Executive Summary

Assignment Overview:

Implementation of the Remote Procedure Calls and Multithreading for a storage application, maintaining data consistency across distributed servers. The client, servers will try to connect over RPC and communicate the data. For RPC communication, I have used RMI (Remote Method Invocation) which is a mechanism that allows one Java Virtual Machine (JVM) to invoke methods on an object running in another JVM. It functions by allowing remote communication between programs written in Java. The locking is handled by the default synchronised() block in Java. 2 Phase commit consensus algorithm is implemented among the participants and coordinator.

Multithreading helps to run multiple threads of a process and increases the concurrency in the application also brings the issue of handling the synchronization in the system. In the application, I handled the issue of synchronization by setting the critical section for multiple data key stores. And threads can access different resources parallelly if there is no conflict with the critical section. The resources are being utilized with the help of RPC communication between client and server. The RMI library is used to register the shared resource and lets the client access the data resource provided by the client.

Technical Overview:

Remote Procedure Calls (RPC) are a protocol that allows a computer program to cause procedures to execute on another address space i.e., on another physical machine. This protocol abstracts complexity, enabling a seamless communication process between client and server. In RPC, a client sends a request message to a remote server to execute a specified procedure using the arguments supplied. The server responds by executing the function and sending the results back to the client.

Multithreading in client-server communication allows simultaneous processing of multiple client requests, enhancing performance and responsiveness. By allocating separate threads for each request, servers can handle interactions concurrently, preventing bottlenecks associated with single-threaded models. This concurrency optimizes resource utilization, with threads sharing memory space within the same process, ensuring efficient communication and quicker response times. Moreover, multithreading ensures fault tolerance, if one thread fails, others continue, preventing server-wide disruptions. However, despite these advantages, multithreading introduces complexity, necessitating careful programming to avoid synchronization issues, deadlocks, and resource contention. This enhances concurrency, leading to faster response times and optimal resource utilization. Multi-threading allows clients to interact independently, even when one client's operation is ongoing.

The two phase commit protocol is initiated by the coordinator upon a user request made by the client to one of the participants. Then the coordinator instantiates the algorithm by broadcasting a request for a commit vote from all the participants. If any participant denies the operation the transaction is reverted and the client is displayed with a proper message. Else the transaction will be instantiated by execution of the operation on all the servers. After all participants commit the transaction, the committed signal is propagated back to the coordinator by each participant. Finally, the acknowledgement is sent to the client by the initial connected participant.

READ.MD

Assumptions:

- Key entered for the store should be string and should not contain any spaces in between.
- Value entered for the store should be integer.
- Time to live for the client expecting response from the server is 5 seconds.
- Client sends a uniques message id which is appended in the request message. To handle unrequested packet.
- All user input commands should be in uppercase (PUT/DELETE/GET). Server will say Invalid command if user tries to enter otherwise.
- Store is initialized with seed data of multiple values.
- Server should be started before the client.

How to start server application:

- start the serverApplication file which provides the RMI of the participants to the client and registers in the registry.
- The server serverApplication should be started with command line arguments i.e a list of port numbers of different participants.

How to start client application:

- The client will be expecting the port number of the server i.e participant inorder to connect, select one of the earlier provided command line argument ports.
- After server is connected, the client mode should be selected.
- UI can be started with different modes "ui", "thread", "exit"
- ui takes to interactive mode.
- thread runs multiple thread of client to run the application.
- exit quits the client application.

Steps to use:

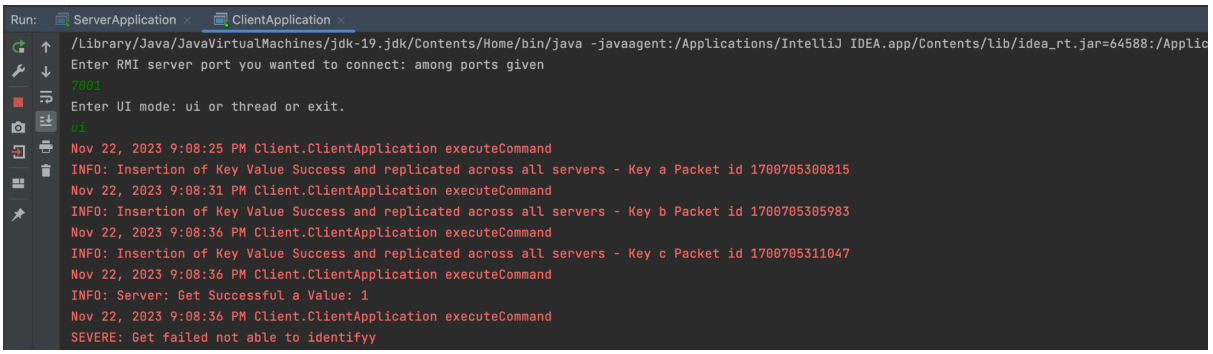
- Every time an operation is performed on the client or server, the log will be ClientLog.log and ServerLog.log in corresponding packages.
- Please enter the user commands in the client application terminal.

```
PUT a 2
GET a
DELETE a
```

