

CSE-531_Project-2 ---Logic Clock Project
Yun Shing Lu

1. Project Statement

For this project, I am going to implement an interface using gRPC to build a distributed banking system so the customers as the client side and the branches as the server side can communicate with each other. By doing this framework, customers can withdraw, deposit, and query money from multiple branches in the bank.

The differences between Project1 and Project2 is I am going to focus on logic clock Implementation so make sure all the events from customers and branches will be ordered and propagate the message to the right target.

2. Project goal

The goal for this project is that we are going to pass the JSON file which contains the multiple events and ids for customers and the account balance for each branch. Our banking system will follow the instructions from the input JSON file and return the corresponding output txt file which contains the logic clock number for all the events.

3. Setup

The relevant technologies I used are as below:

Development environment:

Python: 3.8.8

IDE: VScode

anaconda: 2021.05

Python packages:

Multiprocessing

concurrent

grpc: 1.32.0

4. Implementation Processes

Step1: add the update local clock in Customer.py

```
# Update local clock
clock = max(clock, response.clock) + 1
```

Step2: Created 2 functions in Branch.py

1. Event_Receive : receive event from customers or branches
2. Event_Propagate: propagate event to other branches

```

def Event_Receive(self, request):
    self.clock = max(self.clock, request.clock) + 1
    self.events.append({"customer-request-id": request.id, "logical_clock": self
# Receive propagated event from Branch (max + 1)
def Event_Propagate(self, request):
    self.clock = max(self.clock, request.clock) + 1
    self.events.append({"customer-request-id": request.id, "logical_clock": self

```

Step3: Separate branch and customer into 2 services in main.py

For serverBranch, it only generates events related to customers

For serverBranchforBranch, it only generates events related to branches

Step4: Write the output result into JSON file

1. Write the customer event lists
2. Write the branch event lists
3. Combine customer event lists and branch event lists as total logical ordered events in the third part

5. Results

```

1  [
2      {
3          "id":1,
4          "type":"customer",
5          "events":[
6              {
7                  "customer-request-id":1,
8                  "logical_clock":1,
9                  "interface":"deposit",
10                 "comment":"event_send from customer 1"
11             },
12             {
13                 "customer-request-id":2,
14                 "logical_clock":2,
15                 "interface":"withdraw",
16                 "comment":"event_send from customer 1"
17             }
18         ]
19     },
20     {
21         "id":2,
22         "type":"customer",
23         "events":[
24             {
25                 "customer-request-id":3.

```