
Project 2: Logical Clock

CSE 531- Distributed Systems

1. Summary

The goal of this project is to implement Lamport's logical clock algorithm upon Project 1. As shown in **Diagram A**, the processes use logical clocks and follow the Lamport's algorithm to coordinate.

Major Tasks

1. Implement logical clock in every customer and branch process
2. Implement Lamport's algorithm for clock coordination among the processes

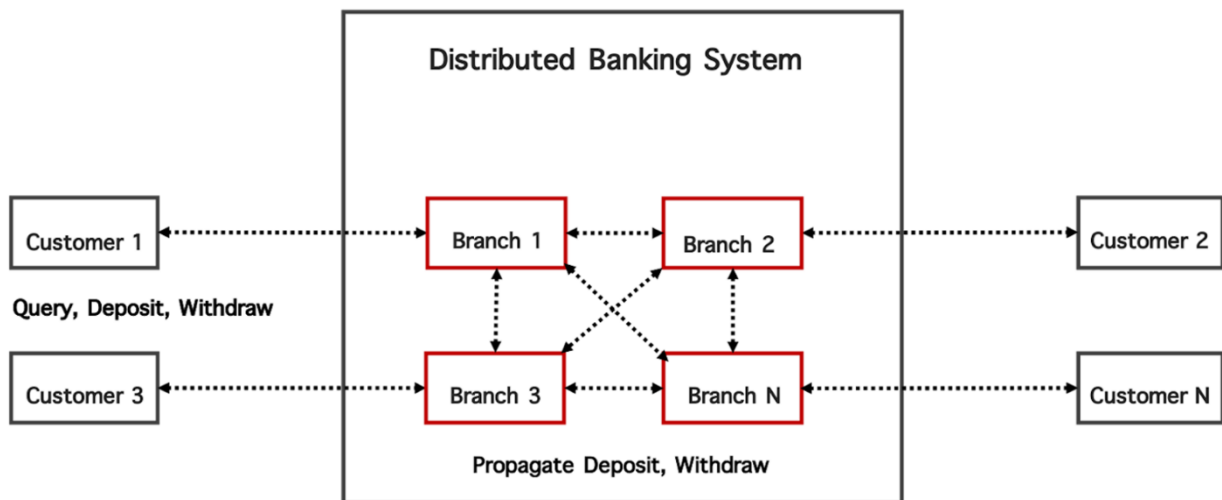


Diagram A: The processes use logical clocks and follow the Lamport's algorithm to coordinate

2. Description

Same as Project 1, customers use Query/Deposit/Withdraw interfaces to issue requests to branches, and branches use Propagate to propagate updates. Every customer Deposit/Withdraw request is handled by one branch which then propagates the update to the other branches.

As illustrated in **Diagram B**, every process maintains its logical lock according to the happens-before relationship between requests 1) of the same process and 2) between send and receive of the same request.

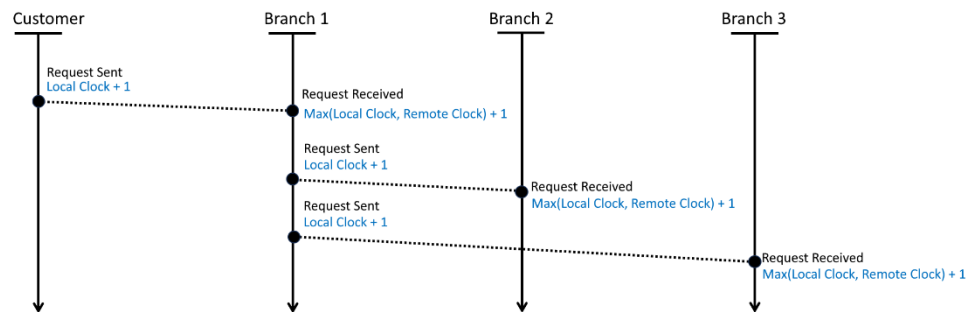


Diagram B: Events triggered by a customer request and their logical times

3. Input and Output

The input file contains a list of Customers and Branch processes. The format of the input file is similar to that of Project 1. Note that the “customer-request-id” parameter in the “events” uniquely identifies every customer request, which needs to be carried over to all the events triggered by this request in order to verify the happens-before relationships are correctly enforced.

```

[ // Start of the Array of Branch and Customer processes
{ // Customer process #1
  "id" : {a unique identifier of a customer},
  "type" : "customer",
  "customer-requests" : [{"customer-request-id": {unique identifier of a customer request} ,
"interface":{deposit | withdraw}, "money": {an integer value} }}
}
{ ... } // Customer process #2
{ ... } // Customer process #3
{ ... } // Customer process #N
{ // 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

