Key-Value Store using Paxos

I. Assignment Overview

A demo can be seen in
https://drive.google.com/file/d/1c3ZLru8rLYmWeAZ2iM7wA6tmmo8MALbw/view?usp=sharing

The current project implements the basic functionalities of Paxos (RMI) for key-value storing. The focus is on implementing sample clients to concurrently communicate with the server and perform three basic operations: 1) PUT (key, value), 2) GET (key), and 3) DELETE (key). Five different instances of server are replicated to increase Server bandwidth and ensure availability. Client is able to access any of the five key-value servers instead of a single server, and get consistent data back from any of the replicas.

The current project consists of four test runs of client-server communication (through Server[1-5].java (under the test folder) and Client.java (under the test folder)). For each test run, a ConcurrentHashmap is created in each of the replicas in order to store the requests of the three operations and ensure thread-safety in a high concurrency environment.

A protocol is designed to dictate whether a request by a client is valid (see details in the Technical Impression). The server is multi-threaded such that it responds to requests from multiple clients one at a time. Under the client-server program, the server runs forever whereas the client will exit and disconnect with the server until it loops through all the requests as specified in the sample text files.

The client side of the current project considers the following main features:

(1) The client is able to contact any of the five key-value replica servers instead of one single server by specifying the desired port numbers;
(2) The protocol designed to communicate packet contents ensures that the server is robust to malformed datagram packets, as mainly reflected in the get(), put() and delete() methods;
(3) Time-stamped client log is formated in both success and error messages.

The server side of the current project considers the following features:

(1) The server and its replicas listen on ports (which have been specified in the serverConfiguration folder);
(2) The server runs forever unless external signal is enforced;

(3) The server display client's request and sends time-stamped response to valid requests back to the client in a human readable fashion;
(4) The server is robust to malformed datagram packets and reports the error in the server log in a human-readable format.
(5) Implementing Paxos protocol ensures that client is able to issue PUT, DELETE and get consistent data back from any of the replicas (even when server(s) crashes).
(6) Each node keeps track of a "log" of current changes and the commits in these logs are all consistent with each other, and ensures fault tolerance.

II. Technical Impression

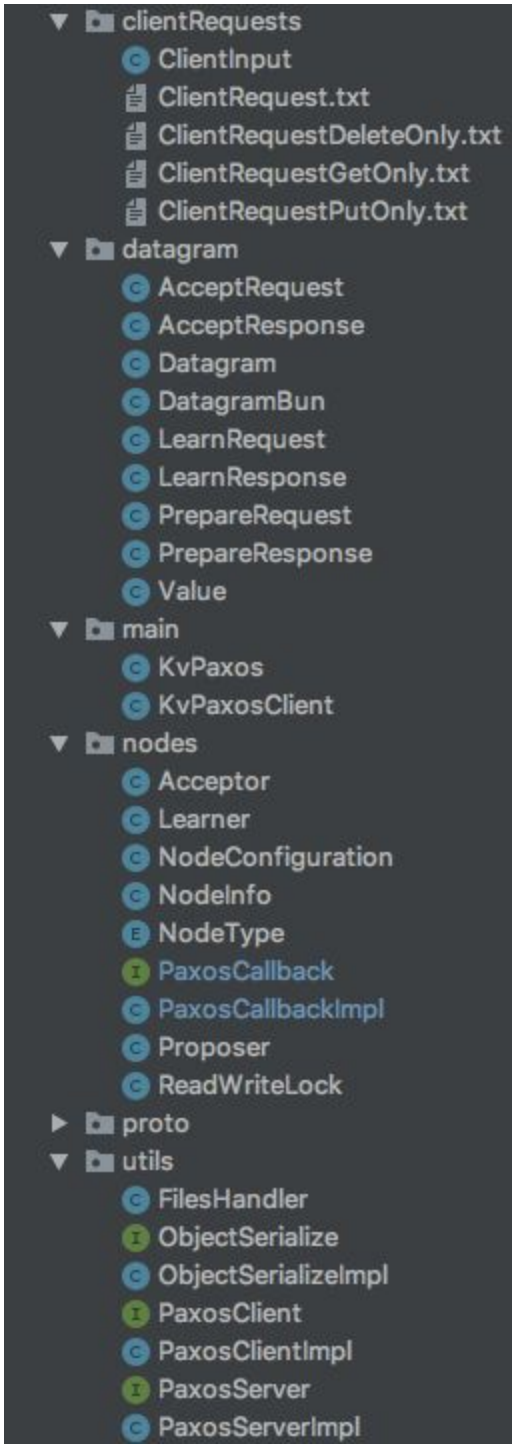
1. Language and Tools

I used Java as the programming language and used Maven to build and manage the project.. Socket is used for communication between client and designated server.
Serialization of message objects is implemented:

(1) to save and persist the state of an object (e.g., client request).
(2) to travel an object across a network.
(3) to allow the key-value store service to take advantage of efficient serialization and easy interface updating.

2. Class Overview

In order to implement all the features in the sending and receiving end-points as discussed in the previous sections, two sections were introduced: main folder, and test folder.
The main folder is divided into 5 subfolders (see screenshot below):



(1) clientRequests: takes in client requests (see txt files for the sample requests);

(2) datagram: data packets that are sent between nodes. For example, AcceptRequest.java represents ACCEPT request sent from Proposer to Accept. It contains information about the server id, instance number, ballot number, and the value of the ballot.

