HTTP/1.0 Proxy Server

31/08/2016
CS425

Manikanta Reddy D
13265
manikant@

Implemented Options

1. A Http/1.0 Proxy Server
2. Flexibility to provide port number as a command line argument
3. Error handling in the form of (Error 500)
4. Ability to handle concurrent connections via child processes
5. Cap on number of child processes to avoid DDOS attacks

Deployment

1. Build
   make
   -- Creates a binary named *proxy*
2. Run
   proxy 1234
   Starts the proxy server listening on port 1234.
   Port number is the first argument and is optional (Defaults to 1234)

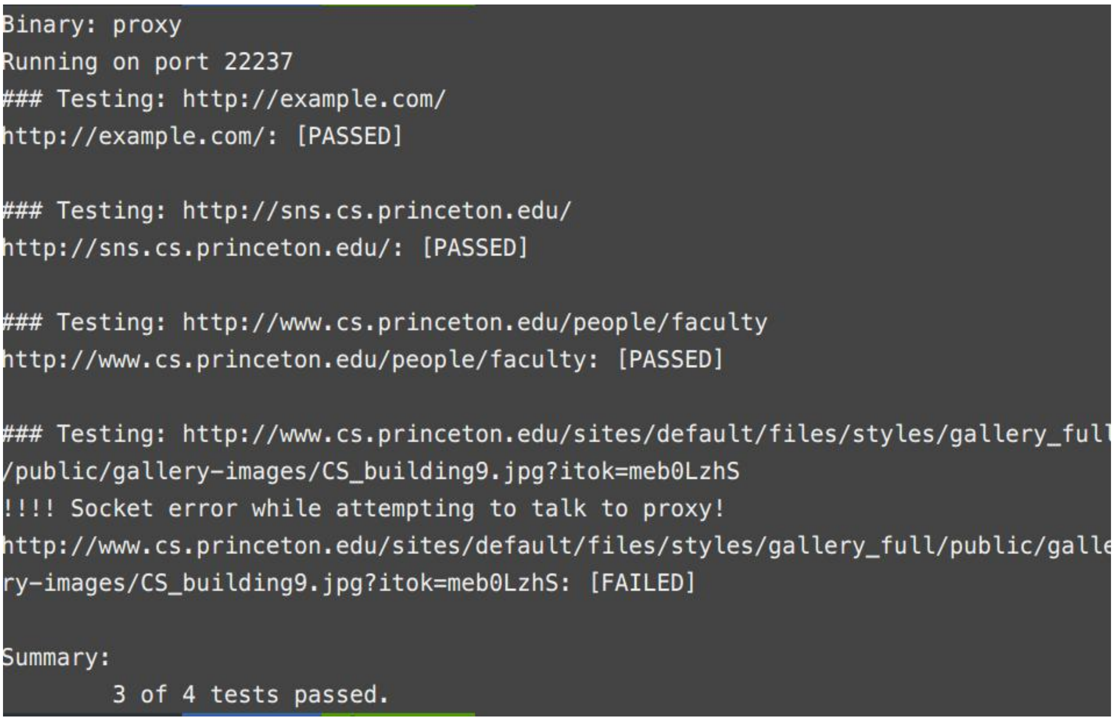
3. Connection to proxy can be established in multiple ways.
   Currently working pathways have been listed below
   A) Configuring proxy in firefox.
   B) Using proxy extension (omega proxy switcher) in chrome.
   C) Telnet to connect to proxy and there by make http requests.

Design Choices

1. Buffer length 4096 (hence the request is bounded to be 4096)
2. Maximum 20 concurrent requests are allowed to the server
3. All errors are reformed as Internal Server Errors
4. To handle split requests the request end is read as two simultaneous '/r/n'

