes to said key.

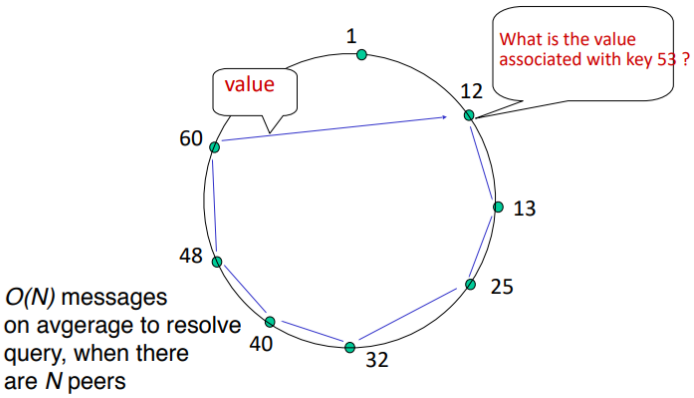
- Distribute (key, value) pairs over millions of peers
- Any peer can query database with a key
- Each peer only knows about a small number of other peers
- Robust to peers coming and going (churn)

Assign key-value pairs to peers

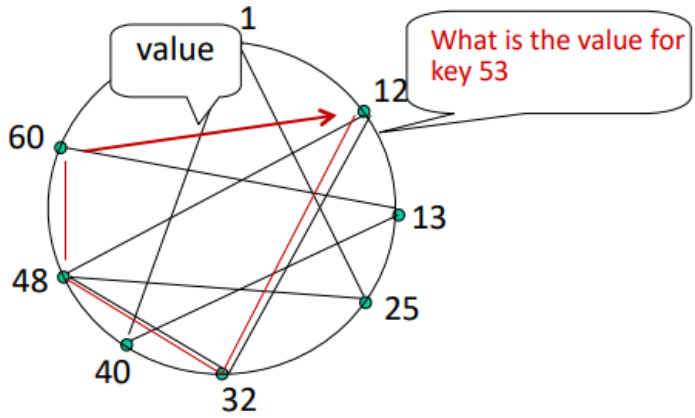
- Rule: assign key-value pair to the peer that has the *closest* ID
- Convention: closest is the immediate successor of the key
- E.g., ID space {0,1,2,3,...,63}
- Suppose 8 peers: 1, 12, 13, 25, 32, 40, 48, 60
    - If key=51, then assigned to peer 60
    - If key=60, then assigned to peer 60
    - If key=61, then assigned to peer 1



Resolving a query (notice how we are looping around the circle)

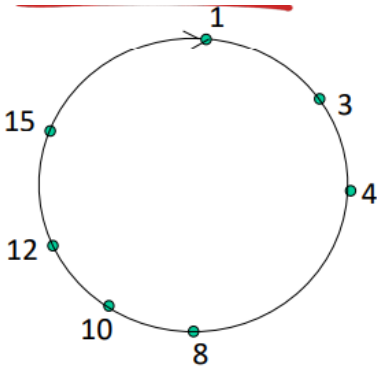


Using shortcuts to cut down on query time making O(log(N)) possible



How do we handle peer churn? (Peer churn is peers coming and going)

- Each peer knows address of its two successors
- Each peer periodically pings its two successors to check aliveness
- If immediate successor leaves, choose next successor as new immediate successor



Ex:

- peer 4 detects peer 5's departure; makes 8 its immediate successor

- 4 asks 8 who its immediate successor is; makes 8's immediate successor its second successor

Sources: Professor Graham's lectures as well as Author's presentation slides