(3) main: KvPaxosServer is responsible for launching the paxos server whereas KvPaxosClient initiates a client object that sets a target proposer (by IP address and port) to connect with, and then submit input to buffer queue so that server could fetch data from.

(4) nodes:

Proposer.java is initially active in Phase 1a of the paxos algorithm as described in Wikipedia. It creates a message called "Prepare", with a unique identification number (ballot) for the message (which is greater than any previous “Prepare” message, and no value is attached at this point yet). The message is analogous to “Prepare to commit, please!”, and is then sent to a Quorum of Acceptors.
A key feature of this method is to set time out for the acceptor to respond. If acceptor is not responding to the prepare request from the proposer in a certain time window, proposer retries phase 1a (send prepare request to acceptors).

It is then active in Phase 2a (often called Accept phase). If a Proposer receives a majority of ACKs from a Quorum of Acceptors, it needs to set a value v to its proposal. If any Acceptors had previously accepted any proposal, then they'll have sent their values to the Proposer.

Acceptor.java is active in Phase 1b of the Paxos algorithm (often called Promise). It follows the steps as described in Wikipedia:
Any of the Acceptors waits for a Prepare message from any of the Proposers. If an Acceptor receives a Prepare message, the Acceptor must look at the ballot number n of the just received Prepare message.
There are two cases. If n is higher than every previous proposal number received from any of the Proposers, by the Acceptor, the Acceptor will return a message, which we call an ACK, to the Proposer to ignore all future proposals having a number less than n.
If the Acceptor accepted a proposal at some point in the past, it must include the previous proposal number and the corresponding accepted value in its response to the Proposer.

(5) utils takes care of files for writing data to disk, serializing/deserializing message objects, writing client input to message queues, and reading messages from queues.

The test folder is to set up five servers, and one client.

3. Data Design and Data Structures

To ensure read-write consistency, I implemented a ReadWriteLock. On top of this, as described in the Setup section, I also used Java ConcurrentHashmap for key-value store database that is used for storing, deleting, and checking key and value. The reaon of using ConcurrentHashMap is that I need very high concurrency in the current project. It has four advantage over hashmap:

(1) It is thread-safe without synchronizing the whole map.
(2) Reads (e.g., get) can happen very fast while write (e.g., delete, put) is done with a lock.
(3) There is no locking at the object level.
(4) The locking is at a much finer granularity at a hashmap bucket level.

4. Protocol for PUT, GET, and DELETE operations

Every client must follow the syntax of <operation> <key> for GET and DELETE, and <operation> <key> <value> for PUT. If invalid request is sent, the server will send time-stamped error message such as “Malformed Request from [IP: " + clientAddress + ", Port: "+ clientPort + "]." + " Syntax: <operation> <key> OR <operation> <key> <value>. For example: get apple".

The protocol of the three operations are as follows:

(1) PUT inserts a key-value pair into the database, and if key exists, no change is made, and error message of “key already exists” is sent to client. With ConcurrentHashmap, only one client can do the put operation once at a time.
(2) GET retrieves the value by key in the database. If no put or delete is in action, multiple clients can get the value concurrently without blocking each other.
(3) DELETE removes the key-value pair by key in the database. If no key exists, error message such as “Key does not exist” is sent over to the client. Only one client can do DELETE once at a time, with all other threads which intend to do DELETE being blocked until lock is released.

5. How to run the project

There are two ways to run the project:

(1) Run the shell scripts:

There are 5 server bash, and 1 client bash.

Run the 5 server bash in 5 separate terminals and then run the clients. To test how the program works for different scenarios below:

a. Only 4 servers alive
b. only 3 servers alive.
c. only 2 servers alive.
d. The server that you killed in the previous steps will be revived

You can simply close the terminal as related to each server. The default server for client to request is the server running on port 30001 (server 1). There is no mechanism to route the client to other server if server 1 is down. So do not bring down server 1.

(2) The second way is to type in command lines:

The current project runs on 5 servers and one client classes. To start the five servers, in ***each*** of your CLI (5 terminals in total for the 5 servers), type in the following:

java -jar [path directory of the **server** jar] [path of server configuration file]

Then open up the 6th CLI window, and type in the following:

java -jar [path directory of the client jar] [port numbers separated by white space] [sampe run txt file path]

For example, if the sample run txt file is ClientRequest.txt, specify it in your last argument, such as:
Java -jar 3001 /Users/Sam/Desktop/project3/out/artifacts/Client/ClientRequest.txt

