

Abstract:

For my capstone, we implemented a file-locking feature for the final assignment for CSCI1380 - Distributed Systems. The project is to implement a file storage system, similar to the ones seen in linux, and provide basic functionality to open, read from, and write to files. We used a version of tapestry, which is a distributed key-value store that we implemented as our second project in the course. For the capstone, we implemented file-locking at a global level, which solves concurrency issues when multiple clients access the file system at the same time.

The file-locking was implemented using Apache Zookeeper's ephemeral sequential nodes along with the file-locking algorithm described in their documentation. We used ephemeral nodes to improve fault tolerance, such that when a client fails or disconnects, the lock that the client holds will automatically be released. This is a crucial feature to prevent the entire system from experiencing a deadlock: when a lock isn't released and everyone else has to wait.

The algorithm implemented also prevents herding (when every client is waiting for the same channel). This is achieved by having each client only listen for the lock request that precedes it to be released. In doing so, Zookeeper doesn't have to inform all clients whenever each lock is released. Instead, it only informs the need to know clients.