Project 1: "Simple" – A simple multi-threaded HTTP/1.0 Web Server

Jian Wang(jianw3) Xunrong Li(xunrongl)

1. Concurrency Design

We create a thread pool of 10 and a queue of 50. All the 10 thread will be created during initialization. Then all the thread will be pending on the queue. Once the listen thread receives a TCP request, it will put the socket file descriptor into the queue and wake up one working thread.
We create a thread pool to save the effort to create threads for every TCP connection.

2. Bonus Features

- **CGI Implementation:**

We implement the CGI functionality in our simple web server. We create a cgi-bin folder under www folder and put a simple file adder.c there. The compiled file adder of this c file is used to add two integers and return the result to client.

- **Fighting DOS attacks:**

We use select before call recv() function and set a timeout value of 10 seconds. If the select timeout, then we will close the connection.

- **Benchmarking your server**

We use httperf to test the performance of our web server. We set the rate to 1000 which means 1000 request per second. We increase the connection number until 10000 there will be some failure (timeout). Following is the result of the httperf test for connection number of 5000/8000/1000: (You can also check it through out.txt)

```
httperf --hog --client=0/1 --server=128.237.209.204 --port=12121 --uri=/ --rate=1000 --
send-buffer=4096 --recv-buffer=16384 --num-conns=5000 --num-calls=1 --burst-
length=2
Maximum connect burst length: 35

Total: connections 5000 requests 5000 replies 5000 test-duration 5.807 s

Connection rate: 861.1 conn/s (1.2 ms/conn, <=168 concurrent connections)
Connection time [ms]: min 9.5 avg 64.9 max 1131.7 median 45.5 stddev 120.9
Connection time [ms]: connect 29.6
Connection length [replies/conn]: 1.000

Request rate: 861.1 req/s (1.2 ms/req)
Request size [B]: 68.0

Reply rate [replies/s]: min 970.7 avg 970.7 max 970.7 stddev 0.0 (1 samples)
Reply time [ms]: response 33.8 transfer 1.5
Reply size [B]: header 64.0 content 879.0 footer 0.0 (total 943.0)
Reply status: 1xx=0 2xx=5000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.00 system 4.70 (user 17.3% system 81.0% total 98.3%)
Net I/O: 850.2 KB/s (7.0*10^6 bps)
-----------------------------------------------------------------------------------------------
httperf --hog --client=0/1 --server=128.237.209.204 --port=12121 --uri=/ --rate=1000 --
send-buffer=4096 --recv-buffer=16384 --n
um-conns=8000 --num-calls=1 --burst-length=2
Maximum connect burst length: 39

Total: connections 8000 requests 8000 replies 8000 test-duration 8.029 s

Connection rate: 996.4 conn/s (1.0 ms/conn, <=92 concurrent connections)
Connection time [ms]: min 22.9 avg 42.9 max 1246.9 median 39.5 stddev 43.3
Connection time [ms]: connect 19.8
Connection length [replies/conn]: 1.000

Request rate: 996.4 req/s (1.0 ms/req)
Request size [B]: 68.0

Reply rate [replies/s]: min 990.9 avg 990.9 max 990.9 stddev 0.0 (1 samples)
Reply time [ms]: response 22.1 transfer 1.1
Reply size [B]: header 64.0 content 879.0 footer 0.0 (total 943.0)
Reply status: 1xx=0 2xx=8000 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.57 system 6.30 (user 19.6% system 78.4% total 98.0%)
Net I/O: 983.7 KB/s (8.1*10^6 bps)
-----------------------------------------------------------------------------------------------
httperf --hog --client=0/1 --server=128.237.209.204 --port=12121 --uri=/ --rate=1000 --
send-buffer=4096 --recv-buffer=16384 --num-conns=10000 --num-calls=1
Maximum connect burst length: 24

Total: connections 10000 requests 10000 replies 9986 test-duration 10.041 s

Connection rate: 995.9 conn/s (1.0 ms/conn, <=186 concurrent connections)
Connection time [ms]: min 11.7 avg 67.7 max 1278.1 median 53.5 stddev 71.5
Connection time [ms]: connect 33.0
Connection length [replies/conn]: 1.000

Request rate: 995.9 req/s (1.0 ms/req)
Request size [B]: 68.0

Reply rate [replies/s]: min 981.6 avg 993.4 max 1005.1 stddev 16.6 (2 samples)
Reply time [ms]: response 33.6 transfer 1.2
Reply size [B]: header 64.0 content 879.0 footer 0.0 (total 943.0)
Reply status: 1xx=0 2xx=9986 3xx=0 4xx=0 5xx=0

CPU time [s]: user 1.57 system 8.26 (user 15.7% system 82.3% total 98.0%)
Net I/O: 982.0 KB/s (8.0*10^6 bps)
```

- ## Testing suite for Simple

  We wrote two python script to test implementation of Simple and its concurrency feature.
  For testSimple.py, it should be executed as:

  ```
  python testSimple.py <port_number> <server_address>
  ```

  In this test, we test GET, HEAD, and POST request for existed resources, non-existed resources and long (over 1025 bytes) random uri input. And we get the correct response for all request.

  For testConcurrency.py, it should be executed as:

  ```
  python testConcurrency.py <port_number> <server_address>
  ```

  In this test, we use 300 threads send GET request to '/index.html' and get all success response.