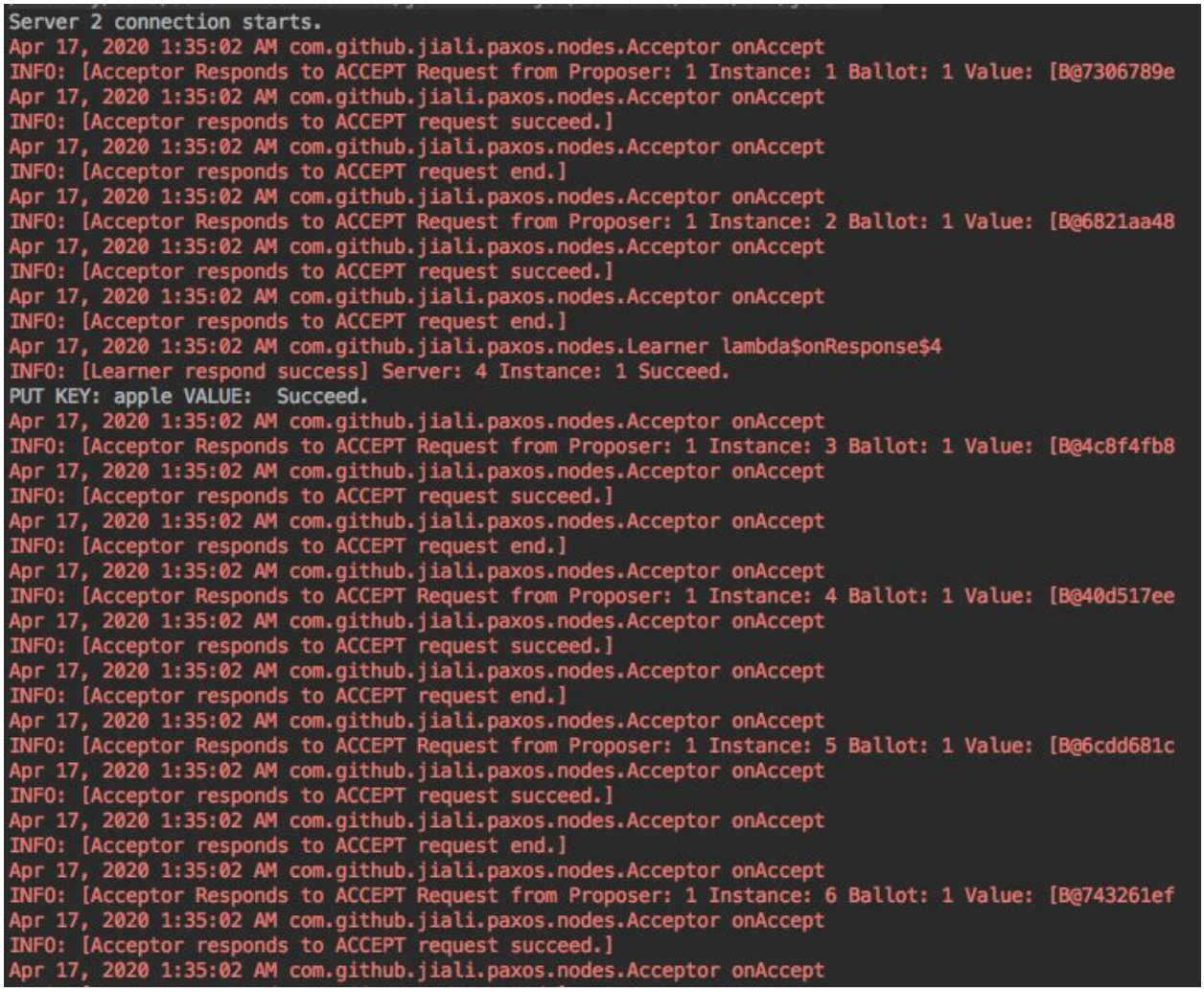
To switch to other sample runs (e.g., ClientRequestGetOnly.txt, ClientRequestPutOnly, ClientRequestDeleteOnly), specify as instructed.
The ClientRequest.txt sample run consists of a mix of at least five of each operation: 5 PUTs, 5 GETs, 5 DELETEs.

The following screenshots should appear in all of the following scenarios:

(1) Only 4 servers alive
(2) only 3 servers alive.
(3) only 2 servers alive.
(4) The server that you killed in the previous steps will be revived

Please note that the red notes are from the logger, while the white notes are going to be feedback for the client.

