**A black and white logo

Description automatically generated**

**Distributed Systems**

**Assignment 2**

|  |
| --- |
| Name: Mostapha Abdulaziz Abdullah. |
| Specialization: AI A3. |
| ID: 227824 |

**Bue Commercial Bank**

**Introduction:**

The banking system aims to handle client transactions across multiple branches. It consists of two master servers responsible for coordinating client operations and worker servers handling client data and transactions. Additionally, the system incorporates redundancy by replicating master servers to ensure fault tolerance.

**Advantages:**

Fault Tolerance: By replicating master servers, the system enhances fault tolerance. If one master server fails, the other can seamlessly take over, ensuring uninterrupted service availability. This redundancy minimizes the impact of server failures on client operations this happens by sending and updating every single transaction or operation the client does in the system it sends the updated value only doesn’t send the whole map of clients.

Scalability: The system is designed to scale horizontally by adding more worker servers and branches. Each branch can accommodate a specific number of clients in the code I initialized 5 clients for every worker server , and additional branches can be added to handle increased client demands without significantly affecting the system's performance.

Concurrency Control: The use of synchronized methods in the WorkerServerImpl class ensures thread safety, preventing race conditons and data corruption during concurent client operations. This concurrency control mechanism guarantees the consistency of client balances and transactions.

**Disadvantages:**

Increased Complexity: The introduction of replication adds complexity to the system, requiring mechanisms for maintaining consistency between master servers. Implementeng and managing replication logic can be challenging and may introduce additional points of failure.

Synchronization Overhead: While synchronized methods ensure thread safety, they can also introduce performance overhead, especially in high-concurrency environments. As synchronezation locks access to critical sections of code, it may lead to contention and decreased throughput under heavy loads.

**Master Server Replication:**

In the provided implementation, the first master server acts as a replica for the second one, and vice versa. This replication is achieved through the otherMasterServer attribute in the MasterServerImpl class. Each master server maintains a reference to the other, allowing them to exchange client data updates in real-time.

**Scenarios and Resilience:**

**Server Failure:** If one master server fails due to hardware or software issues, the other master server can continue to serve client requests seamlessly. Clients connected to the failed server will automatically be redirected to the operational one, ensuring uninterrupted service. Not applied in the code as doctor didn’t ask for but just mentioning that this can be achieved.

* **Worker Server Failure**: In case of a worker server failure, the master server can redistribute client operations to other available worker servers. The system's fault tolerance ensures that client transactions can still be processed without loss of data or service disruption.
* **Network Partitioning:** If network issues cause a partition between master servers, each server can continue to operate independently, serving clients within its branch. Once the network partition is resolved, mechanisms such as reconciliation can be employed to synchronize data between the master servers and ensure consistency across branches.

Note: to achieve the previous points some slight changes need to be done on the code but it can be done overall.

**Conclusion:**

The banking system offers several advantages, including fault tolerance, scalability, and concurrency control. However, it also presents challenges such as increased complixity and synchronization overhead. By replicating master servers and implementing fault-tolerant mechanisms, the system aims to provide reliable and efficient banking services even in the face of failures or network issues.

* **GUI:**

1. **Client GUI**

**A screenshot of a computer

Description automatically generated**

**Snapshot for the execution from the client GUI for all the operations in the system.**

1. **System’s GUI**

**A screenshot of a computer

Description automatically generated**

**Snapshot for the systems GUI when starting the system and initializing the connections for the master server one which connects sharmelsheikh , hurghada and elgona worker servers.**

1. **Console’s GUI**

**A screenshot of a computer

Description automatically generated**

**Snapshot for the execution from the client GUI for all the operations in the system.**