Memcached Lite

Siddhant Godshalwar

1. Introduction

Memcached clients use TCP connections to connect to the server. Clients connect to a running Memcached server's port of choice, transmit commands to the server, read answers, and eventually kill the connection.

To end the session, no command is required to be sent. If a client decides it no longer needs the connection, it can simply terminate it. However, keep in mind that clients are advised to cache their connections rather than opening them again each time they need to save or receive data. This is so that Memcached, which is specifically made to operate very effectively with a very large number of open connections—up to and including more than a thousand—can do so.

1.1 SET Method

Specifically, the set command is whitespace delimited, and consists of two lines:
set <key> <value-size-bytes> \r\n
<value> \r\n

Note that this is a simpler version of the memcached protocol, in which the set command also accepts flags and expiry time, which we will ignore for this assignment.
The server should respond with either "STORED\r\n", or "NOT-STORED\r\n".

1.2 GET Method

Retrieving data is simpler: get <key>\r\n
The server should respond with two lines:
VALUE <key> <bytes> \r\n
<data block>\r\n

After all the items have been transmitted, the server sends the string "END\r\n"

2. TestCases

```
sigods@silo:~/repo/DistributedSystems$ python3 client.py
What should I execute?: set 443 221
STORED

Server said:  STORED

sigods@silo:~/repo/DistributedSystems$
```

```
sigods@silo:~$ cd repo/DistributedSystems/
sigods@silo:~/repo/DistributedSystems$ python3 server.py
In server...
Recieved a connection from ('129.79.247.195', 57144)
```

Fig:- SET Method

```
sigods@silo:~/repo/DistributedSystems$ python3 client.py
What should I execute?: get 0
Server said:  VALUE 0 1
1
END

sigods@silo:~/repo/DistributedSystems$
```

```
sigods@silo:~$ cd repo/DistributedSystems/
sigods@silo:~/repo/DistributedSystems$ python3 server.py
In server...
Recieved a connection from ('129.79.247.195', 57144)
Recieved a connection from ('129.79.247.195', 53806)
0 1
```

Fig:- GET Method

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

sigods@silo:~/repo/DistributedSystems$ python3 client.py
What should I execute?: set 0 223
NOT-STORED

Server said:  NOT-STORED

sigods@silo:~/repo/DistributedSystems$
```

```
sigods@silo:~$ cd repo//DistributedSystems/
sigods@silo:~/repo/DistributedSystems$ python3 server.py
In server...
Recieved a connection from ('129.79.247.195', 42664)
```

Fig:- Same Key entered

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

sigods@silo:~/repo/DistributedSystems$ python3 client.py
What should I execute?: get 778
Server said:  KEY NOT FOUND
sigods@silo:~/repo/DistributedSystems$
```

```
sigods@silo:~$ cd repo/DistributedSystems/
sigods@silo:~/repo/DistributedSystems$ python3 server.py
In server...
Recieved a connection from ('129.79.247.195', 57144)
Recieved a connection from ('129.79.247.195', 53806)
0 1
Recieved a connection from ('129.79.247.195', 52952)
0 1
key1 value1
33sid 44bt
22sm sid
23uy priti
443 221
```

Fig:- Key not found

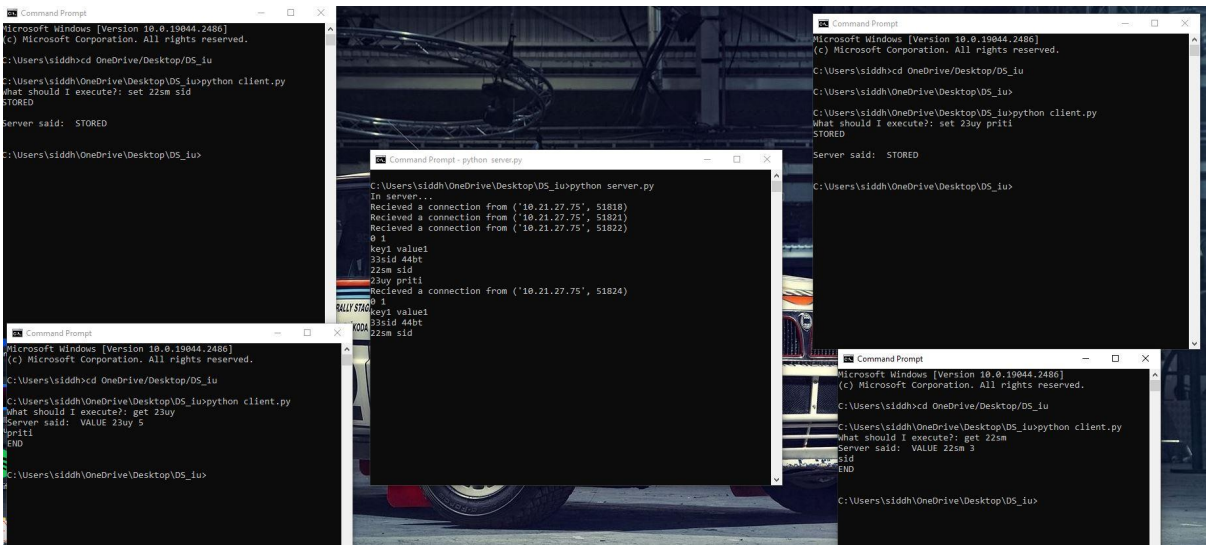


Fig:- Multi Client

3. Limitations:

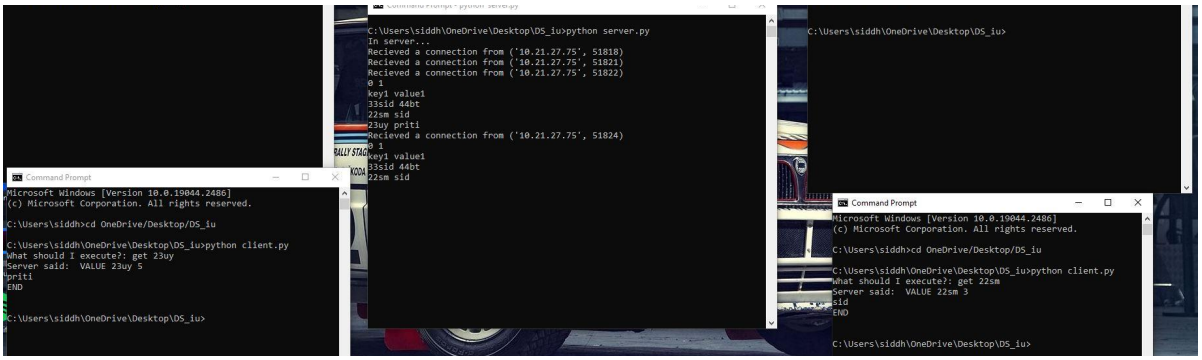
- The client number is fixed at 5 and not more than 5 as the queue length is set to 5.

4. Memcache for General Client:

```
C:\Users\siddh>python
Python 3.9.0a6 (tags/v3.9.0a6:bc1c8af, Apr 27 2020, 21:05:45) [MSC v.1924 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from pymemcache.client.base import Client
>>> Client('127.0.0.1:6942')
<pymemcache.client.base.Client object at 0x00000286688439A0>
>>> client=Client('127.0.0.1:6942')
>>> client.set('some_key', 'some_value')
```

This is the output we get for a generalized client setup over a memcache id from any server. We are able to execute both commands, easily.

5. Concurrency:



When we have more than one client running parallely, it is difficult to identify which client gets to run the set command as the sleep from the first one is still in action.
This is a prime example of concurrency