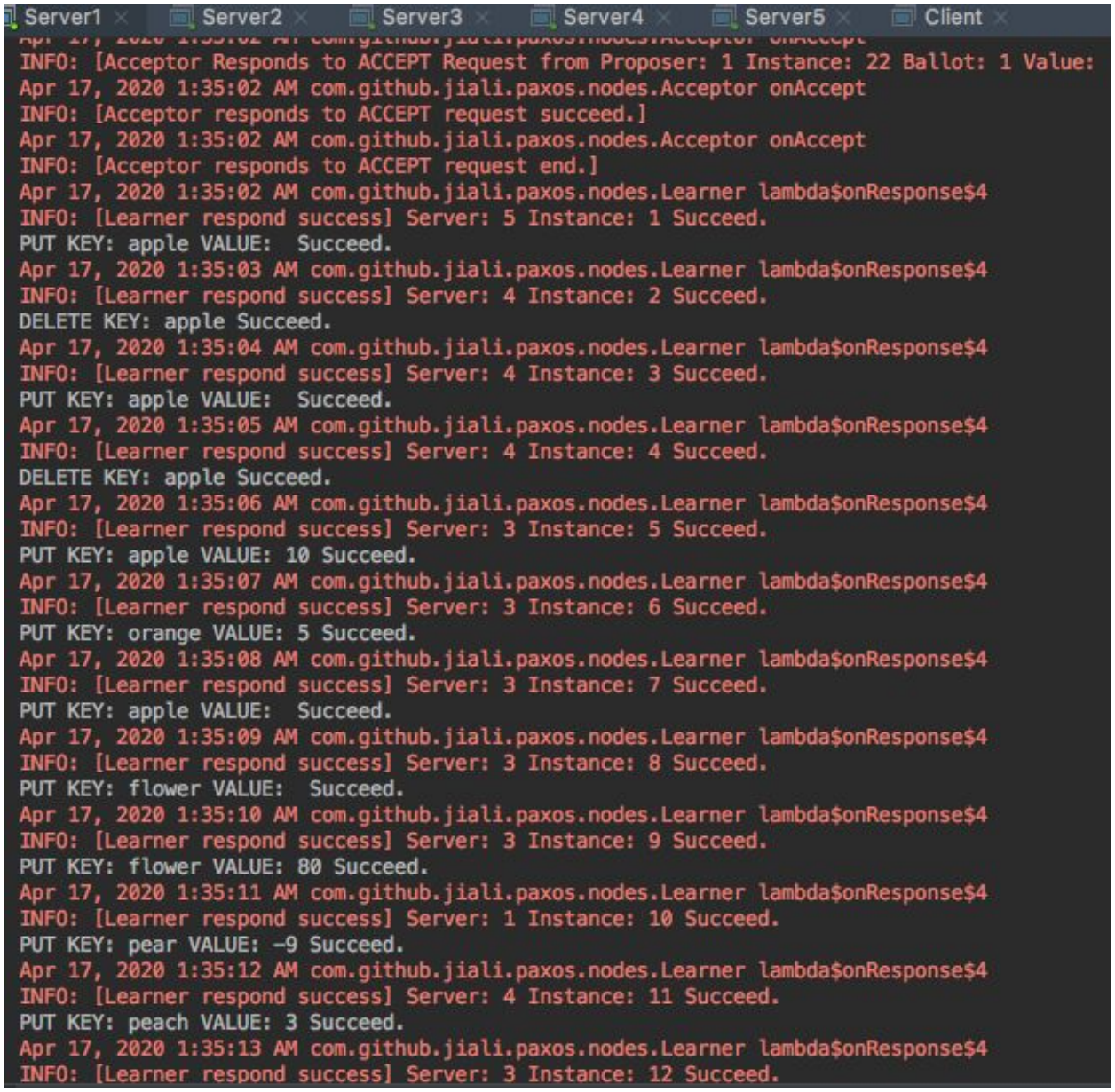
```
Server 1 connection starts.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 1 Ballot: 1 Value: [B@3407c1e6
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 2 Ballot: 1 Value: [B@5280edec
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 3 Ballot: 1 Value: [B@5a1590e
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 4 Ballot: 1 Value: [B@620f1356
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 5 Ballot: 1 Value: [B@5a332a03
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 6 Ballot: 1 Value: [B@5bc73d2
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 7 Ballot: 1 Value: [B@4d40c245
```