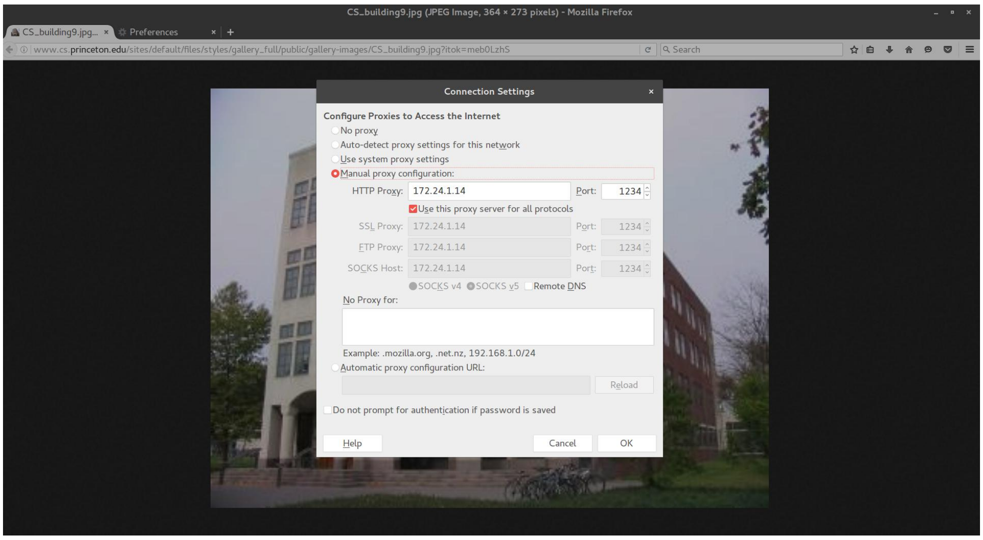
Tests

1. Currently using the *proxy_tester.py, only 3 out of 4 cases pass.*



```
Binary: proxy
Running on port 22237
### Testing: http://example.com/
http://example.com/: [PASSED]

### Testing: http://sns.cs.princeton.edu/
http://sns.cs.princeton.edu/: [PASSED]

### Testing: http://www.cs.princeton.edu/people/faculty
http://www.cs.princeton.edu/people/faculty: [PASSED]

### Testing: http://www.cs.princeton.edu/sites/default/files/styles/gallery_full
/public/gallery-images/CS_building9.jpg?itok=meb0LzhS
!!!! Socket error while attempting to talk to proxy!
http://www.cs.princeton.edu/sites/default/files/styles/gallery_full/public/galle
ry-images/CS_building9.jpg?itok=meb0LzhS: [FAILED]

Summary:
        3 of 4 tests passed.
```

2. But the fourth case passes successfully in a browser.



CS_building9.jpg (JPEG Image, 364 × 273 pixels) - Mozilla Firefox

www.cs.princeton.edu/sites/default/files/styles/gallery_full/public/gallery-images/CS_building9.jpg?itok=meb0LzhS

Connection Settings

Configure Proxies to Access the Internet

○ No proxy
○ Auto-detect proxy settings for this network
○ Use system proxy settings
● Manual proxy configuration:

HTTP Proxy: 172.24.1.14 Port: 1234

☑ Use this proxy server for all protocols

SSL Proxy: 172.24.1.14 Port: 1234
FTP Proxy: 172.24.1.14 Port: 1234
SOCKS Host: 172.24.1.14 Port: 1234

● SOCKS v4 ● SOCKS v5 □ Remote DNS

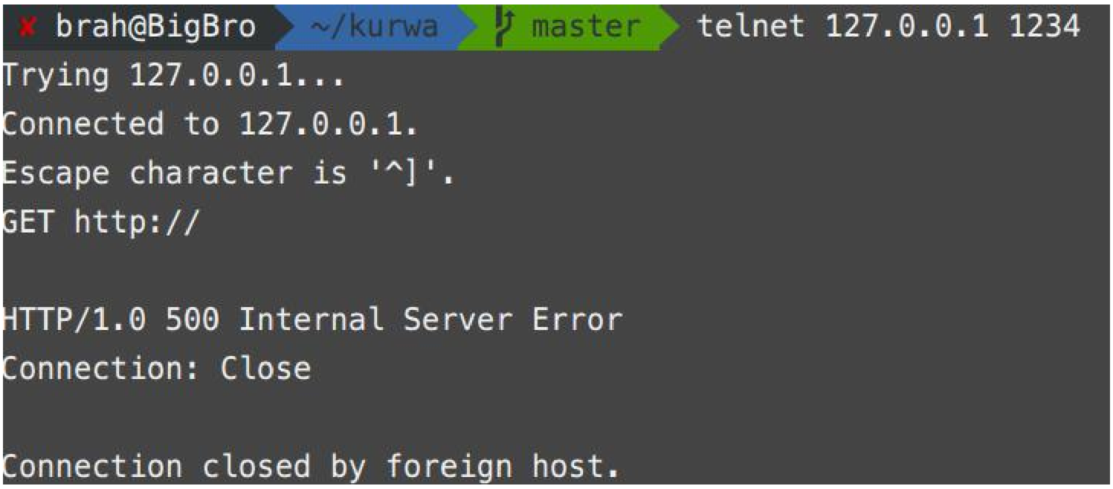
No Proxy for:

