(Abstract) Project proposal for the 2023/2024 fall semester

Background

This project proposal is a continuation of a project from last semester. Next we will present a brief overview of the project and its current progress. We are building a distributed storage system based on a structured p2p network. The main goals of the system are to ensure durable storage by replication and to prevent the main known attacks against such networks. The system works by splitting the nodes in 2 categories - storage nodes and verifier nodes. The verifier nodes monitor the state of the network and reward/punish storage nodes. Currently, the system supports replicated storage based on the Kademlia protocol.

In the current state of the system the verifier node is only one, which makes it both a hot-spot and the main vulnerability.

Next steps

In the next version of the system the main goals will be: * Decentralize the Verifier node * Implement a mechanism where the Verifier nodes have to rotate the storage nodes they are responsible for to prevent malicious Verifiers * As a part of the previous step - decentralize the reward system, so no 1 verifier can make decisions on their own * Implement file invalidation * Implement optional mechanism for files stored in the network, so only the client who stored the file can read it * Optimize the verification mechanism with partial hashes of files * Add a crypto puzzle to the flow of peers joining the network

Optional next steps

As these steps are either unclear or too far-fetched for the scope of this proposal, we are putting them in a different category

- Add signature to files
- Enable file editing/early deletion
- Experiment with ideas for the award mechanisms to determine which works best:
  - Longer storage period → More rewards
  - Faster download → More rewards
  - More popular file → More rewards
- Use hierarchical naming and allow Keeper nodes to have multiple branches of the storage tree. This way, Keeper nodes can select what files they want to store.

Normal operational flow

For context, we include the main flow of the system:

- We start with at least:
  - 1 Keeper node
  - 1 Verifier node
  - 1 Client who wants to store a file
- The Client contacts the Verifier with a request to store a file, for 10 days
- The Verifier proposes a contract, which will cost the Client X number of tokens to store the file for that period
- The Client accepts
- The Verifier takes the file and contacts Keeper nodes, offering them a contract to store the given file for 10 days for Y number of tokens
- The Verifier distributes the file to the Keeper nodes that accept the contract
- The Verifier creates a hash of the file and verifies that the Keeper nodes have the file by asking them to send the Verifier the hash of the file. This check occurs regularly
- The Verifier chooses another 2 Verifiers (based on proximity in the ID space), which should also hold the hash of the file
- The Client is informed that the contract is complete and is given the IDs of the Verifiers that know where the file is stored.
- If the Client wants to retrieve the file, they contact the Verifiers, which forward the request to the Keeper nodes
- If the Client wishes to store the file for longer, they need to establish a new contract before the 10 days period ends