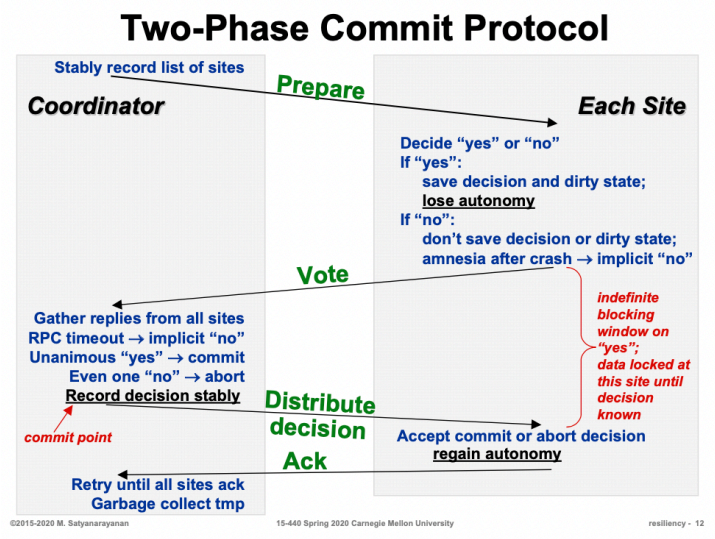
Name: Ting-Yu Lan
AndrewID: tingyula

15-440 Distributed System
Project 4: Two-phase Commit for Group Photo Collage

Protocol between Server and UserNodes



My design follows the lecture note shown above. The server would send prepare messages to user nodes, gather all votes, and make a final decision. As to users, simply lock sources if they vote yes, and reply to the server's final decision.

The server is responsible for counting how many votes it requires to get, and set timeout for votes. If there is a false vote, including any timeout vote, the final decision would be false. If all users send true back to the server before time out, the final decision would be true.

User nodes have to lock sources while waiting for a final decision from the server.

Timeout thresholds

According to the handout, a message is guaranteed to arrive within 3 seconds if it is not lost. Thus, I set a timeout threshold for 6 seconds since we need to measure a round-trip time.

Handle lost message

If a voting message from a user node to the server is lost, simply assume the user vote for false. As to the ACK message, simply resend the server's final decision to user nodes if the server does not get its ACK reply.

If a preparing message sending from the server to the user is lost, it would result in not receiving a vote from that user. It goes back to the situation discussed above.
If a final decision message is lost, again, back to the above scenario. The server would not get an ACK from the user node, so the server would resend the final decision to the user node

Name: Ting-Yu Lan
AndrewID: tingyula

Recover from node failure
I record each step info into a log file on disk. When a node fails, it would first read those log files when it restarts.

Server crashed
A server would lose all commit records if it crashed. Therefore, it has to reconstruct those records when it restarts. My code simply reads through log files and uses basic information (, including filename, image, and sources) to reconstruct a commit. After reconstructing the commit, read through all log files to find out which stage that its transaction is on. If the transaction has not finished yet and it has gotten all votes from user nodes, simply resend the final decision to all user nodes involved. If the transaction has not gotten votes yet, abort that transaction by directly send the server's final decision as false to all user nodes.

User node crashed
A user node needs to make sure that an image would not be published more than once. Therefore, log files for user nodes record which image has been published. When a user node restarts, it simply read through all log records and makes sure that no further transaction will publish those images again.

Others
I use a serializable object to be messages sent between user nodes and the server.