```

```
] // End of the Array of Branch and Customer processes
```

The output will contain three parts: (1) all the events (along with their logical times) taken place on each customer, (2) all the events (along with their logical times) taken place on each branch, and (3) all the events (along with their logical times) triggered by each customer Deposit/Withdraw request. For clarity, order all the events from the same customer/branch/customer-request by their logical times in the output.

```
[ // Start of the Array of Customer processes
{ // Customer process #1
  "id" : {a unique identifier of a customer},
  "type": "customer",
  "events": [{ "customer-request-id": {unique identifier of the customer request}, "logical time":
{timestamp from the logical clock }} ...]
}
{ ... } // Customer process #2
{ ... } // Customer process #3
{ ... } // Customer process #N
] // End of the Customer processes
```

```
[ // Start of the Array of Branch processes
{ // Branch process #1
  "id" : {a unique identifier of a Branch},
  "type": "branch",
  "events": [{ "customer-request-id": {a unique identifier of the customer request that
triggered this event}, "logical time": { timestamp from the logical clock }} ...]
}
{ ... } // Branch process #2
{ ... } // Branch process #3
{ ... } // Branch process #N
] // End of the Branch processes
```

```
[ // Start of the Array of events across all processes
{ // Event #1
  "customer-request-id" : {a unique identifier of a customer request}
  "logical time": [{timestamp from the logical clock } ...]
}
```

```
{ ... } // Event #2
{ ... } // Event #3
{ ... } // Event #N
] // End of the Events
```

Example of the input file

```
[
  {
    "id": 1,
    "type": "customer",
    "customer-requests": [
      {
        "customer-request-id ": 1,
        "interface": "deposit",
        "money": 10
      },
      {
        "customer-request-id": 2,
        "interface": "withdraw",
        "money": 10
      }
    ]
  },
  {
    "id": 2,
    "type": "customer",
    "customer-requests": [
      {
        "customer-request-id ": 3,
        "interface": "deposit",
        "money": 10
      },
      {
        "customer-requests": 4,
        "interface": "withdraw",
        "money": 10
      }
    ]
  },
  {
    "id": 3,
    "type": "customer",
    "customer-requests": [
      {
        "customer-requests": 5,
        "interface": "deposit",
        "money": 10
      },
      {
```

```

        "customer-requests ": 6,
        "interface": "withdraw",
        "money": 10
    }
]
},
{
    "id": 1,
    "type": "branch",
    "balance": 400
},
{
    "id": 2,
    "type": "branch",
    "balance": 400
},
{
    "id": 3,
    "type": "branch",
    "balance": 400
}
]

```

Expected output file:

```

// Part 1: List all the events taken place on each customer
[[
  {
    "id":1,
    "type":"customer",
    "events": [
      {"customer-request-id": 1, "logical_clock": 1, "interface": "deposit", "comment": "event_sent from customer 1"},
      {"customer-request-id": 2, "logical_clock": 2, "interface": "withdraw", "comment": "event_sent from customer
1"}]
    },
    {
      "id":2,
      "type":"customer",
      "events": [
        {"customer-request-id": 3, "logical_clock": 1, "interface": "deposit", "comment": "event_sent from customer 2"},
        {"customer-request-id": 4, "logical_clock": 2, "interface": "withdraw", "comment": "event_sent from customer
2"}]
    }
  ],
  {
    "id":3,
    "type":"customer",
    "events": [
      {"customer-request-id": 5, "logical_clock": 1, "interface": "deposit", "comment": "event_sent from customer 3"},
      {"customer-request-id": 6, "logical_clock": 2, "interface": "withdraw", "comment": "event_sent from customer
3"}]
  }
]]

