Assignment Simple KV Store Report

By: pnalawa@iu.edu

Design:

- We have two parts: **client** and **server**.
- The server can be connected with a **port by giving it via arguments**.
- The server can accept multiple client connections concurrently via **multi-threading**.
- The **max concurrent connections** set are **1024** as per the **memcache protoco**l.
- We are using **TCP** as the underlying communication between client and server with low level **socket programming**.
- Server spawns a new thread for each client connected.

Operations:

- We have implemented 3 operations:
    1. SET
    2. GET
    3. END
- **SET**
    - **Command:**
        - set <key> <size>\r\n <value>\r\n
    - If length is less not equal to 4 it will return ERROR\r\n
    - If key length is greater than 250, it will return ERROR\r\n
    - If the key is already present in the kv store, it will just update it.
    - If <size> is not equal to the <value> it will return ERROR\r\n
- **GET**
    - **Command**
        - get <key>\r\n
    - If length is less not equal to 2 it will return ERROR\r\n
    - If key does not exist it returns NOT-FOUND\r\n
    - If key exists it returns
        - VALUE <key> <size>\r\n <value>\r\n END\r\n
- **END**
    - **Command**
        - end\r\n
    - Closes client thread in server.

Limitations:

- Currently, the race condition where two writes or set operations are performed at same time is not handled. This can be taken as **future improvement** by **adding timestamp** in persistent storage.

- We are limiting the number of concurrent connections to 1024 as per memcache protocol.

Steps to reproduce:

- chmod 755 start_client.sh
- chmod 755 start_server.sh
- chmod 755 1-client.sh
- chmod 755 n-client.sh
- Start Server
    - **./start_server.sh <server_port>**
- You can run **multiple clients** by this command
    - **./start_client.sh <server_port>**

Run tests:

- ./start_server.sh 5001
- ./1-client.sh
- ./n-client.sh

Tests Output:

```
➜  memcached-lite git:(master) ✗ ./n-client.sh
test_success_update_set_command (tests.test_n_client.TestMultipleClient) ... STORED

STORED

STORED

STORED

/opt/local/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/unittest/case.py:550
: ResourceWarning: unclosed <socket.socket fd=5, family=AddressFamily.AF_INET,
type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 59433), raddr=('127.0.0.1',
5001)>
  method()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
/opt/local/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/unittest/case.py:550
: ResourceWarning: unclosed <socket.socket fd=6, family=AddressFamily.AF_INET,
type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 59434), raddr=('127.0.0.1',
5001)>
```

```
  method()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
ok

----------------------------------------------------------------------
Ran 1 test in 0.012s

OK
➜  memcached-lite git:(master) ✗ ./1-client.sh
test_failure_set_command_invalid (tests.test_1_client.TestSingleClient) ... set test0 20 0
NOT_STORED

set test1 20 1
NOT_STORED

set test2 20 2
NOT_STORED

set test3 20 3
NOT_STORED

set test4 20 4
NOT_STORED

/opt/local/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/unittest/case.py:550
: ResourceWarning: unclosed <socket.socket fd=5, family=AddressFamily.AF_INET,
type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 59457), raddr=('127.0.0.1',
5001)>
  method()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
ok
test_failure_set_command_invalid_key_length (tests.test_1_client.TestSingleClient) ... set
test555555555555555555555555555555555555555555555555555555555555555555555555
555555555555555555555555555555555555555555555555555555555555555555555555555
555555555555555555555555555555555555555555555555555555555555555555555555555
555555555555555555555555555555555555555555555555555555555555555555555555555
555555555555555555555555555555555555555555555555555555555555555555555555555
555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555 500 0
ERROR

set
test555555555555555555555555555555555555555555555555555555555555555555555555
555555555555555555555555555555555555555555555555555555555555555555555555555
```

```
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
55555555555555555555555555555555555555555555555 500 1
ERROR

set
test5555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
55555555555555555555555555555555555555555555555 500 2
ERROR

set
test5555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
55555555555555555555555555555555555555555555555 500 3
ERROR

set
test5555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
5555555555555555555555555555555555555555555555555555555555555555555555555555
55555555555555555555555555555555555555555555555 500 4
ERROR

/opt/local/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/unittest/case.py:550
: ResourceWarning: unclosed <socket.socket fd=5, family=AddressFamily.AF_INET,
type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 59458), raddr=('127.0.0.1',
5001)>
  method()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
ok
```

```
test_success_set_command (tests.test_1_client.TestSingleClient) ... set test0 1 0
STORED

set test1 1 1
STORED

set test2 1 2
STORED

set test3 1 3
STORED

set test4 1 4
STORED

/opt/local/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/unittest/case.py:550
: ResourceWarning: unclosed <socket.socket fd=5, family=AddressFamily.AF_INET,
type=SocketKind.SOCK_STREAM, proto=0, laddr=('127.0.0.1', 59459), raddr=('127.0.0.1',
5001)>
  method()
ResourceWarning: Enable tracemalloc to get the object allocation traceback
ok

----------------------------------------------------------------------
Ran 3 tests in 0.029s

OK
```