Report

Project 2: Synchronization

Team Members:

Jaymit Desai 1001447748

Priyanka Jadhav 1001398123

Contents

[Overall Status 2](#_Toc467340538)

[Implementation 2](#_Toc467340539)

[Issues 2](#_Toc467340540)

# Overall Status

The project is completed by implementing berkeley algorithm to obtain initial agreement on the clock and vector clocks for total ordered multicast.

We have also performed Bonus project that adds the feature of distributed locking.

# Implementation

We have created five files for implementing the project.

1. **CoordinatorProcess class:**

This class contains the main method which creates an object of ProcessClass and sets it as coordinator.

1. **SendThread class:**

This is a thread class that implements Runnable interface. The constructor initializes all the instance variables present in the class. The run method calls the startSend method which wraps the timestamp into message by calling sendMsg method and sends it to other processes. For sending message we have used DatagramPacket class.

1. **RecieveThread class:**

This is a thread class that implements Runnable interface. The constructor initializes all the instance variables present in the class. The run method calls the startRecieving method which receives the timestamp wrapped into a message from a process. This class handles vector clocks by calling method handleVC and updateVector.

1. **Message class:**

This class defines the structure and property of a message.

1. **ProcessClass class:**

This is a thread which creates a process and implements Berkeley algorithm. The initCoordinator method makes the process coordinator send its timestamp to other processes in the network. This class also calculates the average time after receiving the timestamp from other slave processors and then forwards the new timestamp to back to the slave processors. A modified version of ProcessClass is used in Assignment 3 which implements Centralized Mutual Exclusion algorithm.

1. SlaveProcess class : This class is used to spawn Slave Processes.

# Issues

1. A mistake while programming was causing the coordinator to keep waiting for process response. This issue was fixed by removing the additional receive calls in the DatagramSocket.
2. In the vector clocks implementation, the process was getting ended even before it could get all the messages from different processes. This issue was fixed by putting a while loop which waits till messages from all processes is received.
3. In the Assignment 3 the Counter.property file created in Mac was not readable by java program because java uses encoding format used in Windows. The problem was fixed by providing a copy of the file Counter.property