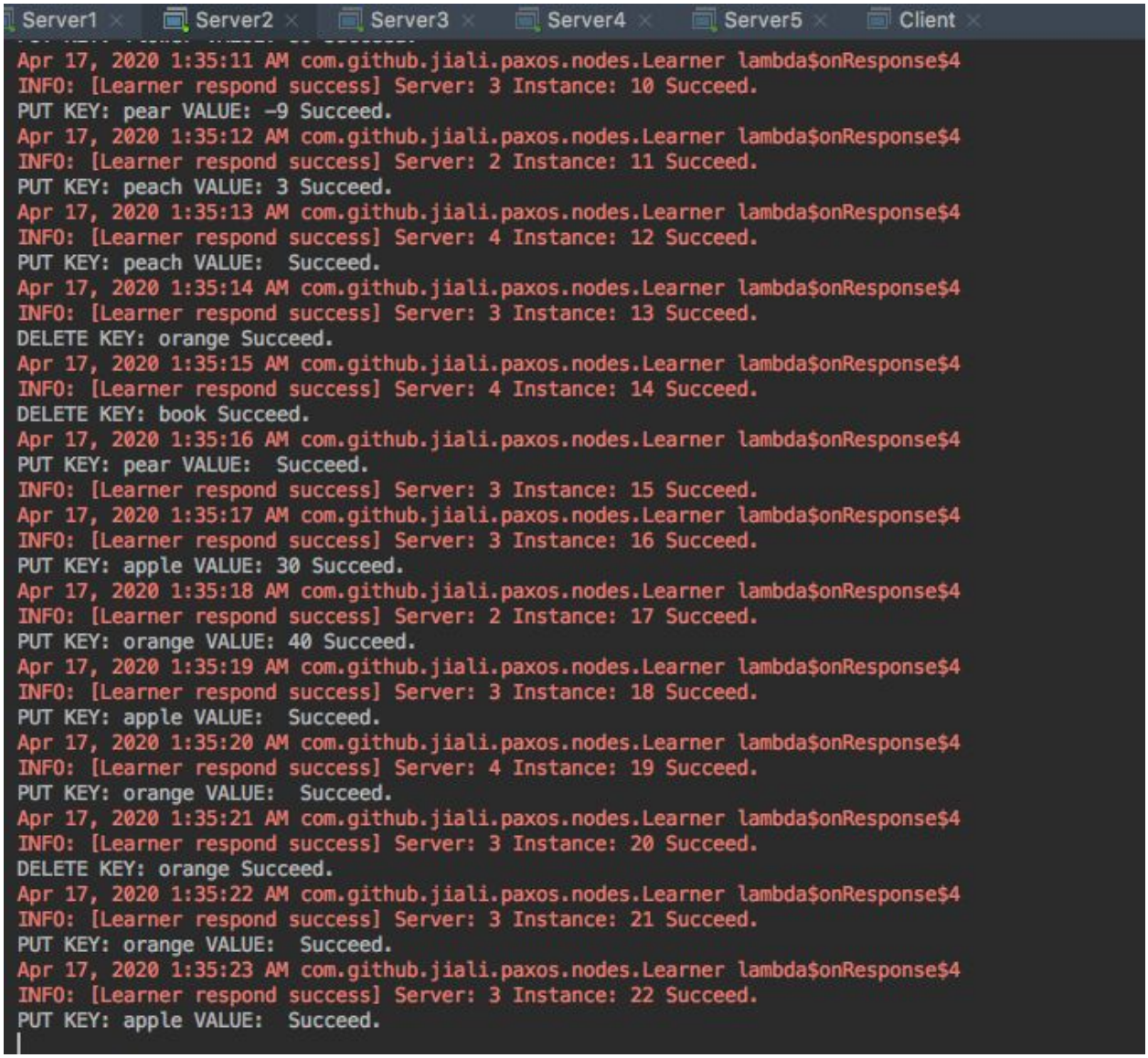

Server 2 connection starts.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 1 Ballot: 1 Value: [B@7306789e
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 2 Ballot: 1 Value: [B@6821aa48
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 1 Succeed.
PUT KEY: apple VALUE: Succeed.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 3 Ballot: 1 Value: [B@4c8f4fb8
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 4 Ballot: 1 Value: [B@40d517ee
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 5 Ballot: 1 Value: [B@6cdd681c
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 6 Ballot: 1 Value: [B@743261ef
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept

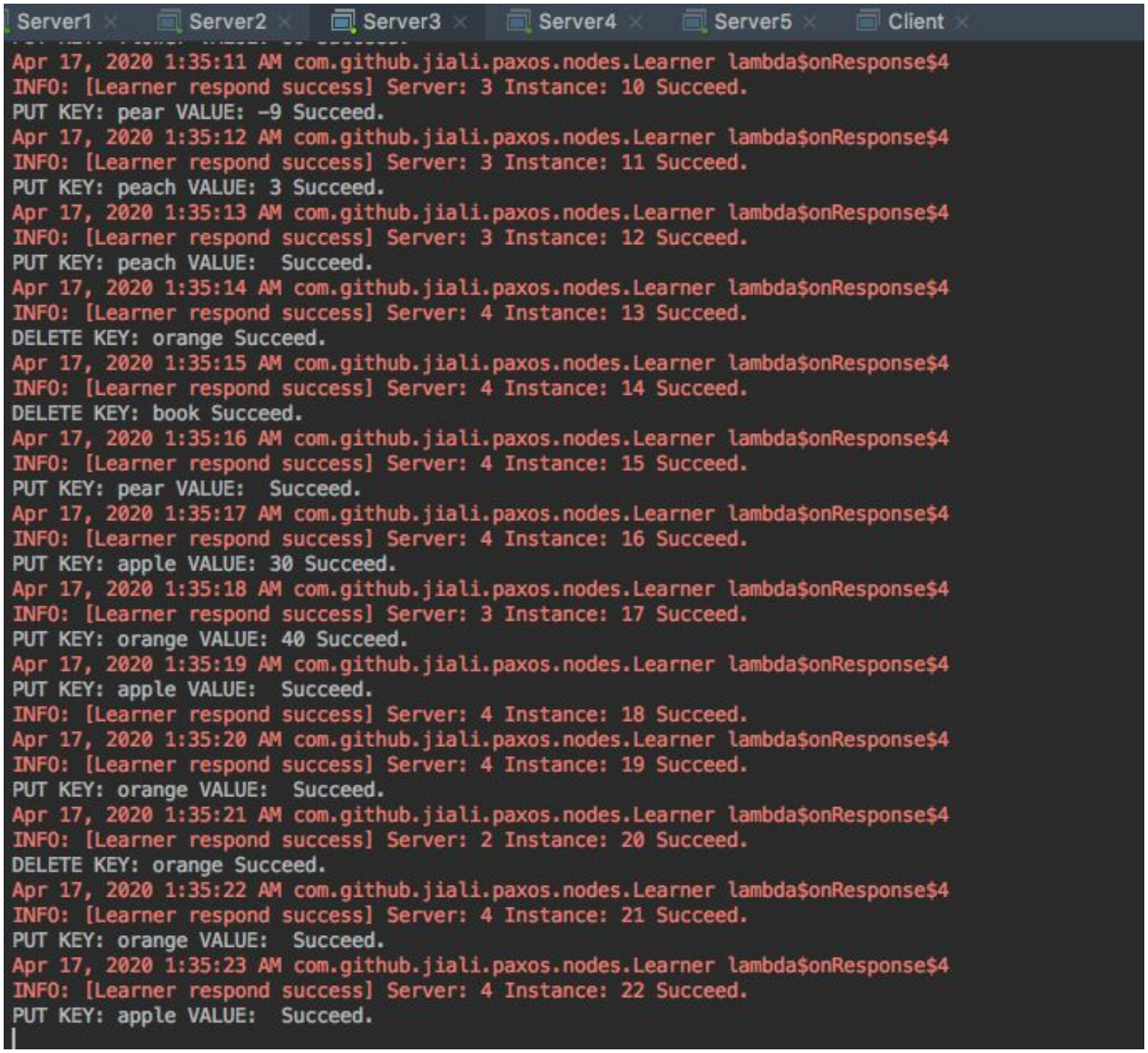
```
Server 3 connection starts.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 1 Ballot: 1 Value: [B@7306789e
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 2 Ballot: 1 Value: [B@6821aa48
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 3 Ballot: 1 Value: [B@4c8f4fb8
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 4 Ballot: 1 Value: [B@40d517ee
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 5 Ballot: 1 Value: [B@6cdd681c
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 6 Ballot: 1 Value: [B@743261ef
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 7 Ballot: 1 Value: [B@90876e5
```