```

```
// Part 2: List all the events taken place on each branch
[[
  "id": 1,
  "type": "branch",
  "events":
    [{"customer-request-id": 1, "logical_clock": 2, "interface": "deposit", "comment": "event_rcv from customer 1"},
    {"customer-request-id": 1, "logical_clock": 3, "interface": "propagate_deposit", "comment": "event_sent to branch 2"},
    {"customer-request-id": 1, "logical_clock": 4, "interface": "propagate_deposit", "comment": "event_sent to branch 3"},
    {"customer-request-id": 3, "logical_clock": 7, "interface": "propagate_deposit", "comment": "event_rcv from branch 2"},
    {"customer-request-id": 5, "logical_clock": 8, "interface": "propagate_deposit", "comment": "event_rcv from branch 3"},
    {"customer-request-id": 2, "logical_clock": 9, "interface": "withdraw", "comment": "event_rcv from customer 1"},
    {"customer-request-id": 2, "logical_clock": 10, "interface": "propagate_withdraw", "comment": "event_sent to branch 2"},
    {"customer-request-id": 2, "logical_clock": 11, "interface": "propagate_withdraw", "comment": "event_sent to branch 3"},
    {"customer-request-id": 4, "logical_clock": 14, "interface": "propagate_withdraw", "comment": "event_rcv from branch 2"},
    {"customer-request-id": 6, "logical_clock": 15, "interface": "propagate_withdraw", "comment": "event_rcv from branch 3"}],
  {
    "id": 2,
    "type": "branch",
    "events":
      [{"customer-request-id": 1, "logical_clock": 4, "interface": "propagate_deposit", "comment": "event_rcv from branch 1"},
      {"customer-request-id": 3, "logical_clock": 5, "interface": "deposit", "comment": "event_rcv from customer 2"},
      {"customer-request-id": 3, "logical_clock": 6, "interface": "propagate_deposit", "comment": "event_sent to branch 1"},
      {"customer-request-id": 3, "logical_clock": 7, "interface": "propagate_deposit", "comment": "event_sent to branch 3"},
      {"customer-request-id": 2, "logical_clock": 11, "interface": "propagate_withdraw", "comment": "event_rcv from branch 1"},
      {"customer-request-id": 4, "logical_clock": 12, "interface": "withdraw", "comment": "event_rcv from customer 2"},
      {"customer-request-id": 4, "logical_clock": 13, "interface": "propagate_withdraw", "comment": "event_sent to branch 1"},
      {"customer-request-id": 5, "logical_clock": 14, "interface": "propagate_deposit", "comment": "event_rcv from branch 3"},
      {"customer-request-id": 4, "logical_clock": 15, "interface": "propagate_withdraw", "comment": "event_sent to branch 3"},
      {"customer-request-id": 6, "logical_clock": 18, "interface": "propagate_withdraw", "comment": "event_rcv from branch 3"}],
    {
      "id": 3,
      "type": "branch",
      "events":
        [{"customer-request-id": 5, "logical_clock": 2, "interface": "deposit", "comment": "event_rcv from customer 3"}],
    }
  ]]
```

```

    {"customer-request-id": 5, "logical_clock": 3, "interface": "propagate_deposit", "comment": "event_sent to branch 1"},
    {"customer-request-id": 1, "logical_clock": 5, "interface": "propagate_deposit", "comment": "event_rcv from branch 1"},
    {"customer-request-id": 5, "logical_clock": 6, "interface": "propagate_deposit", "comment": "event_sent to branch 2"},
    {"customer-request-id": 3, "logical_clock": 8, "interface": "propagate_deposit", "comment": "event_rcv from branch 2"},
    {"customer-request-id": 2, "logical_clock": 12, "interface": "propagate_withdraw", "comment": "event_rcv from branch 1"},
    {"customer-request-id": 6, "logical_clock": 13, "interface": "withdraw", "comment": "event_rcv from customer 3"},
    {"customer-request-id": 6, "logical_clock": 14, "interface": "propagate_withdraw", "comment": "event_sent to branch 1"},
    {"customer-request-id": 4, "logical_clock": 16, "interface": "propagate_withdraw", "comment": "event_rcv from branch 2"},
    {"customer-request-id": 6, "logical_clock": 17, "interface": "propagate_withdraw", "comment": "event_sent to branch 2"}]
}

```

// Part 3: List all the events (along with their logical times) triggered by each customer Deposit/Withdraw request

```

[{"id": 1, "customer-request-id": 1, "type": "customer", "logical_clock": 1, "interface": "deposit", "comment": "event_sent from customer 1"},
{"id": 1, "customer-request-id": 1, "type": "branch", "logical_clock": 2, "interface": "deposit", "comment": "event_rcv from customer 1"},
{"id": 1, "customer-request-id": 1, "type": "branch", "logical_clock": 3, "interface": "propagate_deposit", "comment": "event_sent to branch 2"},
{"id": 2, "customer-request-id": 1, "type": "branch", "logical_clock": 4, "interface": "propagate_deposit", "comment": "event_rcv from branch 1"},
{"id": 1, "customer-request-id": 1, "type": "branch", "logical_clock": 4, "interface": "propagate_deposit", "comment": "event_sent to branch 3"},
{"id": 3, "customer-request-id": 1, "type": "branch", "logical_clock": 5, "interface": "propagate_deposit", "comment": "event_rcv from branch 1"},
{"id": 1, "customer-request-id": 2, "type": "customer", "logical_clock": 2, "interface": "withdraw", "comment": "event_sent from customer 1"},
{"id": 1, "customer-request-id": 2, "type": "branch", "logical_clock": 9, "interface": "withdraw", "comment": "event_rcv from customer 1"},
{"id": 1, "customer-request-id": 2, "type": "branch", "logical_clock": 10, "interface": "propagate_withdraw", "comment": "event_sent to branch 2"},
{"id": 2, "customer-request-id": 2, "type": "branch", "logical_clock": 11, "interface": "propagate_withdraw", "comment": "event_rcv from branch 1"},

