Distributed Systems, Project3- Assignment2

CSE 5306, Fall 2021 Contributors:

PAVAN KUMAR CHAPARLA – 1001865675, pxc5675@mavs.uta.edu

POLA MANOHAR – 1001980283, mxp0284@mavs.uta.edu

05th December, 2021.

REQUIREMENTS:

Implement a 2-phase distributed commit (2PC) protocol and use controlled and randomly injected failures to study how the 2PC protocol handles node crashes. Assume one coordinator and at least three participants in the 2PC protocol. Similar to the previous projects, we use multiple processes to emulate multiple nodes. Vote requests and responses should be carried out using communications. Each node (both the coordinator and the participants) devises a time-out mechanism when no response is received and transits to either the abort or commit state. Design a controlled failure test to evaluate whether the implemented 2PC protocol leads to consistent states across the coordinator and participants. For simplicity, you can assume that only one node fails in the controlled test. Evaluate different possibilities of failures (e.g., coordinator fails before or after sending vote-commit). To emulate a failure, you can impose a much longer delay at a failed node than the time-out period used by other healthy nodes. Node print their states before termination. Verify all nodes converge to the same state regardless of the failure.

Furthermore, evaluate the 2PC protocol by randomly injecting failures to any nodes (e.g., a node may be delayed emulating a failure with a probability at any point during execution). Verify the terminal state to ensure consistency.

CONCEPTS COVERED IN ASSIGNMENT '2':

- We have analysed the project and worked together to start with the usage of 2-phase commit protocol.

-We went through certain course materials online to get the in-depth knowledge of the concept of 2-phase Commit – protocol before starting with the implementation and shared the tasks among us.

-Created one server class and one client thread class to handle the server and the participants.

-Here the Server class acts as the co-ordinator and communicates with the participants.

CHALLENGES DISCUSSED WHILE DOING ASSIGNMENT 2:

-Implementing the timeout for certain nodes to manually run out of time for the coordinator to send a global-abort for all the other nodes and end the process.

-To timeout the coordinator itself which takes some time to recover and then finishes the process.

ISSUES ENCOUNTERED WHILE IMPLEMENTING:

-The Server which acts as coordinator issuing a global abort to all the other connected nodes when some nodes donot respond within the time-constraint was a bit problem in the beginning as we had to catch the timeout issue implemented at the clientThread class and then proceed accordingly.

-The multicasting implemented using the arrayList concept first created few issues regarding the sizes of the list and then handled them correctly with proper analysis.