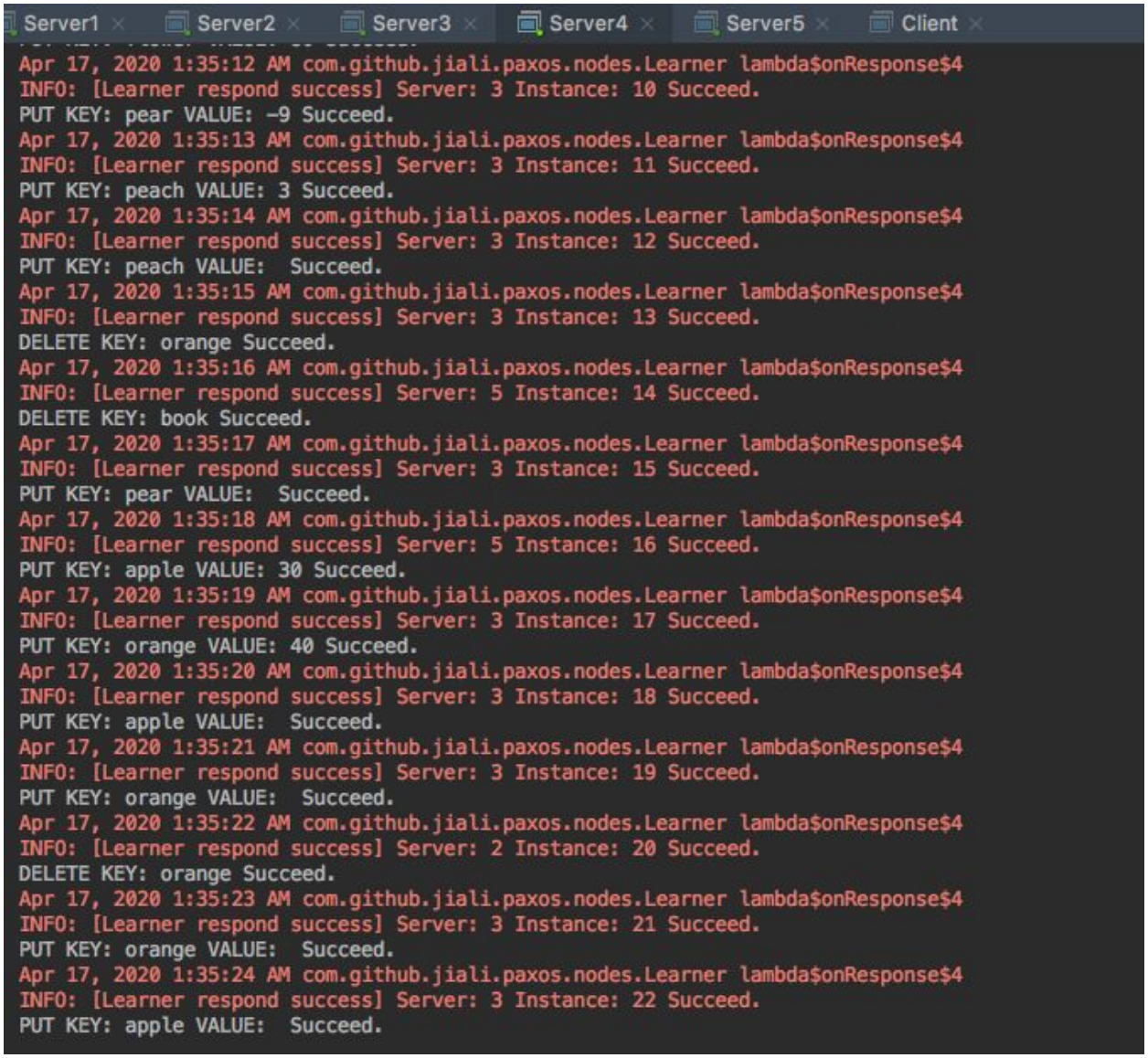
```
Server 4 connection starts.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 1 Ballot: 1 Value: [B@3407c1e6
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 2 Ballot: 1 Value: [B@509b217c
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 3 Ballot: 1 Value: [B@6d695edc
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 4 Ballot: 1 Value: [B@40bc04e1
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 5 Ballot: 1 Value: [B@469c2a30
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 6 Ballot: 1 Value: [B@2b840604
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 7 Ballot: 1 Value: [B@3b92126a
```

