PA-4: MULTI-THREADED WEB SERVER AND ITS LOAD TESTING

ANUBHAV JANA(22M2109)

1. Created the multithreaded web server using thread pool, mutex and conditional variable
2. Load testing the above server using closed loop testing, by varying concurrent number of users (threads) with think time and test duration as other parameters
3. Written a bash script to run load testing and output to a text file and use the text file as a source to plot graph using pyplot [Users vs Throughput] and [Users vs Avg RTT]
4. Make sure the particular process goes into one particular CPU core with all threads of that process into the same core to saturate the CPU, used “taskset”
5. MakeFile to compile and generate executable
6. Test memory leak using valgrind

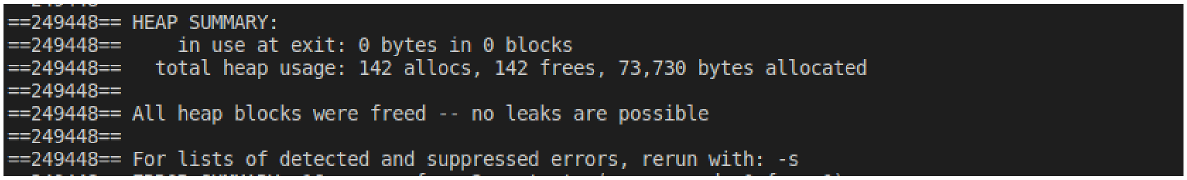
Commands:

ulimit -n 10000

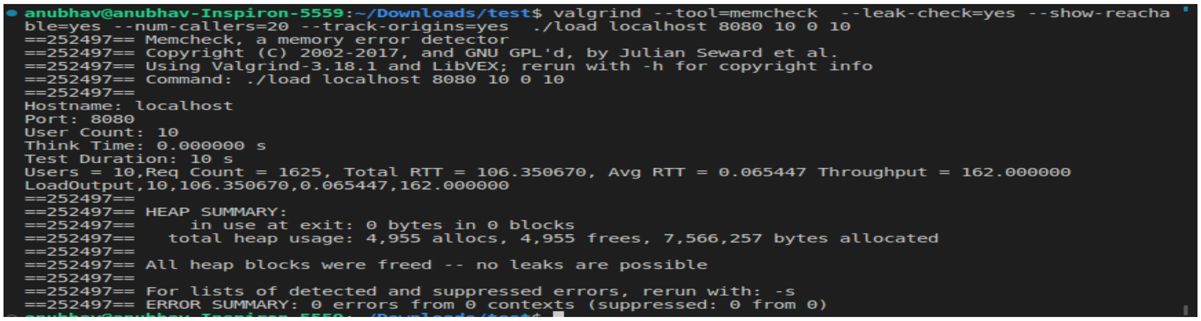
taskset -a -c 1 ./server 8080 [Run Server]

taskset –a –c 1,2,3 ./load localhost 8080 500 0.1 60 [Run load gen]

valgrind --tool=memcheck --leak-check=yes --show-reachable=yes --num-callers=20 --track-origins=yes ./server 8080



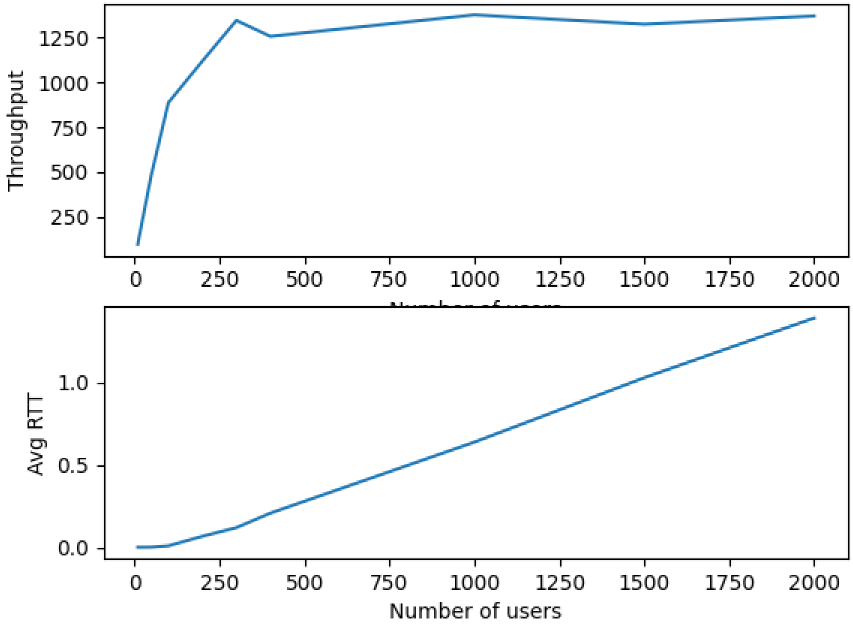
```
==249448== HEAP SUMMARY:
==249448==     in use at exit: 0 bytes in 0 blocks
==249448==   total heap usage: 142 allocs, 142 frees, 73,730 bytes allocated
==249448==
==249448== All heap blocks were freed -- no leaks are possible
==249448==
==249448== For lists of detected and suppressed errors, rerun with: -s
```

