Mp1 Report

Quang Du(quangdu2), Kevin Hou(khou2)

Implementation of Message Delays:

Our method of delay implementation involves a queue which holds packets in the order they are created to be sent out.

On the creation of each packet, the current timestamp is taken, and we had a random value between the values of 0 and 3 seconds to it, causing the delay to be up to 3 seconds. This packet is then put into the queue, and when the current time is larger than the send time, the packet will be sent through our udp socket.

Implementation of Linearizability:

Our method of implementing Linearizability involves the usage of totally ordered broadcasts. To achieve this, we implement a central server, which uses a queue to order the requests from each server.

Using this central server, we implement linearizability by making each server hold new requests when a request is being processed, only releasing upon receiving of an ack from the central server. This guarantees linearizability because each request will be already ordered by the central server.

Implementation of Sequential Consistency:

Our sequential consistency method also employs the usage of totally ordered broadcasts by way of a central server.

Using this central server, all requests to write a key or replace a key will cause the server making the request to hold all other requests. The difference from linearizability being that read requests can be immediately fulfilled without holding by the server's own database.

Implementation of Eventual Consistency W = 1 R = 1

Our implementation of Eventual consistency W=1 R=1 involves each server sending to the one server higher than it in identity order, with the highest server sending to the lowest. It will send a read request then take the value from ack to print. It will also write to the server and wait for an ack before proceeding.

Implementation of Eventual Consistency W = 2 R = 2

The implementation of Eventual consistency W= 2 R = 2 is similar to above, but it will write to the two servers above it in ID. Still looping from highest to lowest. The same implementation will be used, waiting for both to ack from writes. On reads however, it will take the value with the highest timestamp of the two.

Implementation of consistency repair

Our planned implementation of consistency repair will be repair periodically. This will be implemented by having a function which checks the time, running consistency repair every five minutes. Consistency repair will work by having each server one by on request each of their key values, and taking the one with the latest timestamp as the correct one. Consistency repair will not be held by queues except for the delay queue from client. Meaning it will be serviced immediately upon receipt by each of the other servers.