1 Test Cases

We have a series of test cases to ensure that our program runs correctly. These test cases are discussed in the subsections below.

1.1 Database Working Tests

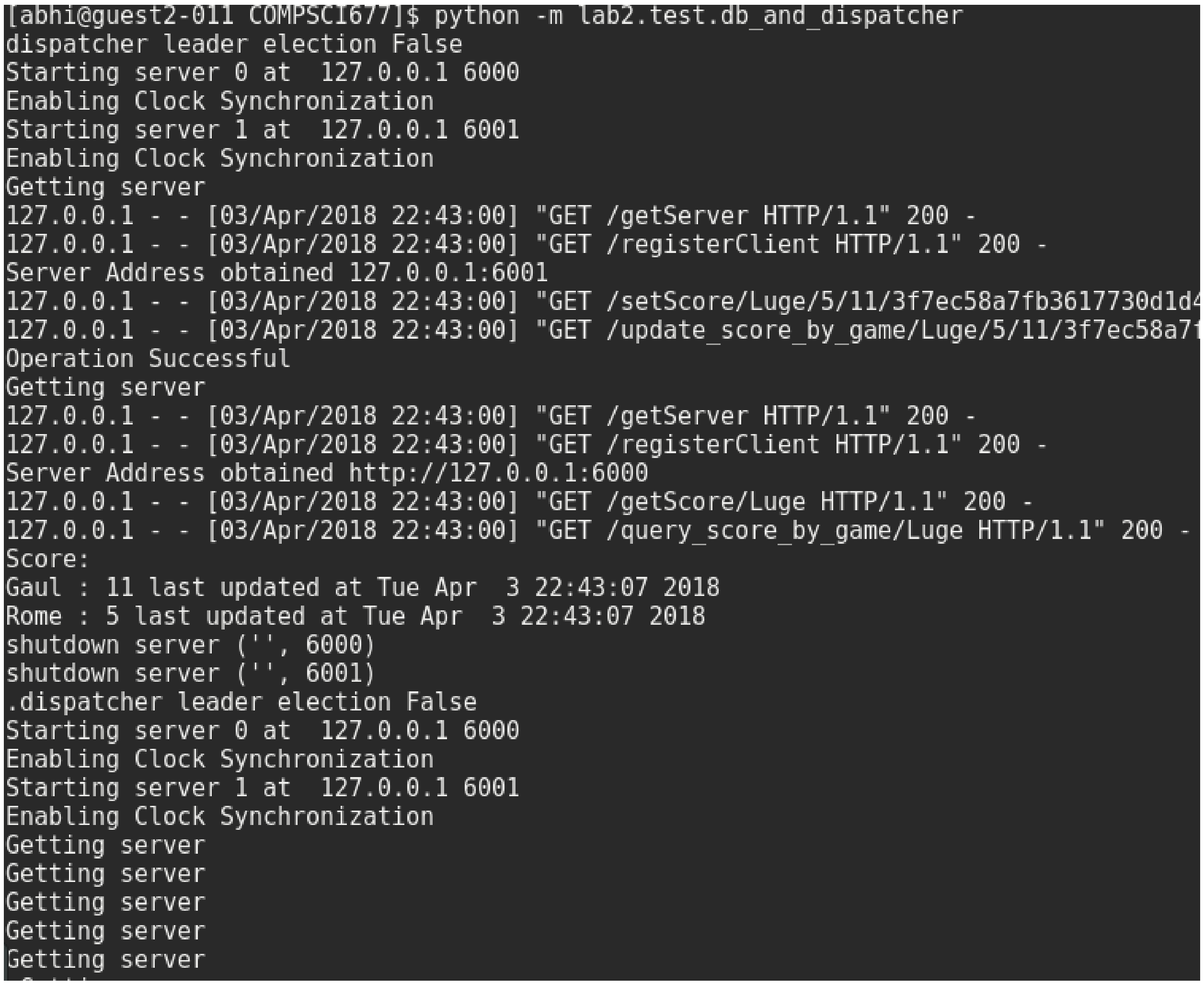
We have two tests for database. All these tests are in `test/db_and_dispatcher_tests.py`.

