

# README

Group 26:

Jiaqi Tian 1000263546

Si Yi Wu 1000430759

Yan Wan 1000287511

## BackEnd Instruction:

To run the backend crawler of the project, follow the procedures:

1. Put the urls of the websites that you want to crawl into **urls.txt**
2. Run the crawler with the python file `run_backend_test.py` by typing the following command:  

```
cd ~/lab3_group_26      # change the working directory to our submitted folder
python run_backend_test.py
```
3. You can check the result of the database created, which is in the same directory with the name of **greenLight.db**

## Access AWS:

The details of the AWS server running the FrontEnd is listed below:

Public IP: [34.235.2.173](https://ip-lookup.amazonaws.com/34.235.2.173)

Public DNS: `ec2-34-235-2-173.compute-1.amazonaws.com`

You can access our website by copying either the **public IP** or **DNS** into your web browser to access.

## Benchmark:

### **Result From Lab2:**

- Max number of requests that can be handled: 200
- Max Number of Requests Per Second (RPS): 233.52 [#/sec] (mean)
- Average Response Time: 27ms
- 99 Percentile Response Time: 856ms

### **Result From Current Lab3:**

- Max number of requests that can be handled: 190
- Max Number of Requests Per Second (RPS): 456.72 [#/sec] (mean)
- Average Response Time: 7ms
- 99 Percentile Response Time: 414ms

## **Comparison:**

From the results listed above, we can see that the max number of requests that can be handled remain roughly the same, while the response time has decreased significantly. The reason is that in Lab 2, the FrontEnd website also needs to compute the word count, the search history and also the most recent searches. These computations all take up a significant amount of time and therefore slows down the FrontEnd performance. However, in Lab3, there is no computation needed for the FrontEnd since FrontEnd will read all the data needed from the database file, and this saves a lot of computation time. The response time has decreased significantly, and therefore results in a larger number of requests per second too.

### Bonus:

For the bonus part, we multi-threaded the pagerank algorithm to speed up the pagerank score computation. The following result demonstrates the performance improvement. For the same urls.txt, the time to compute pagerank score was 0.02233s, while it decreased to 0.02195s with multi-threading.

```
Terminal
File Edit View Search Terminal Help
url=u'http://www.mapquest.com/maps?address=10+King%27s+College+Road&city=Toronto&state=ON&zipcode=M5S+3G4&country=CA'
HTTP Error 302: The HTTP server returned a redirect error that would lead to an infinite loop.
The last 30x error message was:
Found
document title=u'Computer Engineering Research Group'
num words=215
url=u'http://www.eecg.toronto.edu/Welcome.html'
time is 0.02233 ←
[(7, 0.026178095014334116),
(6, 0.022340387334243837),
(2, 0.016144258891122212),
(4, 0.014980610603030464),
(16, 0.013279850058523971),
(5, 0.013055074957968529),
(10, 0.01291746480767899),
(9, 0.012845701114302995),
(11, 0.012845701114302995),
(12, 0.012845701114302995),
(13, 0.012845701114302995),
(14, 0.012845701114302995),
(1, 0.01153846153846154)]
ug238:~/CSC326/bottle-0.12.7%
```

Before Multi-Threading

```
Terminal
File Edit View Search Terminal Help
url=u'http://www.mapquest.com/maps?address=10+King%27s+College+Road&city=Toronto&state=ON&zipcode=M5S+3G4&country=CA'
HTTP Error 302: The HTTP server returned a redirect error that would lead to an infinite loop.
The last 30x error message was:
Found
document title=u'Computer Engineering Research Group'
num words=215
url=u'http://www.eecg.toronto.edu/Welcome.html'
time is 0.02195 ←
[(7, 0.026178095014334116),
(6, 0.022340387334243837),
(2, 0.016144258891122212),
(4, 0.014980610603030464),
(16, 0.013279850058523971),
(5, 0.013055074957968529),
(10, 0.01291746480767899),
(9, 0.012845701114302995),
(11, 0.012845701114302995),
(12, 0.012845701114302995),
(13, 0.012845701114302995),
(14, 0.012845701114302995),
(1, 0.01153846153846154)]
ug238:~/CSC326/bottle-0.12.7%
```

After Multi-Threading

## Console Output for Reference:

ubuntu@ip-172-31-40-148:~\$ ab -n 190 -c 190

http://ec2-34-235-2-173.compute-1.amazonaws.com/?keywords=toronto

This is ApacheBench, Version 2.3 <\$Revision: 1528965 \$>

Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/

Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking ec2-34-235-2-173.compute-1.amazonaws.com (be patient)

Completed 100 requests

Finished 190 requests

Server Software: WSGIServer/0.1

Server Hostname: ec2-34-235-2-173.compute-1.amazonaws.com

Server Port: 80

Document Path: /?keywords=toronto

Document Length: 1542 bytes

Concurrency Level: 190

Time taken for tests: 0.416 seconds

Complete requests: 190

Failed requests: 0

Total transferred: 322430 bytes

HTML transferred: 292980 bytes

Requests per second: 456.72 [#/sec] (mean)

Time per request: 416.007 [ms] (mean)

Time per request: 2.190 [ms] (mean, across all concurrent requests)

Transfer rate: 756.89 [Kbytes/sec] received

### Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	1 2.3	0	9
Processing:	2	19 56.7	6	407
Waiting:	1	19 56.7	6	407
Total:	6	20 58.3	7	415

### Percentage of the requests served within a certain time (ms)

50%	7
66%	7
75%	7
80%	7
90%	8
95%	211
98%	215
99%	414
100%	415 (longest request)

