Memcached-lite

**AIM:** To mimic a Memcached server functionality with a persistent memory in Java.

**File Structure:**

/src

-Server.java

-ClientHandle.java

-LookUpCache.java

-Client.java

-cache.json

-Makefile

-jackson-annotations-2.13.3.jar

-jackson-core-2.13.3.jar

-jackson-databind-2.13.3.jar

**Dependencies:** Please find the included jar files for “jackson-fasterxml” included in “src” directory.

**Implementation:**

**Server:**

In java, I have used the abstraction to create a custom class “LookUpCache” from HashMap. In this class we save the persistent file to save all the key-value pairs that our server receives, that saves in JSON format. I have used “ReaderWriterLock” to synchronize between different clients and to eliminate any problems caused by concurrent updates and reads. To make the server compatible with standard “memcache” libraries I have introduced a dummy flags, so our server response after getting a GET request will be:

VALUE <key> <flag> <value-bytes> \r\n (this <flag> is the dummy response)

<value> \r\n

**Client:**

A client will connect to the server with the given host address and port. We can terminate a client by giving the escape character(!q) as an input. For a SET method we need to provide the value as an input after providing a valid SET command.

**Working:**

**Makefile:**

In the Makefile, we can use the “runServer” command to run the server by providing port, cache\_path, an ideal command will be:

make runServer port=8080 cache\_path=cache.json

To run the client, we can use the “runClient” command by providing the make command like:

make runClient addr=127.0.0.1 port=8080

We can use the “clean” command to delete the existing java compiles class’s to force recompile them.

make clean

**Output:**

**Client-1:**

Text

Description automatically generated

**Client-2**

Text

Description automatically generated

**Client using Standard pymemcache library:** Text

Description automatically generated