**Distributed Systems**

CSE 5306-002 PROJECT REPORT

**n -NODE AND LOCKING DISTRIBUTED SYSTEM**

**Submitted by**

Shubash Muniyappa (1001915563)

Sai Rohit Kalyan Gandham(1002070724)

Instructor: Jia Rao

**Academic Integrity**

I have neither given nor received unauthorized assistance on this work

Signed:A picture containing diagram

Description automatically generatedDate:11/11/2022

SignedA picture containing lawn mower

Description automatically generatedDate: 11/11/2022

**Introduction**

The main aim of the project is to get a hands-on experience on the concept learnt in the lecture in distributed system.

We implemented a n-node distributed system that supports totally ordering events and distributed locking system.

Technology used: Python 3.9

Packages used: json, socket, time, traceback, Thread, numpy, FileLock.

**Implementation**

**Assignment 1(Lamports Clock)**

* We wrote the entire project in the PYTHON programming language. There are four program files 3 clients and a server in this project.
* We are using localhost port \*\*\*\* on the web server. The server implemented in such a way that it is always running and accessible by multiple clients at the same time for communication between server and clients.
* Client code is initiated and connected to server address using server port. Concept of multiple threads is used for a single client to perform continuous conversation.
* We have initialized a logical lamports to compare the individual lamport clock at client end. For example, if lamports clock of client\_1 C(T1) is less than lamports clock of client\_2C(T2).i.e, C(T1) < C(T2), then the communication will be successful.
* All the messages will be transmitted in FIFO order.

**Assignment 2(Vector Clock)**

* The machine's vector clock is utilized to record events. Each machine has a vector clock that starts off as a list of zeros. Each time a transmit or receive event takes place, the vector's local clock is set to a new value that is increased by 1. We presumed that the distributed system consists of three identical machines.
* The nodes communicated with each other using sockets. Two separate port numbers are assigned to each machine in the system, one for receiving messages and the other for transmitting messages. In each machine, two threads were designated for this function. These threads are in charge of changing the vector clock and operate independently.
* Thread-1: The sender() function is assigned to this thread. The process is waiting for user input. The function invites the user to enter the receiving machine number and the message as soon as the client presses a key. It uses the port address of the receiving computer to deliver the message encoded with the vector clock while also advancing the local clock by one.
* Thread 2: The listener() function is assigned to this thread. As soon as the machine receives a message, the function is called. It merely prints the received message and the vector clock prior to and following its receipt. The local vector clock and the received vector clock are compared, and the greater of the two is used as the new local vector clock.

**Assignment 3(Decentralized File Locking schema)**

* We wrote the entire project in the PYTHON programming language. There are four program files 2clients and 3 servers in this project.
* On the request from the client, server will be able to acknowledged a particular client by specifying the client number and the “OK” message will be sent to the client and incrementing the global count value by 1.
* If all the three servers acknowledge the request from that client, then the global count will be 3 so that the file lock is initiated, and file operation will be performed.
* Once the file is extracted (lock time 10s), the global count will be reset to “0” for the next iteration.
* Once all the servers are up and running the, then all clients will regularly request permission from servers to access the file. For a particular client to access the file, all the servers need to provide the permission to that particular client. Only then, the file will be accessed by the particular client and will be locked to other clients to access at that point time (in our code lock time is 10s).
* **Challenges**
* Establishing UDP and TCP communications using sockets in python.
* Implementing file Lock and unlock system in Assignment\_3.
* Understanding how the clock (Lamport and Vector) changes from client to client

**Observations**

* We can initiate multiple clients to simultaneously request for multiple communications.
* We observed difference between TCP and UDP communications using sockets.
* Mainly we learnt, how the total order multicasting is used in communication between multiple clients.
* From the assignment\_3 we got to know about the decentralized and decentralized locking schema of a file.
* We noticed how the Lamport and vector clock helps us to achieve total order multicasting and implement in python.
* Distinguishing between logical and physical clocks. And the significance of logical clocks over physical clocks.

**Contributions**

* Server-Client communications using TCP in Assignment\_1 is implemented by Shubash.
* Server-Client communications using UDP in Assignment\_2 and Assignment\_3 is implemented by Rohit.
* Lamports clock and vector clock algorithm is implemented by Rohit.
* Decentralized file locking schema is implemented by Shubash.
* Readme file and report was of a mutual contribution.