CSE513 Lab2

11/18/2021

Yixin Xu, Tongguang Yu

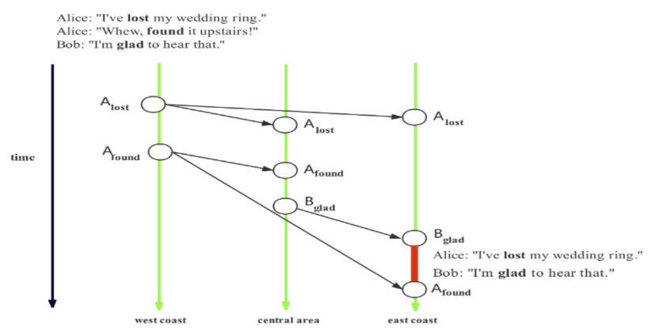
Background Description

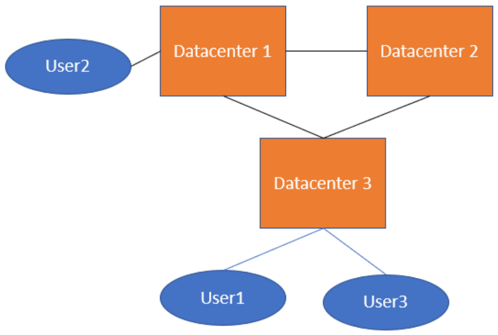
When a distributed system is composed with multiple servers/datacenters and multiple clients. Internet might deliver the result in unexpected order.

To solving the problem earlier, server replicated servers commit replicated requests in causal ordering.

Structure

The structure of our implication is listed below. Server.py will initiate three datacenters and communicate with each other. Client can connect any of the datacenters, then Datacenter will sync the data based on the timestamp generated by Lamport clock.


Alice: "I've **lost** my wedding ring."
Alice: "Whew, **found** it upstairs!"
Bob: "I'm **glad** to hear that."
time
A_{lost}
A_{found}
A_{lost}
A_{found}
B_{glad}
A_{lost}
B_{glad}
Alice: "I've **lost** my wedding ring."
Bob: "I'm **glad** to hear that."
A_{found}
west coast
central area
east coast



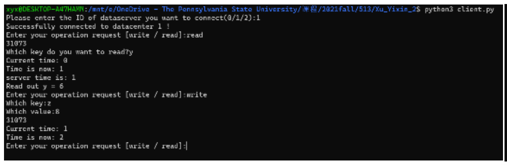
Our program can store the data from users, then data is synced by timestamp on a delayed network. Datacenters are able to solve consistency issue and share data to other users.

Class datacenter stores both metadata of server connection of server/clients, and key value data with version information. After receiving client's write request, datacenter will send replication request, *dependency_check* function will check if dependency condition is satisfied. If is not, remote datacenter will wait until data prorogates.

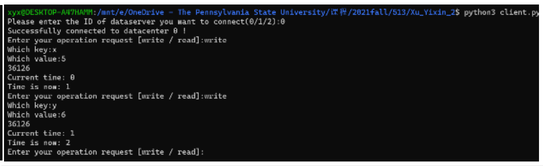
Client is responsible for making reading and write request. Lamport clock is managed by both server and client. Clock is synchronized in every connection. When sending a message, time will increment. For receiving a clock message, local clock will compare with received time. If remote time is greater, local clock be updated to remote lock.

Usage and Output

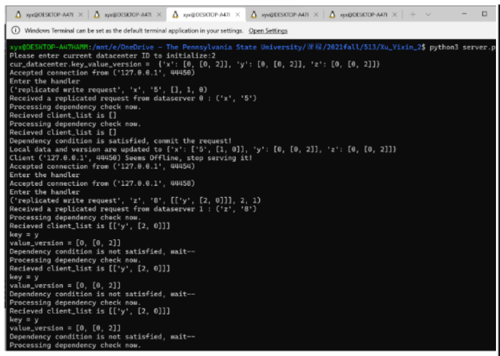
To run the program, we can launch the program by *python3 client.py* and *python3 server.py*. The sample output is generated below.



```
xyx@DESKTOP-A47HAMM:/mnt/e/OneDrive - The Pennsylvania State University/课程/2021fall/513/Xu_Yixin_2$ python3 client.py
Please enter the ID of dataserver you want to connect(0/1/2):1
Successfully connected to datacenter 1 !
Enter your operation request [write / read]:read
31073
Which key do you want to read?y
Current time: 0
Time is now: 1
server time is: 1
Read out y = 6
Enter your operation request [write / read]:write
Which key:z
Which value:8
31073
Current time: 1
Time is now: 2
Enter your operation request [write / read]:
```