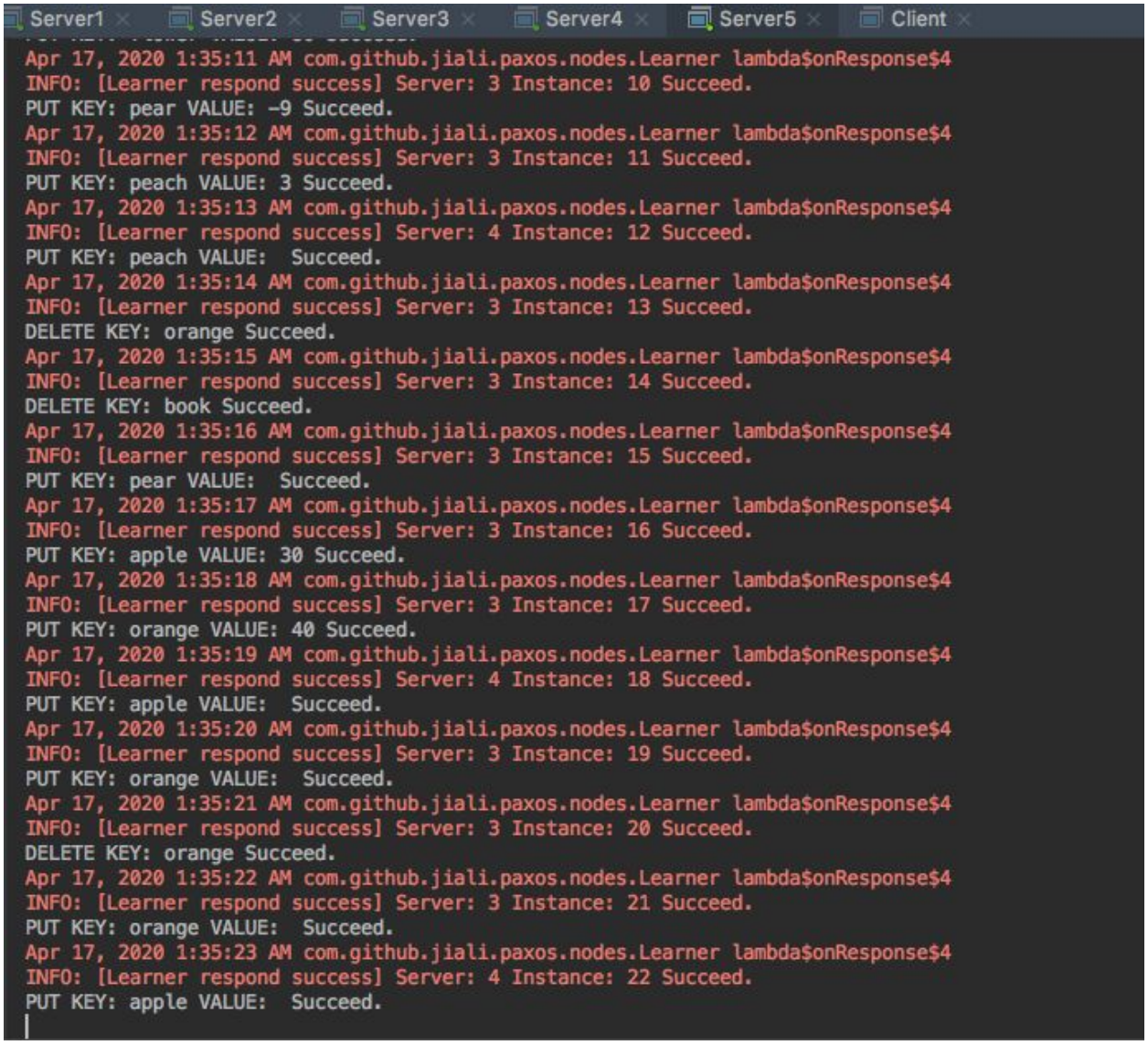
```
Server 5 connection starts.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 1 Ballot: 1 Value: [B@758a4fe9
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 1 Succeed.
PUT KEY: apple VALUE:  Succeed.
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 2 Ballot: 1 Value: [B@12939c3f
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 3 Ballot: 1 Value: [B@62d23592
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 4 Ballot: 1 Value: [B@50fbd6c3
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 5 Ballot: 1 Value: [B@5f41b5c
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 6 Ballot: 1 Value: [B@2cad4b8d
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
```


Server1
Server2
Server3
Server4
Server5
Client
INFO: [Acceptor Responds to ACCEPT Request from Proposer: 1 Instance: 22 Ballot: 1 Value:
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request succeed.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Acceptor onAccept
INFO: [Acceptor responds to ACCEPT request end.]
Apr 17, 2020 1:35:02 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 5 Instance: 1 Succeed.
PUT KEY: apple VALUE: Succeed.
Apr 17, 2020 1:35:03 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 2 Succeed.
DELETE KEY: apple Succeed.
Apr 17, 2020 1:35:04 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 3 Succeed.
PUT KEY: apple VALUE: Succeed.
Apr 17, 2020 1:35:05 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 4 Succeed.
DELETE KEY: apple Succeed.
Apr 17, 2020 1:35:06 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 5 Succeed.
PUT KEY: apple VALUE: 10 Succeed.
Apr 17, 2020 1:35:07 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 6 Succeed.
PUT KEY: orange VALUE: 5 Succeed.
Apr 17, 2020 1:35:08 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 7 Succeed.
PUT KEY: apple VALUE: Succeed.
Apr 17, 2020 1:35:09 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 8 Succeed.
PUT KEY: flower VALUE: Succeed.
Apr 17, 2020 1:35:10 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 9 Succeed.
PUT KEY: flower VALUE: 80 Succeed.
Apr 17, 2020 1:35:11 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 1 Instance: 10 Succeed.
PUT KEY: pear VALUE: -9 Succeed.
Apr 17, 2020 1:35:12 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 11 Succeed.
PUT KEY: peach VALUE: 3 Succeed.
Apr 17, 2020 1:35:13 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 12 Succeed.