```

```
{ "id": 1, "customer-request-id": 2, "type": "branch", "logical_clock": 11, "interface": "propagate_withdraw", "comment":  
"event_sent to branch 3"},  
  
{ "id": 3, "customer-request-id": 2, "type": "branch", "logical_clock": 12, "interface": "propagate_withdraw", "comment":  
"event_rcv from branch 1"},  
  
{ "id": 2, "customer-request-id": 3, "type": "customer", "logical_clock": 1, "interface": "deposit", "comment": "event_sent  
from customer 2"},  
  
{ "id": 2, "customer-request-id": 3, "type": "branch", "logical_clock": 5, "interface": "deposit", "comment": "event_rcv  
from customer 2"},  
  
{ "id": 2, "customer-request-id": 3, "type": "branch", "logical_clock": 6, "interface": "propagate_deposit", "comment":  
"event_sent to branch 1"},  
  
{ "id": 2, "customer-request-id": 3, "type": "branch", "logical_clock": 7, "interface": "propagate_deposit", "comment":  
"event_sent to branch 3"},  
  
{ "id": 1, "customer-request-id": 3, "type": "branch", "logical_clock": 7, "interface": "propagate_deposit", "comment":  
"event_rcv from branch 2"},  
  
{ "id": 3, "customer-request-id": 3, "type": "branch", "logical_clock": 8, "interface": "propagate_deposit", "comment":  
"event_rcv from branch 2"},  
  
{ "id": 2, "customer-request-id": 4, "type": "customer", "logical_clock": 2, "interface": "withdraw", "comment": "event_sent  
from customer 2"},  
  
{ "id": 2, "customer-request-id": 4, "type": "branch", "logical_clock": 12, "interface": "withdraw", "comment": "event_rcv  
from customer 2"},  
  
{ "id": 2, "customer-request-id": 4, "type": "branch", "logical_clock": 13, "interface": "propagate_withdraw", "comment":  
"event_sent to branch 1"},  
  
{ "id": 1, "customer-request-id": 4, "type": "branch", "logical_clock": 14, "interface": "propagate_withdraw", "comment":  
"event_rcv from branch 2"},  
  
{ "id": 2, "customer-request-id": 4, "type": "branch", "logical_clock": 15, "interface": "propagate_withdraw", "comment":  
"event_sent to branch 3"},  
  
{ "id": 3, "customer-request-id": 4, "type": "branch", "logical_clock": 16, "interface": "propagate_withdraw", "comment":  
"event_rcv from branch 2"},  
  
{ "id": 3, "customer-request-id": 5, "type": "customer", "logical_clock": 1, "interface": "deposit", "comment": "event_sent  
from customer 3"},  
  
{ "id": 3, "customer-request-id": 5, "type": "branch", "logical_clock": 2, "interface": "deposit", "comment": "event_rcv  
from customer 3"},  
  
{ "id": 3, "customer-request-id": 5, "type": "branch", "logical_clock": 3, "interface": "propagate_deposit", "comment":  
"event_sent to branch 1"},  
  
{ "id": 3, "customer-request-id": 5, "type": "branch", "logical_clock": 6, "interface": "propagate_deposit", "comment":  
"event_sent to branch 2"},
```



```
{ "id": 1, "customer-request-id": 5, "type": "branch", "logical_clock": 8, "interface": "propagate_deposit", "comment": "event_rcv from branch 3"},  
  
{ "id": 2, "customer-request-id": 5, "type": "branch", "logical_clock": 14, "interface": "propagate_deposit", "comment": "event_rcv from branch 3"},  
  
{ "id": 3, "customer-request-id": 6, "type": "customer", "logical_clock": 2, "interface": "withdraw", "comment": "event_sent from customer 3"},  
  
{ "id": 3, "customer-request-id": 6, "type": "branch", "logical_clock": 13, "interface": "withdraw", "comment": "event_rcv from customer 3"},  
  
{ "id": 3, "customer-request-id": 6, "type": "branch", "logical_clock": 14, "interface": "propagate_withdraw", "comment": "event_sent to branch 1"},  
  
{ "id": 1, "customer-request-id": 6, "type": "branch", "logical_clock": 15, "interface": "propagate_withdraw", "comment": "event_rcv from branch 3"},  
  
{ "id": 3, "customer-request-id": 6, "type": "branch", "logical_clock": 17, "interface": "propagate_withdraw", "comment": "event_sent to branch 2"},  
  
{ "id": 2, "customer-request-id": 6, "type": "branch", "logical_clock": 18, "interface": "propagate_withdraw", "comment": "event_rcv from branch 3"}]
```

