DS Project -2 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: 10 October 2022**

Abstract

Implemented Vector clocks for N node distributed system. The communication between nodes is done via socket TCP L4 protocol. I have gained a better understanding towards how goroutines in Golang work. This was essential to ensure the goroutine in the background is always waiting for incoming requests from clients and waiting for user input to decide which node the requests need to be sent. The application publishes the state of the vector clock for each of the nodes involved in the transaction.

Project Description:

- The application is written in Golang.
- **All interactions between N nodes are configured in main.go**
- 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 vector clock [number of events that has occurred]

Deliverables can be tested and verified by selecting options from the Menu ranging from 1 - 3.

- 1. Send a request to server 1 and receive ack.
- 2. Send a request to server 2 and receive ack.
- 3. Send a request to server 3 and receive ack.

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 Go over Python, even though Python is my go to programming language.
- Fundamentals of goroutines.
- Communication b/w goroutines.
- Develop an interactive application, which takes user inputs and does the intended tasks.
- Understanding N/W topology, how TCP connection helps in reusing a connection once it’s been established.
- 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 Channels in Go.
- Familiarized with go packages which played the role of logging and leverage the logger to implement vector clocking.
- Multiple debugging to figure out why the second TCP connection always was in hung state, despite the connection being established.

.