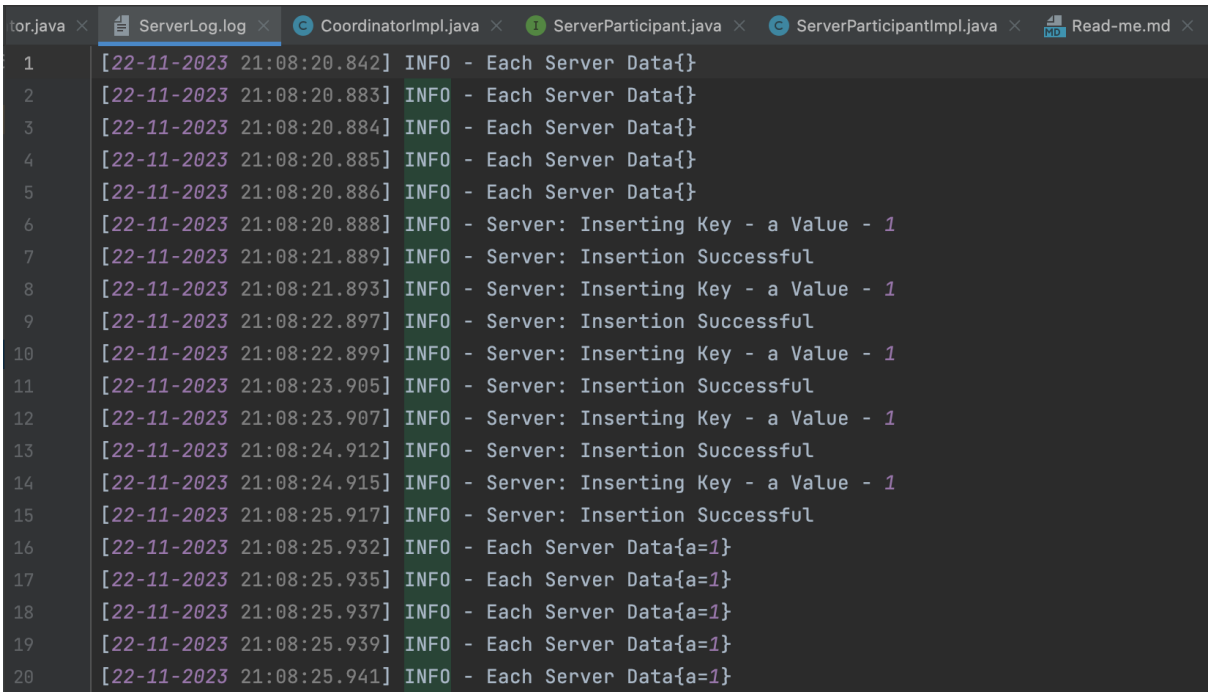
- The logs contains the data of the server information, error, warnings, also the store details of each server.

Server Application Initial and Final States:

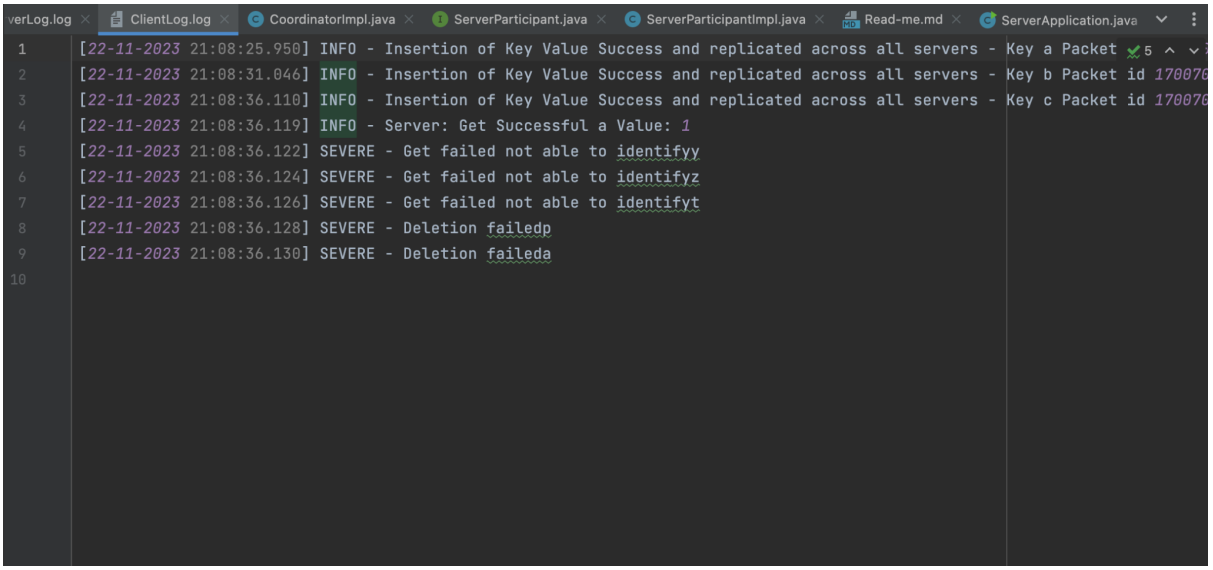
Client Application:



Server-log:



Client-Log:


verLog.log
ClientLog.log
CoordinatorImpl.java
ServerParticipant.java
ServerParticipantImpl.java
Read-me.md
ServerApplication.java
[22-11-2023 21:08:25.950] INFO - Insertion of Key Value Success and replicated across all servers - Key a Packet
[22-11-2023 21:08:31.046] INFO - Insertion of Key Value Success and replicated across all servers - Key b Packet id
[22-11-2023 21:08:36.110] INFO - Insertion of Key Value Success and replicated across all servers - Key c Packet id
[22-11-2023 21:08:36.119] INFO - Server: Get Successful a Value: 1
[22-11-2023 21:08:36.122] SEVERE - Get failed not able to identifyy
[22-11-2023 21:08:36.124] SEVERE - Get failed not able to identifyz
[22-11-2023 21:08:36.126] SEVERE - Get failed not able to identifyt
[22-11-2023 21:08:36.128] SEVERE - Deletion failedp
[22-11-2023 21:08:36.130] SEVERE - Deletion faileda