Server1
Server2
Server3
Server4
Server5
Client
Apr 17, 2020 1:35:11 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 10 Succeed.
PUT KEY: pear VALUE: -9 Succeed.
Apr 17, 2020 1:35:12 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 2 Instance: 11 Succeed.
PUT KEY: peach VALUE: 3 Succeed.
Apr 17, 2020 1:35:13 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 12 Succeed.
PUT KEY: peach VALUE: Succeed.
Apr 17, 2020 1:35:14 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 13 Succeed.
DELETE KEY: orange Succeed.
Apr 17, 2020 1:35:15 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 14 Succeed.
DELETE KEY: book Succeed.
Apr 17, 2020 1:35:16 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
PUT KEY: pear VALUE: Succeed.
INFO: [Learner respond success] Server: 3 Instance: 15 Succeed.
Apr 17, 2020 1:35:17 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 16 Succeed.
PUT KEY: apple VALUE: 30 Succeed.
Apr 17, 2020 1:35:18 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 2 Instance: 17 Succeed.
PUT KEY: orange VALUE: 40 Succeed.
Apr 17, 2020 1:35:19 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 18 Succeed.
PUT KEY: apple VALUE: Succeed.
Apr 17, 2020 1:35:20 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 19 Succeed.
PUT KEY: orange VALUE: Succeed.
Apr 17, 2020 1:35:21 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 20 Succeed.
DELETE KEY: orange Succeed.
Apr 17, 2020 1:35:22 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 21 Succeed.
PUT KEY: orange VALUE: Succeed.
Apr 17, 2020 1:35:23 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 22 Succeed.
PUT KEY: apple VALUE: Succeed.


Server1
Server2
Server3
Server4
Server5
Client
Apr 17, 2020 1:35:11 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 10 Succeed.
PUT KEY: pear VALUE: -9 Succeed.
Apr 17, 2020 1:35:12 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 11 Succeed.
PUT KEY: peach VALUE: 3 Succeed.
Apr 17, 2020 1:35:13 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 12 Succeed.
PUT KEY: peach VALUE: Succeed.
Apr 17, 2020 1:35:14 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 13 Succeed.
DELETE KEY: orange Succeed.
Apr 17, 2020 1:35:15 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 14 Succeed.
DELETE KEY: book Succeed.
Apr 17, 2020 1:35:16 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 15 Succeed.
PUT KEY: pear VALUE: Succeed.
Apr 17, 2020 1:35:17 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 16 Succeed.
PUT KEY: apple VALUE: 30 Succeed.
Apr 17, 2020 1:35:18 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 17 Succeed.
PUT KEY: orange VALUE: 40 Succeed.
Apr 17, 2020 1:35:19 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
PUT KEY: apple VALUE: Succeed.
INFO: [Learner respond success] Server: 4 Instance: 18 Succeed.
Apr 17, 2020 1:35:20 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 19 Succeed.
PUT KEY: orange VALUE: Succeed.
Apr 17, 2020 1:35:21 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 2 Instance: 20 Succeed.
DELETE KEY: orange Succeed.
Apr 17, 2020 1:35:22 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 21 Succeed.
PUT KEY: orange VALUE: Succeed.
Apr 17, 2020 1:35:23 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 4 Instance: 22 Succeed.
PUT KEY: apple VALUE: Succeed.


Server1
Server2
Server3
Server4
Server5
Client
Apr 17, 2020 1:35:12 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 10 Succeed.
PUT KEY: pear VALUE: -9 Succeed.
Apr 17, 2020 1:35:13 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 11 Succeed.
PUT KEY: peach VALUE: 3 Succeed.
Apr 17, 2020 1:35:14 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 12 Succeed.
PUT KEY: peach VALUE: Succeed.
Apr 17, 2020 1:35:15 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 13 Succeed.
DELETE KEY: orange Succeed.
Apr 17, 2020 1:35:16 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 5 Instance: 14 Succeed.
DELETE KEY: book Succeed.
Apr 17, 2020 1:35:17 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 15 Succeed.
PUT KEY: pear VALUE: Succeed.
Apr 17, 2020 1:35:18 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 5 Instance: 16 Succeed.
PUT KEY: apple VALUE: 30 Succeed.
Apr 17, 2020 1:35:19 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 17 Succeed.
PUT KEY: orange VALUE: 40 Succeed.
Apr 17, 2020 1:35:20 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 18 Succeed.
PUT KEY: apple VALUE: Succeed.
Apr 17, 2020 1:35:21 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 19 Succeed.
PUT KEY: orange VALUE: Succeed.
Apr 17, 2020 1:35:22 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 2 Instance: 20 Succeed.
DELETE KEY: orange Succeed.
Apr 17, 2020 1:35:23 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 21 Succeed.
PUT KEY: orange VALUE: Succeed.
Apr 17, 2020 1:35:24 AM com.github.jiali.paxos.nodes.Learner lambda$onResponse$4
INFO: [Learner respond success] Server: 3 Instance: 22 Succeed.
PUT KEY: apple VALUE: Succeed.



6. Area of Improvement

A number of improvements could be made to make the project cleaner and closer to real-life scenario applications. First, using gRPC instead of socket is encouraged. Second, the current situation only considers key of any size and no restrictions are imposed on the type of keys. Future work should consider other user inputs and provide validation of possible values. Third, the current project lets user to specify ports to latch on in order to spawn multiple threads.I also required client to sleep for 50 to 100 milliseconds for each request to control possible incoming data flows. Future work should focus on how thread pools (e.g., schedule thread pool and fixed size thread pool) should perform. Fourth, two

mechanisms are used to allow single-write and multiple reads: (1) ReadWriteLock class to impose locks on reads and writes respectively, and (2) ConcurrentHashmap was created for storing key-value pairs for single-write and multiple reads. Potential improvements in accessibility such as caching of the most recently used key-value pairs could enhance user experience and improve I/O speeds. Lastly, the client robustness to server failure could be enhanced by using a timeout mechanism to deal with an unresponsive server or stalled connection.