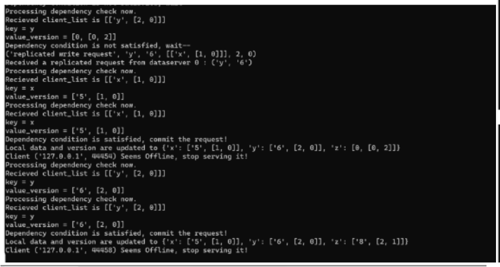


```
xyx@DESKTOP-A47HAMM:/mnt/e/OneDrive - The Pennsylvania State University/课程/2021fall/513/Xu_Yixin_2$ python3 client.py
Please enter the ID of dataserver you want to connect(0/1/2):0
Successfully connected to datacenter 0 !
Enter your operation request [write / read]:write
Which key:x
Which value:5
36126
Current time: 0
Time is now: 1
Enter your operation request [write / read]:write
Which key:y
Which value:6
36126
Current time: 1
Time is now: 2
Enter your operation request [write / read]:
```

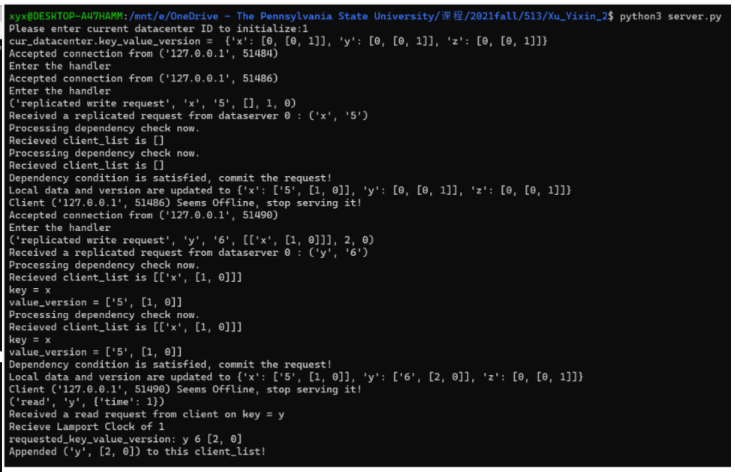


Windows Terminal can be set as the default terminal application in your settings. Open Settings

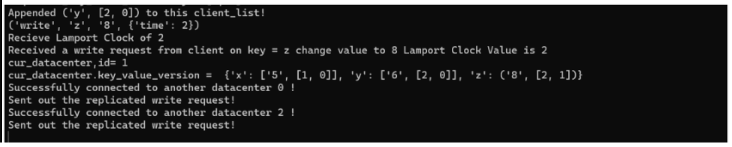
```
xyx@DESKTOP-A47HAMM:/mnt/e/OneDrive - The Pennsylvania State University/课程/2021fall/513/Xu_Yixin_2$ python3 server.p
Please enter current datacenter ID to initialize:2
cur_datacenter.key_value_version =  {'x': [0, [0, 2]], 'y': [0, [0, 2]], 'z': [0, [0, 2]]}
Accepted connection from ('127.0.0.1', 44450)
Enter the handler
('replicated write request', 'x', '5', [], 1, 0)
Received a replicated request from dataserver 0 : ('x', '5')
Processing dependency check now.
Recieved client_list is []
Processing dependency check now.
Recieved client_list is []
Dependency condition is satisfied, commit the request!
Local data and version are updated to {'x': ['5', [1, 0]], 'y': [0, [0, 2]], 'z': [0, [0, 2]]}
Client ('127.0.0.1', 44450) Seems Offline, stop serving it!
Accepted connection from ('127.0.0.1', 44454)
Enter the handler
Accepted connection from ('127.0.0.1', 44458)
Enter the handler
('replicated write request', 'z', '8', [['y', [2, 0]]], 2, 1)
Received a replicated request from dataserver 1 : ('z', '8')
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = [0, [0, 2]]
Dependency condition is not satisfied, wait--
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = [0, [0, 2]]
Dependency condition is not satisfied, wait--
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = [0, [0, 2]]
Dependency condition is not satisfied, wait--
Processing dependency check now.
```