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 Logical Clock Project includes **two (2)** deliverables.

1. **Logical Clock Project Written Report:** Your written report must be a single PDF with the correct naming convention: Your Name_Logical_Clock_Written Report.
2. **Logical Clock Project Code:** The project code must be submitted as a .zip file of your source code.
3. **Final Output:** The final output must be a single JSON file with the correct naming convention Your Name_Logical_Clock.

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 Logical Clock Project will be graded.

Rubric

Rubrics communicate specific criteria for evaluation. Prior to starting any graded coursework, you are expected to read through the rubric, so you know how you will be assessed. You are encouraged to self-assess your work and make informed revisions before submitting. Engaging in this learning practice will support you in developing your highest-quality deliverables. *Points may be deducted at the discretion of the course team for disorganized submissions that convolute the grading process.*

Component	Criteria	No Attempt	Undeveloped	Developing	Approaching	Proficient	Exemplary
Test Case 40%	Check the order of the events are correct. All the events should follow happens-before relationship	Provided no response.	About 20% of the events follow the correct order.	About 40% of the event follow the correct order.	About 60% of the events follow the correct order.	About 80% of the events follow the correct order.	All the events follow the correct order.
Code (40%)	The code should contains all the components in Project 1, implementation of logical clock for both customers and branches, and implementation of the logical clock algorithm	Provided no response.	Provided project code that is syntactically or semantically invalid.	<p>Provided project code is functional.</p> <p>Project code performs a few of the functions as described in the final report.</p> <p>Project code is not engineered or designed.</p> <p>No documentation is provided.</p>	<p>Provided project code is functional. Project code performs most of the functions as described in the final report. Project code is thought-out and engineered. Project code provides documentation and code comments (where appropriate).</p>	<p>Provided project code is functional. Project code performs most of the functions as described in the final report. Project code is thought-out and engineered. Project code provides documentation and code comments (where appropriate).</p>	<p>Provided project code that is functional. Project code performs the functions as described in the final report. Project code is excellently engineered. Project code provided helpful documentation and code comments (where appropriate).</p>
Report (20%)	Problem statement and goal	Provided no response.	Provided an incoherent problem statement and goal with no connection to the project description.	Provided a somewhat coherent problem statement and Goal with little or misguided connections with the project description.	Provided a basic, understandable problem statement and goal that loosely connects with the project description.	Provided a clear problem statement and goal that directly connects with the project description.	Provided a focused problem statement and goal that distinctly and in meaningful ways connects with the project description.
	Setup	Provided no response.	Provided an incomplete explanation of the setup.	Provided a reasonable explanation of the setup, but missing steps.	Provided a complete explanation of the setup, but some steps are not working.	Provided complete and understandable explanation of the setup. All steps are working.	Provided complete and clear explanation of the setup. All steps are accurate and working correctly.

	Implementation Processes	Provided no response.	Provided an incomplete explanation and/or one or more of the major tasks may be missing.	Provided a general explanation and implementation. Inaccuracies may be present.	Provided a reasonable explanation and implementation. Steps may be illogical or missing.	Provided a logical explanation and implementation. Steps are logical, but may be disjointed or unrelated to one another.	Provided a sound explanation and implementation. All steps are accurate, related, and working correctly.
	Results	Provided no response.	<p>Provided incorrect or fake results.</p> <p>The explanation is fully incorrect with no merit in approach or connection with the results.</p>	<p>Provides some results and/or unrelated results.</p> <p>The explanation is somewhat incorrect with little merit in approach or connection with the results.</p>	<p>Provides mostly correct results.</p> <p>The explanation is reasonable with some ambiguity.</p>	<p>Results are accurate.</p> <p>The explanation is logical with no ambiguity.</p>	<p>Results are accurate.</p> <p>The explanation is well-developed with strong justifications directly related to the implementation results.</p>