Shanghai Jiaotong University

Big Data Processing Technology

Project 2: Distributed Lock Design

Author:
FEI Yixiao
118260910031
LI Yanhao
118260910036
LUO Dihao
118260910039
SHEN Shengyang
118260910042
YAN Shuhan
118260910050

Supervisor:
Chentao WU
Xin XIE

October 23, 2019

1 Introduction

Our Distributed Lock Design is based on python and is composed by following files:

- Leader Server
  - leader_server.py: The leader server is the first part that running in the system. The leader server will bind a socket and keep listenning when the system starts. The leader server will create a new thread to process the communication when a new follower server or client bind the same socket with the leader server.And the thread processing the messages between leader server and follower servers or clients will assign a unique ID to them if it is their first time to connect with the leader server.

    To check the owner of a distributed lock, the follower server accesses its map directly and sends the results to the clients.

    When the leader server handling preempt/release requests: It will 1. modify its local map 2. check the request is pending or not 3. send an answer to the client
- Follower Server
  - follower_server.py: A follower server will bind two socket ports. One is used to connect with the leader server and another is used to keep listenning for clients. If there is a client connected with the follower, the follower server will assign a unique ID to the client and create a new thread to process the requests from this client.

    To check the owner of a distributed lock, the follower server accesses its map directly and sends the results to the clients.

    To check the owner of a distributed lock, the follower server accesses its map directly and sends the results to the clients.
- Client
  - client.py : A client can bind a socket port which is same as the server and send request to the server. Here, the client can send three kinds of request, check lock, release lock and preempt lock.

1.1 Usage

- leader_server.py : run this file directly to start the leader server.
- follower_server_a.py: run this file to start one follower server.

- follower_server_b.py: run this file to start another follower server.
- client_a : run this file to start a client.
- client_b : run this file to start the second client.
- client_c : run this file to start the third client.

  Notice: you must start all the servers before start the clients.

2 Example

First, we start the leader server. The port is 9001.

```
(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 leader_server.py
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=
('127.0.0.1', 9001)>
```

Then, we start the follower server a and b. We set the port of follower server a to be 9002, b to be 9003. And they will get their ids: 2 and 3.

```
(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/follower_server_a.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 62
652), raddr=('127.0.0.1', 9001)> 2
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 90
02)> listening
```

```
(base) MacBook-Pro-2:~ apple$ python Desktop/DistributedLockDesign-master/follow
er_server_b.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 60879), raddr=('127.0.0.1', 9001)> 3
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 9003)> listening
```

And the leader server will connect to follower server a and b.

```
(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 leader_server.py
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=
('127.0.0.1', 9001)>
Connect from ('127.0.0.1', 62652)
active threads:
2
Connect from ('127.0.0.1', 62655)
active threads:
3
```

Next, we start client a, and make it directly connect to leader server which means port 9001. It will get its ID:10001.

```
(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien
t_a.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 54024), raddr=('127.0.0.1', 9001)>
client_id: 10001
distributeLock:>
```

Then client a tried to preempt lock01.

```
(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien
t_a.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 54024), raddr=('127.0.0.1', 9001)>
client_id: 10001
distributeLock:> preempt lock01
2019-10-17 20:07:04  Request: preemptlock
2019-10-17 20:07:04  Response:  PreemptLock Success:10001
distributeLock:>
```

Leader server will check the lock list and give it a response. Also update the new state of lock01 and broadcast to all the follower servers.

```
(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 leader_server.py
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=
('127.0.0.1', 9001)>
Connect from ('127.0.0.1', 62652)
active threads:
2
Connect from ('127.0.0.1', 62655)
active threads:
3
Connect from ('127.0.0.1', 62923)
active threads:
4
broadcast new lock_map
```

```
(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/follower_server_a.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 53
666), raddr=('127.0.0.1', 9001)> 2
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 90
02)> listening
UpdateLockmap
```

```
(base) MacBook-Pro-2:~ apple$ python Desktop/DistributedLockDesign-master/follow
er_server_b.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 62655), raddr=('127.0.0.1', 9001)> 3
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 9003)> listening
UpdateLockmap
```

We start client b which is connected to follower server a. It will get its ID: 20001.

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien]
t_b.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60432), raddr=('127.0.0.1', 9002)>
client_id: 20001
distributeLock:>
```

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/follower_server_a.py ]
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60
423), raddr=('127.0.0.1', 9001)> 2
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 90
02)> listening
UpdateLockmap
connect with  ('127.0.0.1', 60432)
```

The client b check the lock01. The follower server a will tell it who owns lock01.

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien]
t_b.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60432), raddr=('127.0.0.1', 9002)>
client_id: 20001
distributeLock:> check lock01
2019-10-21 10:50:53  Request: checklock
2019-10-21 10:50:53  Response:  CheckLock:Lock:10001
distributeLock:>
```

The client b preempt the lock01. Obviously it will fail.

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien]
t_b.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60432), raddr=('127.0.0.1', 9002)>
client_id: 20001
distributeLock:> check lock01
2019-10-21 10:50:53  Request: checklock
2019-10-21 10:50:53  Response:  CheckLock:Lock:10001
distributeLock:> preempt lock01
2019-10-21 10:54:16  Request: preemptlock
2019-10-21 10:54:16  Response:  PreemptLock Failed
distributeLock:>
```

Then client b preempt the lock02. Leader server will check the lock list and give it a response. Also update the new state of lock02 and broadcast to all the follower servers.

