15-640 Project 4 Document

Luxiao Ding (id: luxiaod)

Designs:

- The Server has a startCommit method that informs the Server that a new candidate collage has been posted.
- When the Server receives a candidate collage, it writes log right away. Then Server sends a query to commit message to all involving cohorts and receives replies from them. If Server has not received the reply from certain cohort within **3 seconds (timeout)**, Server regards it as a No.
- Cohorts use the **callback** mechanism to receive messages from Server.
- Server uses **getMessage()** method to receive messages from cohorts.
- When a cohort receive a query to commit message, it checks 1) whether the source files exist 2) whether source files are locked 3) the return value of askUser method. Only when all the source files exist and are not locked and the askUser returns true will the cohort vote Yes. Otherwise, it votes No.
- If the cohort votes No, it send back the voting to Server; if it votes Yes, it locks all involving source files, writes log, and sends voting message to Server.
- When the Server receives at least one No, it aborts the transaction. Otherwise, Server commits this transaction. Once the Server makes the decision, it writes log, and then **keeps sending** the decision message to cohorts concerning this transaction **until it receives Acks from them**.
- When a cohort receives a commit decision message from Server, it will first write log, and then delete source files concerning this transaction. It then writes log again and sends Ack message to Server. When a cohort receives an abort decision message from Server, it will first write log, and then remove source files concerning this transaction from a HashSet called locked_files. It then writes log again and sends Ack message to Server.
- For **failure recovery**, there is a readSnapshot() method in both the Server and cohorts. Every time a Server or a cohorts starts/restarts, it will first check whether there is a log called “snapshot” in the current directory. If yes, it reads the log and reconstructs its data structure to the state when it wrote log the last time.