Database Locking Test In this test case we test that our database is correctly locked to ensure that only one thread writes to the files at once, hence, ensuring that the data is consistent. In this test, we create `Cacofonix` and use dispatcher to register it to one of the servers. `Cacofonix` then produce a series of updates by performing *incrementMedalTally* with randomly selected teams, and medal types. There are 1000 requests of this type and each of these requests are executed in a different thread. At the end we retrieve the exact medal count for each team and medal type, and check these counts with the test's internal database to make sure that the data retrieved is correct, hence, there is consistency in the database.

Update and retrieval test This is the basic test used to make sure that whatever data `Cacofonix` updates, the client is able to retrieve the exact same data.

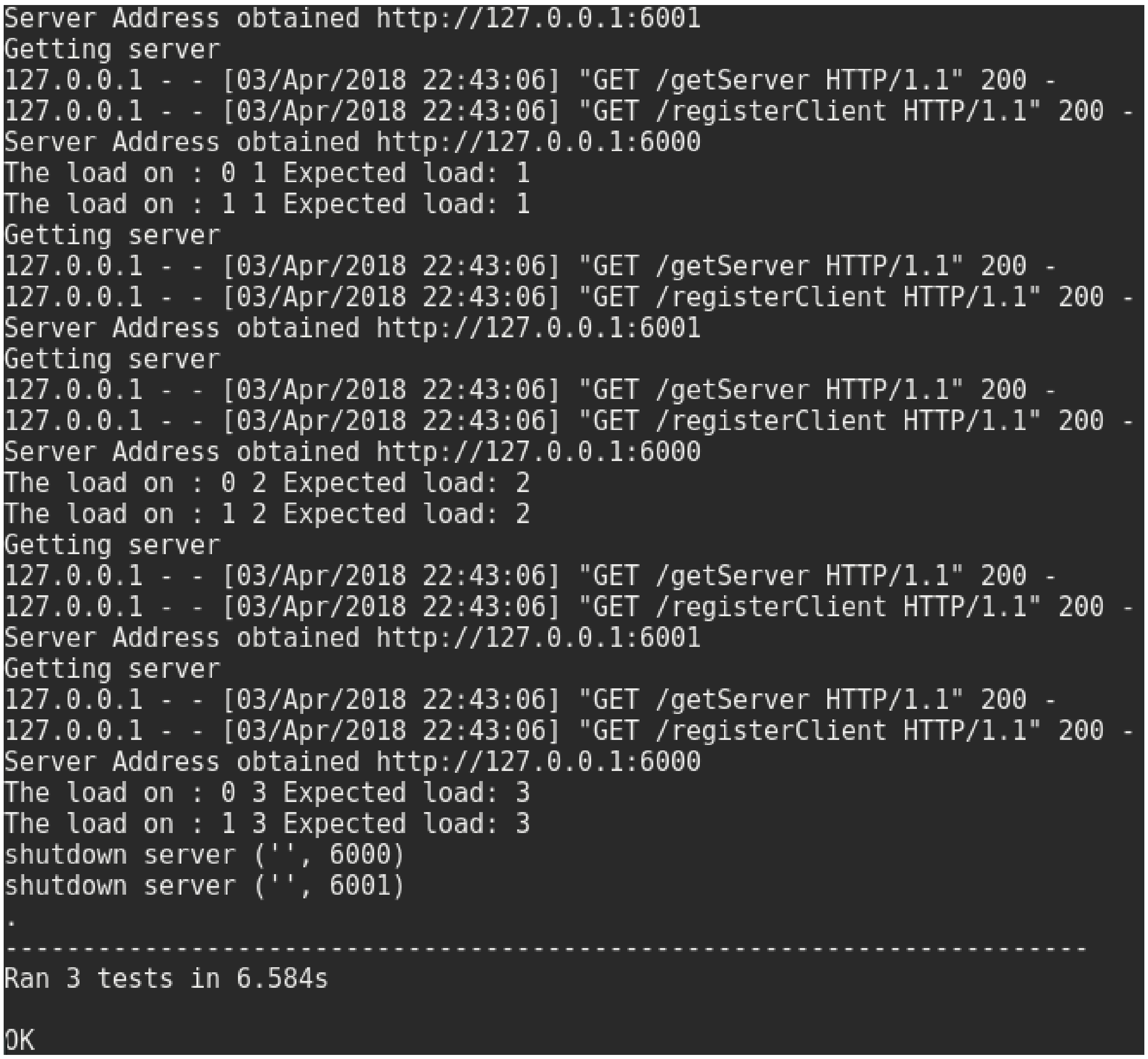
1.2 Load Balancing Test

This test is used to ensure that dispatcher indeed performs load balancing. This test is in `test/db_and_dispatcher_tests.py`. In this test, we create twice the number of clients as the number of servers. According to our scheme of load balancing, it should happen that the load on each client will be 2. At the end we check this.



```
[abhi@guest2-011 COMPSCI677]$ python -m lab2.test.db_and_dispatcher
dispatcher leader election False
Starting server 0 at  127.0.0.1 6000
Enabling Clock Synchronization
Starting server 1 at  127.0.0.1 6001
Enabling Clock Synchronization
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained 127.0.0.1:6001
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /setScore/Luge/5/11/3f7ec58a7fb3617730d1d4
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /update_score_by_game/Luge/5/11/3f7ec58a7f
Operation Successful
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained http://127.0.0.1:6000
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /getScore/Luge HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:00] "GET /query_score_by_game/Luge HTTP/1.1" 200 -
Score:
Gaul : 11 last updated at Tue Apr  3 22:43:07 2018
Rome : 5 last updated at Tue Apr  3 22:43:07 2018
shutdown server ('', 6000)
shutdown server ('', 6001)
.dispatcher leader election False
Starting server 0 at  127.0.0.1 6000
Enabling Clock Synchronization
Starting server 1 at  127.0.0.1 6001
Enabling Clock Synchronization
Getting server
Getting server
Getting server
Getting server
Getting server
```

