Ira A. Fulton Schools of Engineering
Arizona State University

CSE 531: Distributed and Multiprocessor Operating Systems

# Client-Centric Consistency Written Report

## Problem Statement

Implementation of a client-centric consistency model in a distributed banking system. By implementing “read-your-writes” consistency we have to ensure that a customer’s read operations always reflect the latest write operations performed by the customer even if they were in different branches.

## Goal

The goal of the project is to understand various client-centric consistency models and its applications in a real world scenario, here the “read-your-writes” in a distributed banking system. The consistency model aims to provide accurate information to the customers enhancing reliability and trust in the banking system.

## Setup

Tech Stack:

- **Python** 3.9
- **gRPC** 1.58.0
- **Protobuf** 4.24.3
- **IDE** : IDEA IntelliJ 2023.2 Community Edition
- **Operating System** : macOS Ventura 13.5.2
- **Github** : Free,Pro, and Team

## Implementation Processes

1. Modify the protocol buffer messages and remove clock logic from Project 2
2. Use the Protobuf file to generate the gRPC server stub code in python

3. Implement a flag variable to make sure the propagation has been completed to all branches before another read or write operation is performed.
4. If flag is 1 block the operation, else continue
5. Modify the results from Project 2 to form output.json which stores the results of a single customer
6. Execute run_branch and run_customer on the inputs given in Project Description
7. Test the code against input file given as test case in canvas modules
8. Verify the banking operations are functionally correct and validate the test cases with checker scripts
9. Push final code to github with README file explaining the running of the project

## Results

1. **Understanding of gRPC concepts**
    - **Result** : Thorough understanding of the fundamentals of gRPC, such as protocol buffers, service definitions, and remote procedure calls
    - **Justification**: Through the course of the project, students have understood the fundamentals of gRPC which is crucial to understand modern communication protocols used in various software applications
2. **Client-Centric Consistency**
    - **Result**: Implement read-your-writes consistency on the distributed banking system
    - **Justification**: Implementing read-your-writes consistency in Project 3 has helped students understand client-centric consistency models in real world applications.
3. **Understanding Consistency**
    - **Result**: Implement client-centric consistency in the distributed banking system
    - **Justification**: Implementing the client-centric consistency model in the banking system has helped us understand the core concepts of consistency and its various data and client centric models.
4. **Output of the project**
    - **Result:** Project 3 outputs show the client-centric consistency model read-your-writes being implemented with the customer performing operations on different branches and still reading their own latest writes accurately.
    - **Justification:** The output file generated for the project shows that the customer operations

```
(venv) krishnaprasadpa@Krishnaprasads-MacBook-Pro banking-system % python3 checker.py output.json
Consistent balance between branch 1 and branch 2. Balance=10
Consistent balance between branch 2 and branch 3. Balance=20
Consistent balance between branch 3 and branch 4. Balance=30
Consistent balance between branch 4 and branch 5. Balance=40
Consistent balance between branch 5 and branch 6. Balance=50
Consistent balance between branch 6 and branch 7. Balance=60
Consistent balance between branch 7 and branch 8. Balance=70
Consistent balance between branch 8 and branch 9. Balance=80
Consistent balance between branch 9 and branch 10. Balance=90
Consistent balance between branch 10 and branch 1. Balance=100
Consistent balance between branch 1 and branch 2. Balance=90
Consistent balance between branch 2 and branch 3. Balance=80
Consistent balance between branch 3 and branch 4. Balance=70
Consistent balance between branch 4 and branch 5. Balance=60
Consistent balance between branch 5 and branch 6. Balance=50
Consistent balance between branch 6 and branch 7. Balance=40
Consistent balance between branch 7 and branch 8. Balance=30
Consistent balance between branch 8 and branch 9. Balance=20
Consistent balance between branch 9 and branch 10. Balance=10
19 out of 19 cross-branch query events are correct.
(venv) krishnaprasadpa@Krishnaprasads-MacBook-Pro banking-system %
```