Distributed Systems – Project 3

Rahul Vanukuri

**Two Phase Distributed Commit (2PC) Protocol**

Two phase distributed commit protocol is used as recovery system, in distributed systems it consists of two phases voting phase and decision phase. The voting phase consists of participant sites voting whether to commit or abort the message. Coordinator sites are involved in the decision phase to decide whether a transaction should be committed or aborted.

For example Transaction T1 starts at site S1, which is the coordinator, and T1 also runs sites S2, S3, and S4, which are participating sites.

Voting phase - In the voting phase, S1 is ready to commit and is waiting for the other sites that are running the same transaction to commit or abort. Now S1 sends a message to S2, S3, and S4, making sure that they are ready or not. So the participants at sites S2, S3, and S4 will send their replies, indicating whether they are ready to commit or not. If any of the participants (S2, S3, and S4) reply that they are not ready, the transaction will be aborted.

Decision phase - In the decision phase, the participant decides whether to commit or abort by following the rules.

1.) If all the participating sites send a reply to the coordinator saying that they are ready to commit, then S1 will commit the transaction (this rule only works when all the participating sites are in the ready phase).

2.) The second rule is that if any of the participants are not ready to commit, then the transaction will be aborted.

So this is how the Two phase distributed commit protocol works

**Part 1**

Even before sending a "prepare" message to the participants, the coordinator is under fail condition and therefore cannot send any message to the participants (nodes). So the output we get as the coordinator is under fail conditions is unknown and will be aborted. Later, after 5 seconds, we brought the coordinator back online, sent the prepared message to the nodes, and got the response "no" from the participants as the coordinator failed to send the message earlier.

**Part 2**

If any of the participants are not ready to commit, then the transaction will be aborted. Coming to Part 2 of the project, if the transaction coordinator does not receive a yes message (to commit) from any of the participant sites, the transaction will be aborted.

Part 3

Transaction information will be stored on disk using a txt file on the local machine. Now the coordinator sends the "prepare" message and gets the response of yes," which means they are prepared to commit, and the coordinator commits the message to all the participant sites. If the coordinator fails to commit at the second node, then all the other nodes, including the second node, will be aborted. Again, the TC will be backed online, start sending "prepare" messages, and commit the transaction from the failed node.

Part 4

In the final part, the node will store all the transaction information in a text file on the local machine. Now, after receiving the response to the "prepared" message from the nodes, the coordinator commits the transaction; if any of the nodes fail, the node will be backed up online, and TC starts sending the prepare message and commit the transaction to only the particular node that has failed to commit.

**Problems encountered:**

We have used the timeout function time.sleep() to wait for the failure nodes to get backups online, which took too long and caused delays in processing a transaction. so we have less timeout to improve the processing time of the overall transaction.

**Outcome:**

Learned how to implement Two Phase Distributed Commit (2PC) Protocol in the distributed systems.

Learned how to use multithreading simentaneously