Figure 1: DB and Dispatcher Start



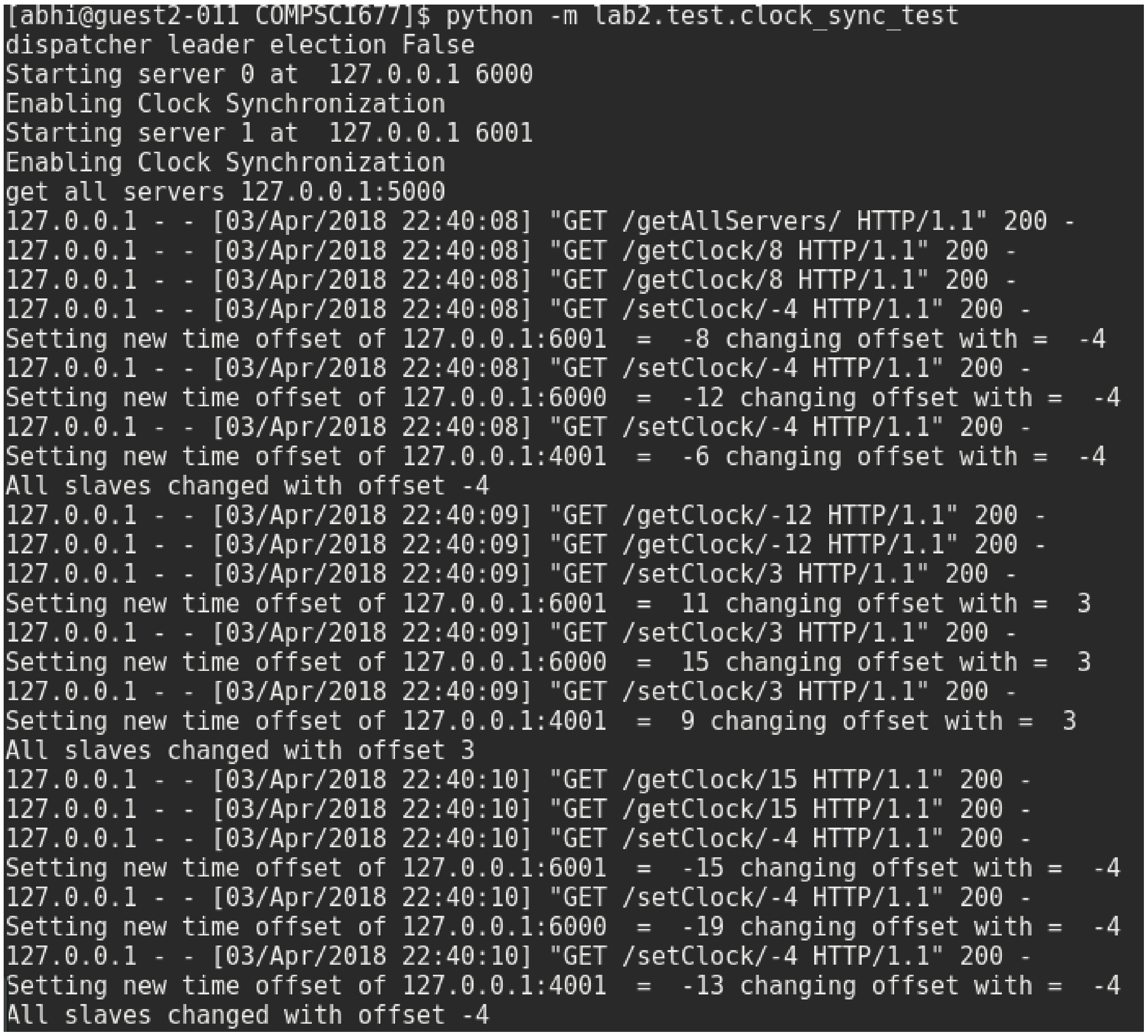
```
Server Address obtained http://127.0.0.1:6001
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained http://127.0.0.1:6000
The load on : 0 1 Expected load: 1
The load on : 1 1 Expected load: 1
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained http://127.0.0.1:6001
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained http://127.0.0.1:6000
The load on : 0 2 Expected load: 2
The load on : 1 2 Expected load: 2
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained http://127.0.0.1:6001
Getting server
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /getServer HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:43:06] "GET /registerClient HTTP/1.1" 200 -
Server Address obtained http://127.0.0.1:6000
The load on : 0 3 Expected load: 3
The load on : 1 3 Expected load: 3
shutdown server ('', 6000)
shutdown server ('', 6001)
.
----------------------------------------------------------------------
Ran 3 tests in 6.584s

OK
```