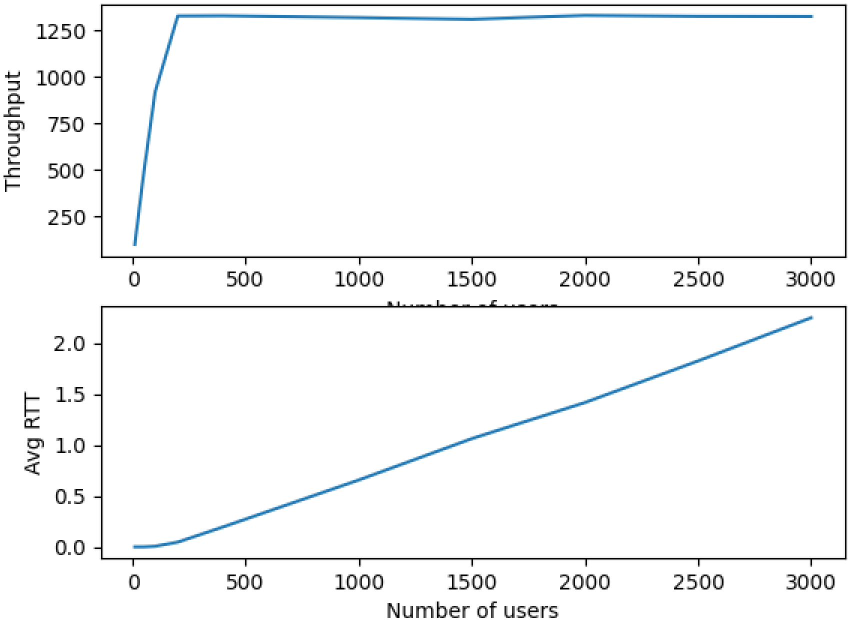


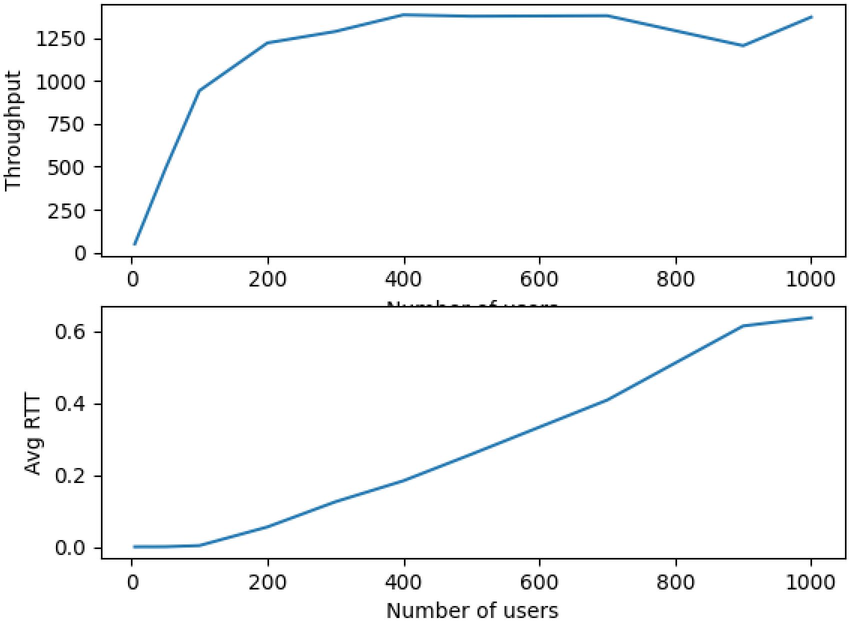
```
anubhav@anubhav-Inspiron-5559:~/Downloads/test$ valgrind --tool=memcheck  --leak-check=yes --show-reacha
ble=yes --num-callers=20 --track-origins=yes  ./load localhost 8080 10 0 10
==252497== Memcheck, a memory error detector
==252497== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==252497== Using Valgrind-3.18.1 and LibVEX; rerun with -h for copyright info
==252497== Command: ./load localhost 8080 10 0 10
==252497==
Hostname: localhost
Port: 8080
User Count: 10
Think Time: 0.000000 s
Test Duration: 10 s
Users = 10,Req Count = 1625, Total RTT = 106.350670, Avg RTT = 0.065447 Throughput = 162.000000
LoadOutput,10,106.350670,0.065447,162.000000
==252497==
==252497== HEAP SUMMARY:
==252497==     in use at exit: 0 bytes in 0 blocks
==252497==   total heap usage: 4,955 allocs, 4,955 frees, 7,566,257 bytes allocated
==252497==
==252497== All heap blocks were freed -- no leaks are possible
==252497==
==252497== For lists of detected and suppressed errors, rerun with: -s
==252497== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
anubhav@anubhav-Inspiron-5559:~/Downloads/test$
```