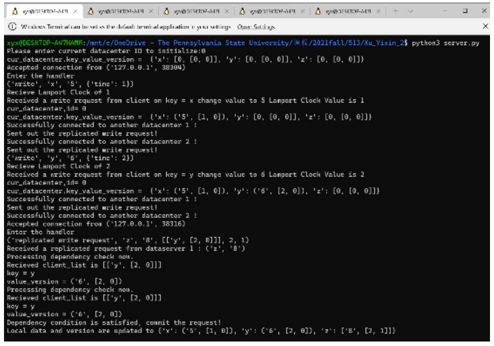
```
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = [0, [0, 2]]
Dependency condition is not satisfied, wait--
('replicated write request', 'y', '6', [['x', [1, 0]]], 2, 0)
Received a replicated request from dataserver 0 : ('y', '6')
Processing dependency check now.
Recieved client_list is [['x', [1, 0]]]
key = x
value_version = ['5', [1, 0]]
Processing dependency check now.
Recieved client_list is [['x', [1, 0]]]
key = x
value_version = ['5', [1, 0]]
Dependency condition is satisfied, commit the request!
Local data and version are updated to {'x': ['5', [1, 0]], 'y': ['6', [2, 0]], 'z': [0, [0, 2]]}
Client ('127.0.0.1', 44454) Seems Offline, stop serving it!
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = ['6', [2, 0]]
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = ['6', [2, 0]]
Dependency condition is satisfied, commit the request!
Local data and version are updated to {'x': ['5', [1, 0]], 'y': ['6', [2, 0]], 'z': ['8', [2, 1]]}
Client ('127.0.0.1', 44458) Seems Offline, stop serving it!
```



```
xyx@DESKTOP-A47HAMM:/mnt/e/OneDrive - The Pennsylvania State University/课程/2021fall/513/Xu_Yixin_2$ python3 server.py
Please enter current datacenter ID to initialize:1
cur_datacenter.key_value_version =  {'x': [0, [0, 1]], 'y': [0, [0, 1]], 'z': [0, [0, 1]]}
Accepted connection from ('127.0.0.1', 51484)
Enter the handler
Accepted connection from ('127.0.0.1', 51486)
Enter the handler
('replicated write request', 'x', '5', [], 1, 0)
Received a replicated request from dataserver 0 : ('x', '5')
Processing dependency check now.
Recieved client_list is []
Processing dependency check now.
Recieved client_list is []
Dependency condition is satisfied, commit the request!
Local data and version are updated to {'x': ['5', [1, 0]], 'y': [0, [0, 1]], 'z': [0, [0, 1]]}
Client ('127.0.0.1', 51486) Seems Offline, stop serving it!
Accepted connection from ('127.0.0.1', 51490)
Enter the handler
('replicated write request', 'y', '6', [['x', [1, 0]]], 2, 0)
Received a replicated request from dataserver 0 : ('y', '6')
Processing dependency check now.
Recieved client_list is [['x', [1, 0]]]
key = x
value_version = ['5', [1, 0]]
Processing dependency check now.
Recieved client_list is [['x', [1, 0]]]
key = x
value_version = ['5', [1, 0]]
Dependency condition is satisfied, commit the request!
Local data and version are updated to {'x': ['5', [1, 0]], 'y': ['6', [2, 0]], 'z': [0, [0, 1]]}
Client ('127.0.0.1', 51490) Seems Offline, stop serving it!
('read', 'y', {'time': 1})
Received a read request from client on key = y
Recieve Lamport Clock of 1
requested_key_value_version: y 6 [2, 0]
Appended ('y', [2, 0]) to this client_list!
```



```
Appended ('y', [2, 0]) to this client_list!
('write', 'z', '8', {'time': 2})
Recieve Lamport Clock of 2
Received a write request from client on key = z change value to 8 Lamport Clock Value is 2
cur_datacenter,id= 1
cur_datacenter.key_value_version =  {'x': ['5', [1, 0]], 'y': ['6', [2, 0]], 'z': ('8', [2, 1])}
Successfully connected to another datacenter 0 !
Sent out the replicated write request!
Successfully connected to another datacenter 2 !
Sent out the replicated write request!
```



Windows Terminal can be set as the default terminal application in your settings. Open Settings

```
xyx@DESKTOP-A47HAMM:/mnt/e/OneDrive - The Pennsylvania State University/课程/2021fall/513/Xu_Yixin_2$ python3 server.py
Please enter current datacenter ID to initialize:0
cur_datacenter.key_value_version =  {'x': [0, [0, 0]], 'y': [0, [0, 0]], 'z': [0, [0, 0]]}
Accepted connection from ('127.0.0.1', 38304)
Enter the handler
('write', 'x', '5', {'time': 1})
Recieve Lamport Clock of 1
Received a write request from client on key = x change value to 5 Lamport Clock Value is 1
cur_datacenter,id= 0
cur_datacenter.key_value_version =  {'x': ('5', [1, 0]), 'y': [0, [0, 0]], 'z': [0, [0, 0]]}
Successfully connected to another datacenter 1 !
Sent out the replicated write request!
Successfully connected to another datacenter 2 !
Sent out the replicated write request!
('write', 'y', '6', {'time': 2})
Recieve Lamport Clock of 2
Received a write request from client on key = y change value to 6 Lamport Clock Value is 2
cur_datacenter,id= 0
cur_datacenter.key_value_version =  {'x': ('5', [1, 0]), 'y': ('6', [2, 0]), 'z': [0, [0, 0]]}
Successfully connected to another datacenter 1 !
Sent out the replicated write request!
Successfully connected to another datacenter 2 !
Accepted connection from ('127.0.0.1', 38316)
Enter the handler
('replicated write request', 'z', '8', [['y', [2, 0]]], 2, 1)
Received a replicated request from dataserver 1 : ('z', '8')
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = ('6', [2, 0])
Processing dependency check now.
Recieved client_list is [['y', [2, 0]]]
key = y
value_version = ('6', [2, 0])
Dependency condition is satisfied, commit the request!
Local data and version are updated to {'x': ('5', [1, 0]), 'y': ('6', [2, 0]), 'z': ['8', [2, 1]]}
```