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien]
t_b.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60432), raddr=('127.0.0.1', 9002)>
client_id: 20001
distributeLock:> check lock01
2019-10-21 10:50:53  Request: checklock
2019-10-21 10:50:53  Response:  CheckLock:Lock:10001
distributeLock:> preempt lock01
2019-10-21 10:54:16  Request: preemptlock
2019-10-21 10:54:16  Response:  PreemptLock Failed
distributeLock:> preempt lock02
2019-10-21 10:55:08  Request: preemptlock
2019-10-21 10:55:08  Response:  PreemptLock Success
distributeLock:>
```

```
[(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 leader_server.py ]
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=
('127.0.0.1', 9001)>
Connect from ('127.0.0.1', 62652)
active threads:
2
Connect from ('127.0.0.1', 62655)
active threads:
3
Connect from ('127.0.0.1', 62923)
active threads:
4
broadcast new lock_map
broadcast new lock_map
```

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/follower_server_a.py ]
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60
423), raddr=('127.0.0.1', 9001)> 2
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 90
02)> listening
UpdateLockmap
connect with  ('127.0.0.1', 60432)
UpdateLockmap
```

```
[(base) MacBook-Pro-2:~ apple$ python Desktop/DistributedLockDesign-master/follow]
er_server_b.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 62655), raddr=('127.0.0.1', 9001)> 3
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 9003)> listening
UpdateLockmap
UpdateLockmap
```

Start client c which is connected to follower server b. It will get its ID: 30001

```
[(base) MacBook-Pro-2:~ apple$ python Desktop/DistributedLockDesign-master/follow]
er_server_b.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 62655), raddr=('127.0.0.1', 9001)> 3
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 9003)> listening
UpdateLockmap
UpdateLockmap
connect with  ('127.0.0.1', 63077)
```

```
[(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 client_c.py    ]
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 60959), raddr=('127.0.0.1', 9003)>
client_id: 30001
distributeLock:>
```

Client c preempt lock 02. It will fail.

```
[(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 client_c.py    ]
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 63077), raddr=('127.0.0.1', 9003)>
client_id: 30001
distributeLock:> preempt lock02
2019-10-21 14:09:50  Request: preemptlock
2019-10-21 14:09:50  Response:  PreemptLock Failed
distributeLock:>
```

Then client b release the lock02. Leader server will check the lock list and remove lock02 from the list. Also update the new state of lock02 and broadcast to all the follower servers.

```
[(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien]
t_b.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 62998), raddr=('127.0.0.1', 9002)>
client_id: 20001
distributeLock:> check lock01
2019-10-21 14:02:06  Request: checklock
2019-10-21 14:02:06  Response:  CheckLock:Lock:10001
distributeLock:> preempt lock02
2019-10-21 14:02:17  Request: preemptlock
2019-10-21 14:02:17  Response:  PreemptLock Success
distributeLock:> release lock02
2019-10-21 14:10:08  Request: releaselock
2019-10-21 14:10:08  Response:  ReleaseLock Success
distributeLock:>
```

```
(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 leader_server.py
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=
('127.0.0.1', 9001)>
Connect from ('127.0.0.1', 62652)
active threads:
2
Connect from ('127.0.0.1', 62655)
active threads:
3
Connect from ('127.0.0.1', 62923)
active threads:
4
broadcast new lock_map
broadcast new lock_map
broadcast to remove lock_map
```

```
(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/follower_server_a.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 62
652), raddr=('127.0.0.1', 9001)> 2
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 90
02)> listening
UpdateLockmap
connect with  ('127.0.0.1', 62998)
UpdateLockmap
RemoveLockmap
```

```
(base) MacBook-Pro-2:~ apple$ python Desktop/DistributedLockDesign-master/follow
er_server_b.py
start to connect with leader_server
follower send data
leader socket and follower_id:
<socket.socket fd=3, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 62655), raddr=('127.0.0.1', 9001)> 3
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
proto=0, laddr=('127.0.0.1', 9003)> listening
UpdateLockmap
UpdateLockmap
connect with  ('127.0.0.1', 63077)
RemoveLockmap
```

Now client c and successfully preempt lock02. And client a will know the owner of lock02 if it check lock02.

```
(base) MacBook-Pro-2:DistributedLockDesign-master apple$ python3 client_c.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 63077), raddr=('127.0.0.1', 9003)>
client_id: 30001
distributeLock:> preempt lock02
2019-10-21 14:09:50  Request: preemptlock
2019-10-21 14:09:50  Response:  PreemptLock Failed
distributeLock:> preempt lock02
2019-10-21 14:28:57  Request: preemptlock
2019-10-21 14:28:57  Response:  PreemptLock Success
distributeLock:>
```

```
(base) MacBook-Pro-2:~ apple$ python3 Desktop/DistributedLockDesign-master/clien
t_a.py
start to connect with server
client send data
server socket: <socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKin
d.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 62923), raddr=('127.0.0.1', 9001)>
client_id: 10001
distributeLock:> preempt lock01
2019-10-21 13:57:08  Request: preemptlock
2019-10-21 13:57:08  Response:  PreemptLock Success:10001
distributeLock:> check lock02
2019-10-21 14:29:26  Request: checklock
2019-10-21 14:29:26  Response:  CheckLock:Lock:30001
distributeLock:>
```