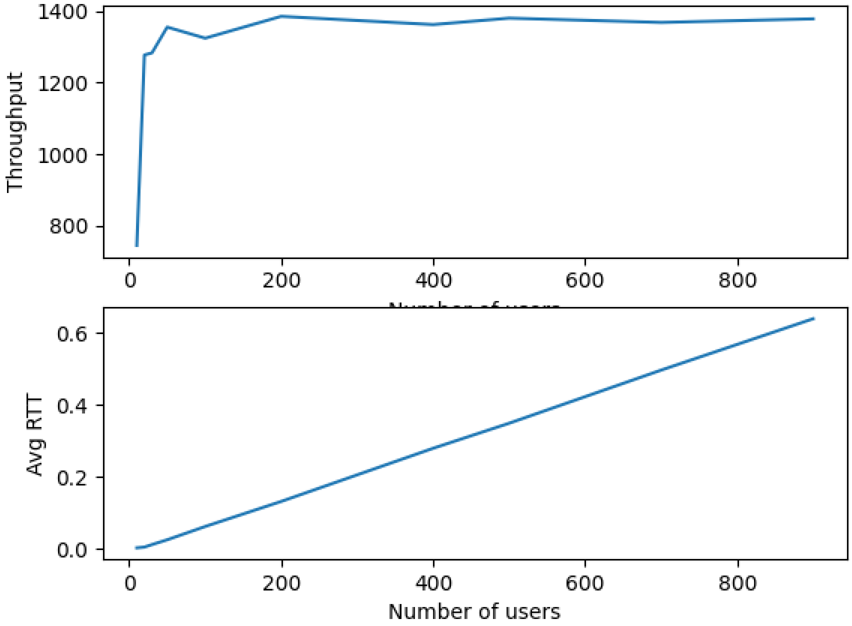
Wrote a special signal handler to handle CTRL+C to stop server and close the listen socket and free up all other threads.

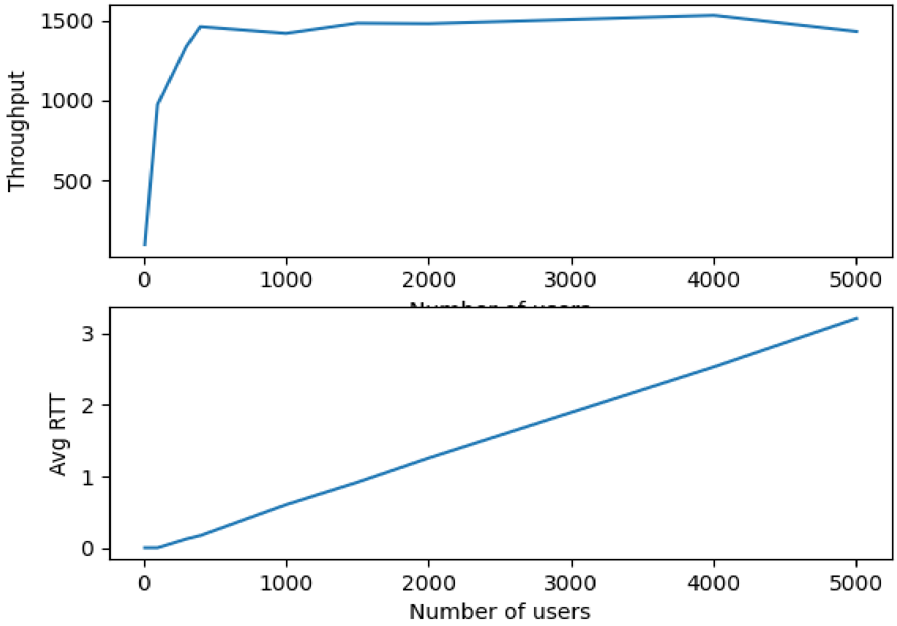
I have plotted 4 graphs using 4 different user counts with think time = 0.01 for the first image and think time=0.1 for the rest of the 3 images and test duration = 60




Throughput
1250
1000
750
500
250
0
500
1000
1500
2000
2500
3000
Number of users
Avg RTT
2.0
1.5
1.0
0.5
0.0
0
500
1000
1500
2000
2500
3000
Number of users


Throughput
1250
1000
750
500
250
0
0
200
400
600
800
1000
Number of users
Avg RTT
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
Number of users


Throughput
1400
1200
1000
800
0
200
400
600
800
Number of users
Avg RTT
0.6
0.4
0.2
0.0
0
200
400
600
800
Number of users


Throughput
1500
1000
500
0
1000
2000
3000
4000
5000
Number of users
Avg RTT
3
2
1
0
0
1000
2000
3000
4000
5000
Number of users

The zip contains the following files:

html_files

Load_gen.c

Server.c

MakeFile