DS Project -3 Report

CSE-5306 DISTRIBUTED SYSTEMS 2022 Fall

Professor: **Dr Md Hasanuzzaman Noor**

Teaching Assistant **Mohammad Samiul Arshad**

Group Member: Rithesh Harish Bhat
Student Id: 1002058955

I have neither given nor received unauthorized assistance on this work.

Sign: Rithesh **Date: 01 November 2022**

Abstract

Implemented 2 phase Distributed commit in Python leveraging Flask as a web framework in a multiprocess environment. The Project showcases end-end flow of a 2 phase transaction commit, Transaction coordinator initiates the transaction and sends the “Prepare” message across N nodes and awaits response form each of the nodes, if both nodes acknowledge the prepare message within the timelimit, the transaction will be committed, the project also takes care of scenarios where TC / Node when goes down, still is able to handle the transaction and end it at the right state

Project Description:

- The application is written in Python.
- **All interactions between N nodes are configured in main.py**
- The application is intuitive and interactive, The menu showcases the capabilities the application possesses.
- The application prints out log at the beginning and end of each n/w call regarding the state of transaction.

Deliverables can be tested and verified by selecting options from the Menu ranging from 1 - 5.

- case 1. TC goes down before sending prepare.

- case 2 Simulating node failure, node does not send ack with YES / no to tc
- case 3 happy case.
- case 4: tc reads info from file to send commit info
- case 5: Node reads info from file to get transaction id and fetches commit id from tc

Learnings:

- Aim was to take up ownership and finish the project Solo, which gives me an opportunity to introspect on the concepts that I need to work on. It was a conscious decision to pick Python for Project-3, even though Go was preferred for initial 2 project, just to ensure i remain acquainted with the syntaxes and fundamentals of python.
- Fundamentals of Threads and MultiProcesses.
- Communication b/w Web framework and Processes.
- Develop an interactive application, which takes user inputs and does the intended tasks.
- Understanding Flask Endpoints and API creation for interaction.
- Tried to follow the Single Responsibility principle and ensure each function does 1 task.
- Decoupling and reusing utility functions.

Challenges:

- Having been working in linux in the past, getting used to Windows OS for development was challenging. It took some time to figure out environment variables and PATH configuration of golang binaries.
- Understanding the working of Multiprocess in Python.
- Familiarized with Flask to make it run out of Main Thread.
- Multiple debugging to figure out Multiprocess interaction b/w web framework.

.