Figure 2: DB and Dispatcher Start End

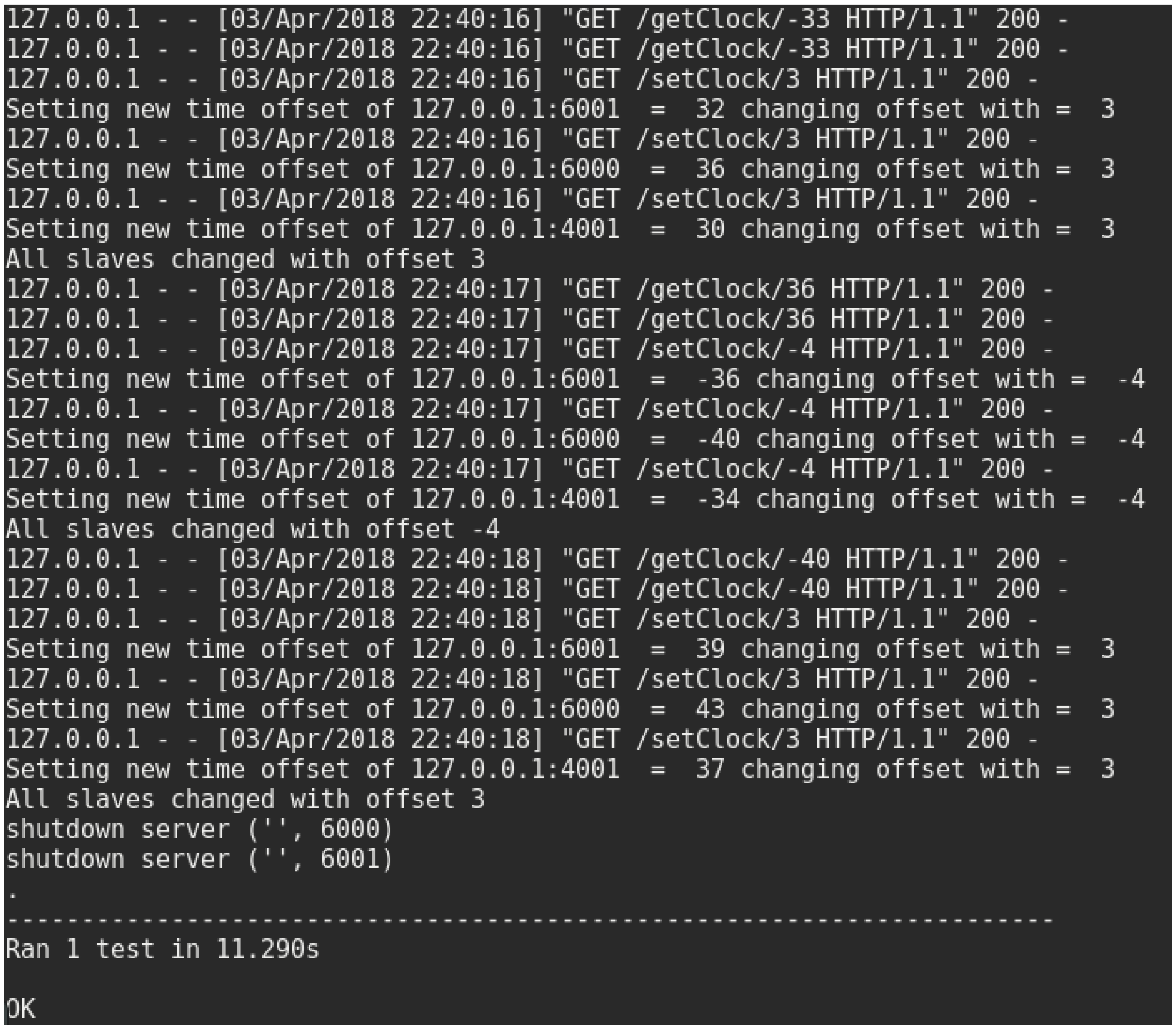
1.3 Clock Synchronization Test

In this test, we ensure two main properties of clock synchronization, (i) drift between each two servers is less than δ, and (ii) the Berkely clock synchronization algorithm is correct. The code for this test resides in `test/clock_sync_test.py`. In this test we create 2 front end servers and 1 database server (recall that database server also takes part in the clock synchronization). For each server we set a series of random offsets, which has to be less than δ and store these offsets in an internal offsets array. We then perform the clock synchronization 10 times in both the front end servers and in the internal offsets. We then check that the times of each front end server matches with the internal offset. Moreover, we also check that time difference between each clock is less than δ.



```
[abhi@guest2-011 COMPSCI677]$ python -m lab2.test.clock_sync_test
dispatcher leader election False
Starting server 0 at  127.0.0.1 6000
Enabling Clock Synchronization
Starting server 1 at  127.0.0.1 6001
Enabling Clock Synchronization
get all servers 127.0.0.1:5000
127.0.0.1 - - [03/Apr/2018 22:40:08] "GET /getAllServers/ HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:08] "GET /getClock/8 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:08] "GET /getClock/8 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:08] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6001  =  -8 changing offset with =  -4
127.0.0.1 - - [03/Apr/2018 22:40:08] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6000  =  -12 changing offset with =  -4
127.0.0.1 - - [03/Apr/2018 22:40:08] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:4001  =  -6 changing offset with =  -4
All slaves changed with offset -4
127.0.0.1 - - [03/Apr/2018 22:40:09] "GET /getClock/-12 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:09] "GET /getClock/-12 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:09] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6001  =  11 changing offset with =  3
127.0.0.1 - - [03/Apr/2018 22:40:09] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6000  =  15 changing offset with =  3
127.0.0.1 - - [03/Apr/2018 22:40:09] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:4001  =  9 changing offset with =  3
All slaves changed with offset 3
127.0.0.1 - - [03/Apr/2018 22:40:10] "GET /getClock/15 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:10] "GET /getClock/15 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:10] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6001  =  -15 changing offset with =  -4
127.0.0.1 - - [03/Apr/2018 22:40:10] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6000  =  -19 changing offset with =  -4
127.0.0.1 - - [03/Apr/2018 22:40:10] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:4001  =  -13 changing offset with =  -4
All slaves changed with offset -4
```

