
Project 3: Client-Centric Consistency

CSE 531- Distributed Systems

1. Summary

The goal of this project is to implement the client-centric consistency model on top of Project 1. Your job is to implement the essential functions that enforce a client-centric consistency model, read-your-writes, of the replicated data in the bank.

Major tasks:

1. Track read and write events by the same customer across different branch processes
2. Implement read-your-writes consistency among the branch processes

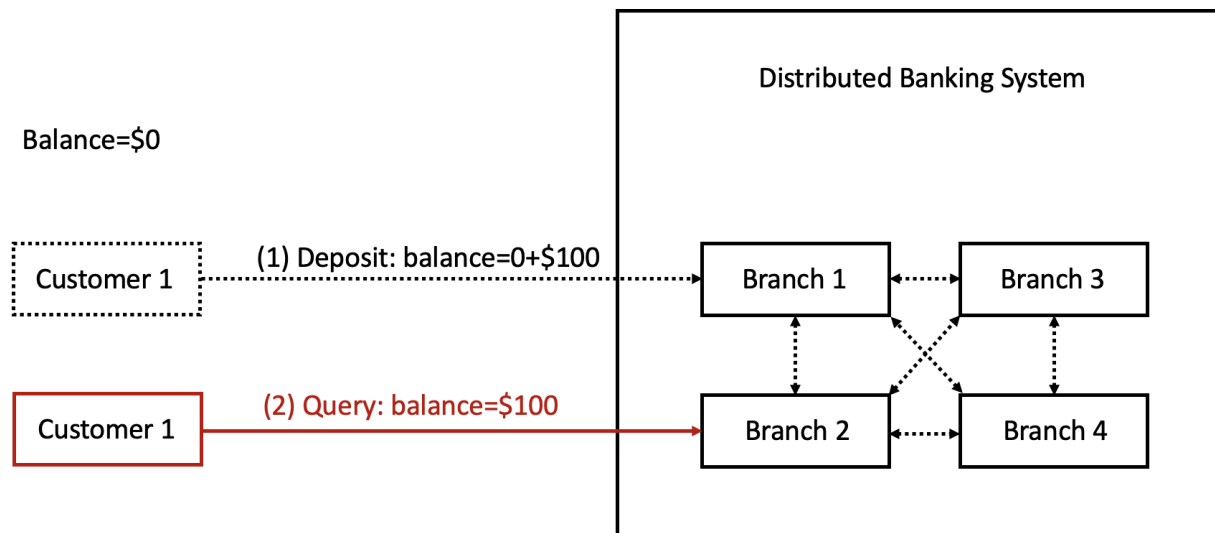


Diagram A: The customer changes branches while submitting requests to the Bank.

2. Description

In this project, the same customer process performs operations on the bank account through different branches. For example, as shown in Diagram A, Customer 1 performs the following operations on a bank account with initially \$0 balance: deposits \$100 to the bank account at Branch 1 and query the balance at Branch 2. The query should return \$100.

To achieve this, you are tasked to implement the **read-your-writes** consistency model that we have learned in class. A data store (the distributed bank system) is said to provide **read-your-writes** consistency if the effect of a write operation (deposit or withdraw) by a process on data item x (balance) will always be seen by a successive read operation (query) on x by the same process.

3. Input and Output

The input file contains one Customer and multiple Branch processes. The format of the input file follows Project 1, with the addition of a branch parameter that indicates the target branch for each event.

```
[ // array of processes
  { // Customer process #1
    "id" : {a unique identifier of a customer},
    "type" : "customer",
    "events" : [{ "id" : {unique identifier of an event}, "interface" : {query | deposit | withdraw},
    "money" : {an integer value}, "branch" : {a unique identifier of the branch} }]}
  { // Branch process #1
    "id" : {a unique identifier of a branch},
    "type" : "branch"
    "balance" : {the initial amount of money stored in the branch}
  }
  { ... } // Branch process #2
  { ... } // Branch process #3
  { ... } // Branch process #N
]
```

Output file format for the read-your-writes.

```
[ // array of customers
  { // Customer process #1
    "id" : { a unique identifier of a customer }
    "recv" : [a list of successful returns of the events from the customer process]
  }
]
```

Below is the example input file and the expected output file.

Example input file

```
[
  {
    "id" : 1,
    "type" : "customer",
    "events" : [{ "id":1, "interface":"deposit", "money":400, "branch":1 }, { "id": 2, "interface":"query", "branch":1 }, { "id": 3, "interface":"query", "branch": 2} ],
  },
  {
    "id" : 1,
    "type" : "branch",
    "balance" : 0
  },
  {
    "id" : 2,
    "type" : "branch",
    "balance" : 0
  }
]
```

Expected output file:

```
[
  {"id":1,"recv":[{"interface":"deposit","branch":1,"result":"success"}]},
  {"id":1,"recv":[{"interface":"query","branch":1,"balance":400}]},
  {"id":1,"recv":[{"interface":"query","branch":2,"balance":400}]}
]
```

4. Submission Directions for Project Deliverables

You must submit your deliverables in the designated submission space in the course. Students may **not** email or use other means to submit the project for course team review and grading. Students should review the academic integrity and plagiarism policies prior to beginning this project.

Your Client-Centric Consistency Project includes **two (2)** deliverables. You may submit the code an unlimited number of times prior to the deadline.

1. **Client-Centric Consistency Project Written Report:** Your written report must be a single PDF with the correct naming convention: Your Name_Client-Centric Consistency_Written Report.
2. **Client-Centric Consistency Project Code:** The project code must be submitted as a .zip file of your source code.

Final submissions missing either of the deliverables will be graded based on what was submitted against the rubric criteria. Please review the rubric for how your Client-Centric Consistency Project will be graded.

Late submission will not be accepted.