01_HW_07

March 30, 2022

1 Distributed Systems (LTAT.06.007)

1.0.1 Seminar 7: Process/Node replication in python

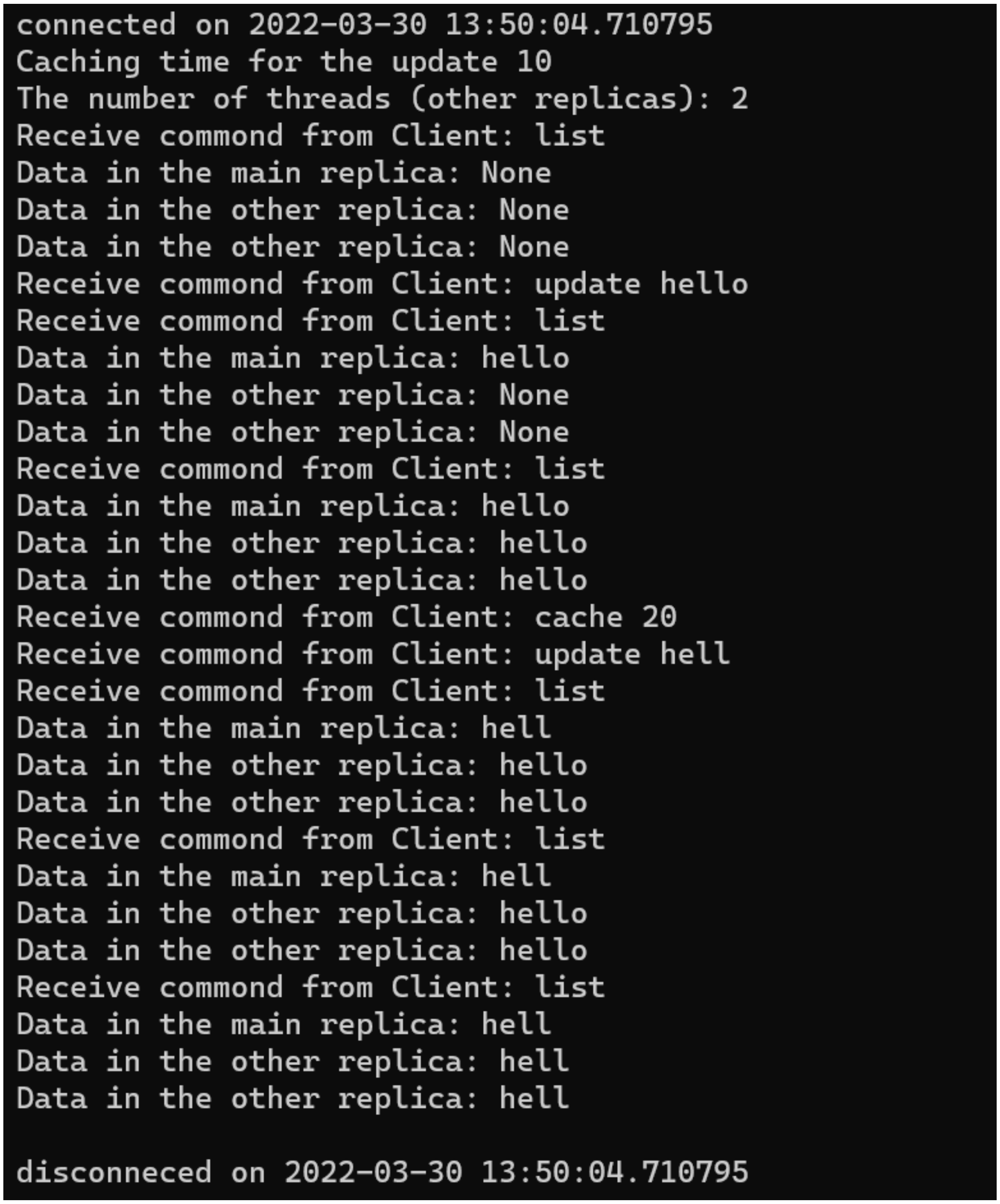
Student : ChengHan Chung

1.1 Task

From previous seminars, we have already implemented some functions with multi-threads by using RPyC ThreadedServer in python. For this task, we will implement the **Passive Replication**:

- All the messaging passing is just between the client and the main thread(main replica) in the server.
  - Like seminar 6, when the client sends one command/message, the server execute the related function.
- Create **N (N>=1)** threads in the server side ($ **program N**).
  - These **N** threads are like other **N** replicas which just copy the data of the main replica.
  - Suggestion: you can create the thread using class in Python (thread class).
- Show the data of all replicas ($ **list**)
- update the data of the main replica ($ **update data**)
  - The data can be an integer, a string or others.
- set the update(cache) time t ($ **cache t**).
  - When the main replica update its data, other replicas update its own data after t seconds (e.g., t=20).
  - Please note that the update process is always running automatically, this command just changes the update time. If the client does not send this command, the default update time is 10 seconds.

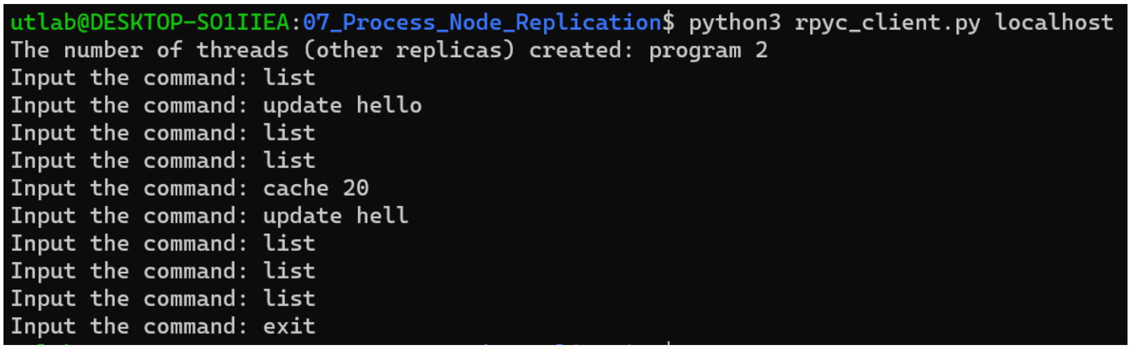
(A) : 1. The result of server side:



```
connected on 2022-03-30 13:50:04.710795
Caching time for the update 10
The number of threads (other replicas): 2
Receive commond from Client: list
Data in the main replica: None
Data in the other replica: None
Data in the other replica: None
Receive commond from Client: update hello
Receive commond from Client: list
Data in the main replica: hello
Data in the other replica: None
Data in the other replica: None
Receive commond from Client: list
Data in the main replica: hello
Data in the other replica: hello
Data in the other replica: hello
Receive commond from Client: cache 20
Receive commond from Client: update hell
Receive commond from Client: list
Data in the main replica: hell
Data in the other replica: hello
Data in the other replica: hello
Receive commond from Client: list
Data in the main replica: hell
Data in the other replica: hello
Data in the other replica: hello
Receive commond from Client: list
Data in the main replica: hell
Data in the other replica: hell
Data in the other replica: hell

disconneced on 2022-03-30 13:50:04.710795
```

2. The result of client side:



```
utlab@DESKTOP-SO1IIEA:07_Process_Node_Replication$ python3 rpyc_client.py localhost
The number of threads (other replicas) created: program 2
Input the command: list
Input the command: update hello
Input the command: list
Input the command: list
Input the command: cache 20
Input the command: update hell
Input the command: list
Input the command: list
Input the command: list
Input the command: exit
```