Figure 3: Clock Synchronization Start



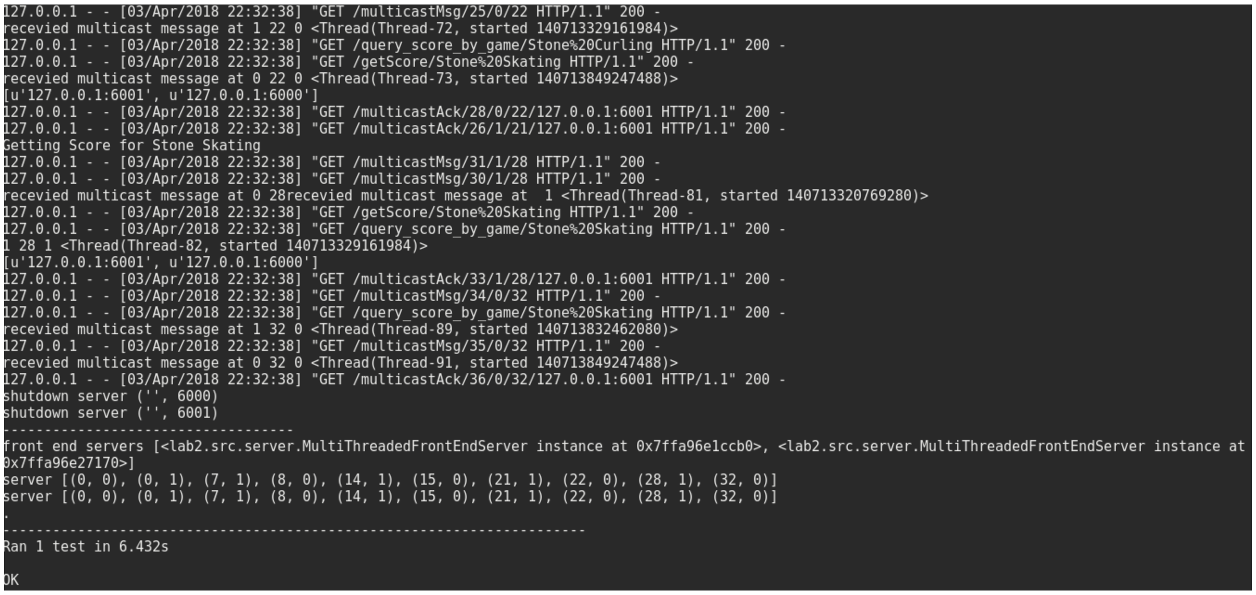
```
127.0.0.1 - - [03/Apr/2018 22:40:16] "GET /getClock/-33 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:16] "GET /getClock/-33 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:16] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6001  =  32 changing offset with =  3
127.0.0.1 - - [03/Apr/2018 22:40:16] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6000  =  36 changing offset with =  3
127.0.0.1 - - [03/Apr/2018 22:40:16] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:4001  =  30 changing offset with =  3
All slaves changed with offset 3
127.0.0.1 - - [03/Apr/2018 22:40:17] "GET /getClock/36 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:17] "GET /getClock/36 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:17] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6001  =  -36 changing offset with =  -4
127.0.0.1 - - [03/Apr/2018 22:40:17] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6000  =  -40 changing offset with =  -4
127.0.0.1 - - [03/Apr/2018 22:40:17] "GET /setClock/-4 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:4001  =  -34 changing offset with =  -4
All slaves changed with offset -4
127.0.0.1 - - [03/Apr/2018 22:40:18] "GET /getClock/-40 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:18] "GET /getClock/-40 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:40:18] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6001  =  39 changing offset with =  3
127.0.0.1 - - [03/Apr/2018 22:40:18] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:6000  =  43 changing offset with =  3
127.0.0.1 - - [03/Apr/2018 22:40:18] "GET /setClock/3 HTTP/1.1" 200 -
Setting new time offset of 127.0.0.1:4001  =  37 changing offset with =  3
All slaves changed with offset 3
shutdown server ('', 6000)
shutdown server ('', 6001)
.
----------------------------------------------------------------------
Ran 1 test in 11.290s

OK
```

Figure 4: Clock Synchronization End

1.4 Total Multicast Ordering Test

In this test, we ensure that the total multicast ordering is correct. This test resides in `test/total_ordering_test.py`. In this test, we first start two front end servers, and two clients. Each of the clients performs periodic pull and sends 5 requests per second. Load Balancing of Dispatcher ensures that each client is registered with different server. After 2 seconds we exit the clients and shutdown the server. We then obtain all the events obtained from each front end server in the total multicast order. We then compare the two list of events obtained to make sure that each element of the list is same.



```
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastMsg/25/0/22 HTTP/1.1" 200 -
recevied multicast message at 1 22 0 <Thread(Thread-72, started 140713329161984)>
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /query_score_by_game/Stone%20Curling HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /getScore/Stone%20Skating HTTP/1.1" 200 -
recevied multicast message at 0 22 0 <Thread(Thread-73, started 140713849247488)>
[u'127.0.0.1:6001', u'127.0.0.1:6000']
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastAck/28/0/22/127.0.0.1:6001 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastAck/26/1/21/127.0.0.1:6001 HTTP/1.1" 200 -
Getting Score for Stone Skating
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastMsg/31/1/28 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastMsg/30/1/28 HTTP/1.1" 200 -
recevied multicast message at 0 28recevied multicast message at  1 <Thread(Thread-81, started 140713320769280)>
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /getScore/Stone%20Skating HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /query_score_by_game/Stone%20Skating HTTP/1.1" 200 -
1 28 1 <Thread(Thread-82, started 140713329161984)>
[u'127.0.0.1:6001', u'127.0.0.1:6000']
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastAck/33/1/28/127.0.0.1:6001 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastMsg/34/0/32 HTTP/1.1" 200 -
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /query_score_by_game/Stone%20Skating HTTP/1.1" 200 -
recevied multicast message at 1 32 0 <Thread(Thread-89, started 140713832462080)>
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastMsg/35/0/32 HTTP/1.1" 200 -
recevied multicast message at 0 32 0 <Thread(Thread-91, started 140713849247488)>
127.0.0.1 - - [03/Apr/2018 22:32:38] "GET /multicastAck/36/0/32/127.0.0.1:6001 HTTP/1.1" 200 -
shutdown server ('', 6000)
shutdown server ('', 6001)
----------------------------------
front end servers [<lab2.src.server.MultiThreadedFrontEndServer instance at 0x7ffa96e1ccb0>, <lab2.src.server.MultiThreadedFrontEndServer instance at
0x7ffa96e27170>]
server [(0, 0), (0, 1), (7, 1), (8, 0), (14, 1), (15, 0), (21, 1), (22, 0), (28, 1), (32, 0)]
server [(0, 0), (0, 1), (7, 1), (8, 0), (14, 1), (15, 0), (21, 1), (22, 0), (28, 1), (32, 0)]
.
----------------------------------------------------------------------
Ran 1 test in 6.432s

OK
```

Figure 5: Raffle Selection

This can be run by executing `python server.py --is_leader_election False --is_clock_sync False --is_raffle True`, `python database_server.py --is_leader_election False --is_clock_sync False`, and 2 clients using `python client_pull.py --dispatcher_ip_addr 127.0.0.1`.

1.5 Cases where program does not work correctly

We do not think that there are any such cases where our program does not work correctly.