Example: .mozilla.org, .net.nz, 192.168.1.0/24

○ Automatic proxy configuration URL:

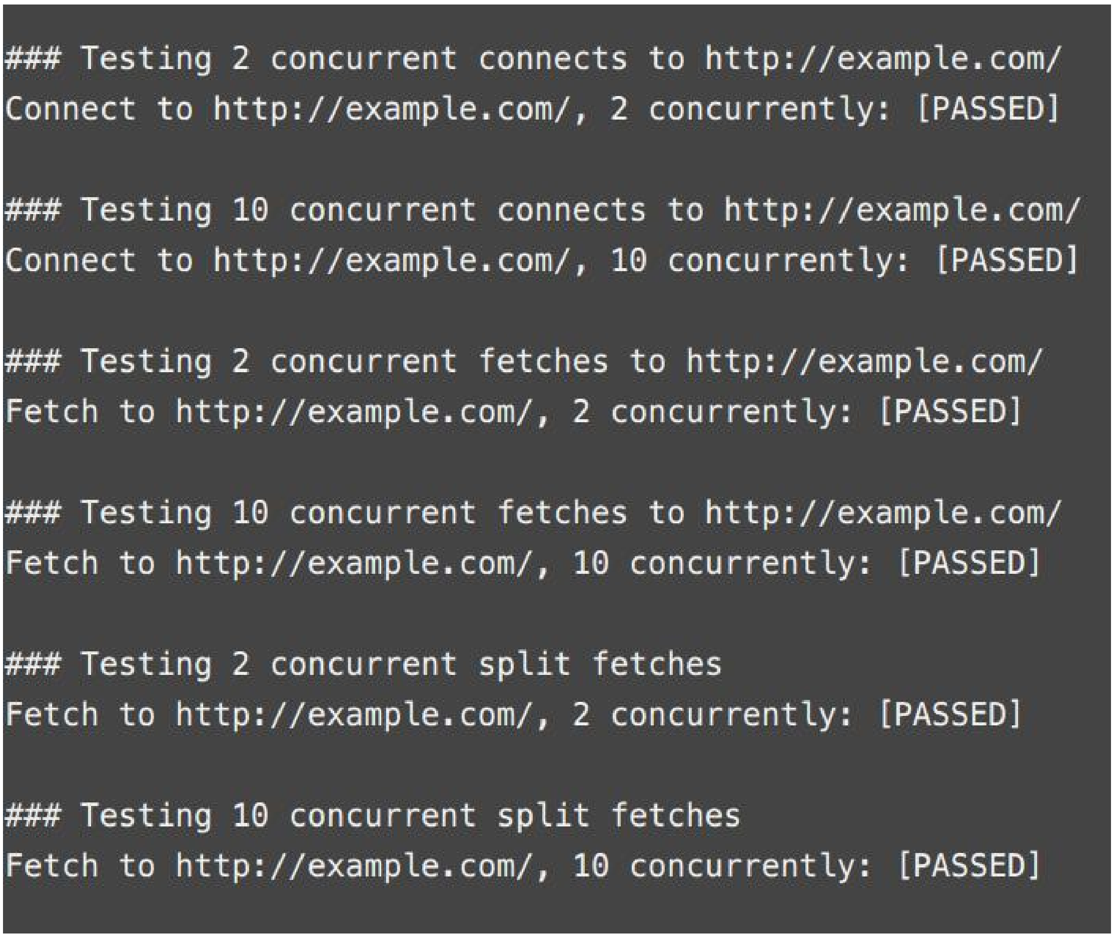
□ Do not prompt for authentication if password is saved

3. 500 Internal error for mal formed request



```
✘ brah@BigBro  ~/kurwa  ⎇ master  telnet 127.0.0.1 1234
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
GET http://

HTTP/1.0 500 Internal Server Error
Connection: Close

Connection closed by foreign host.
```

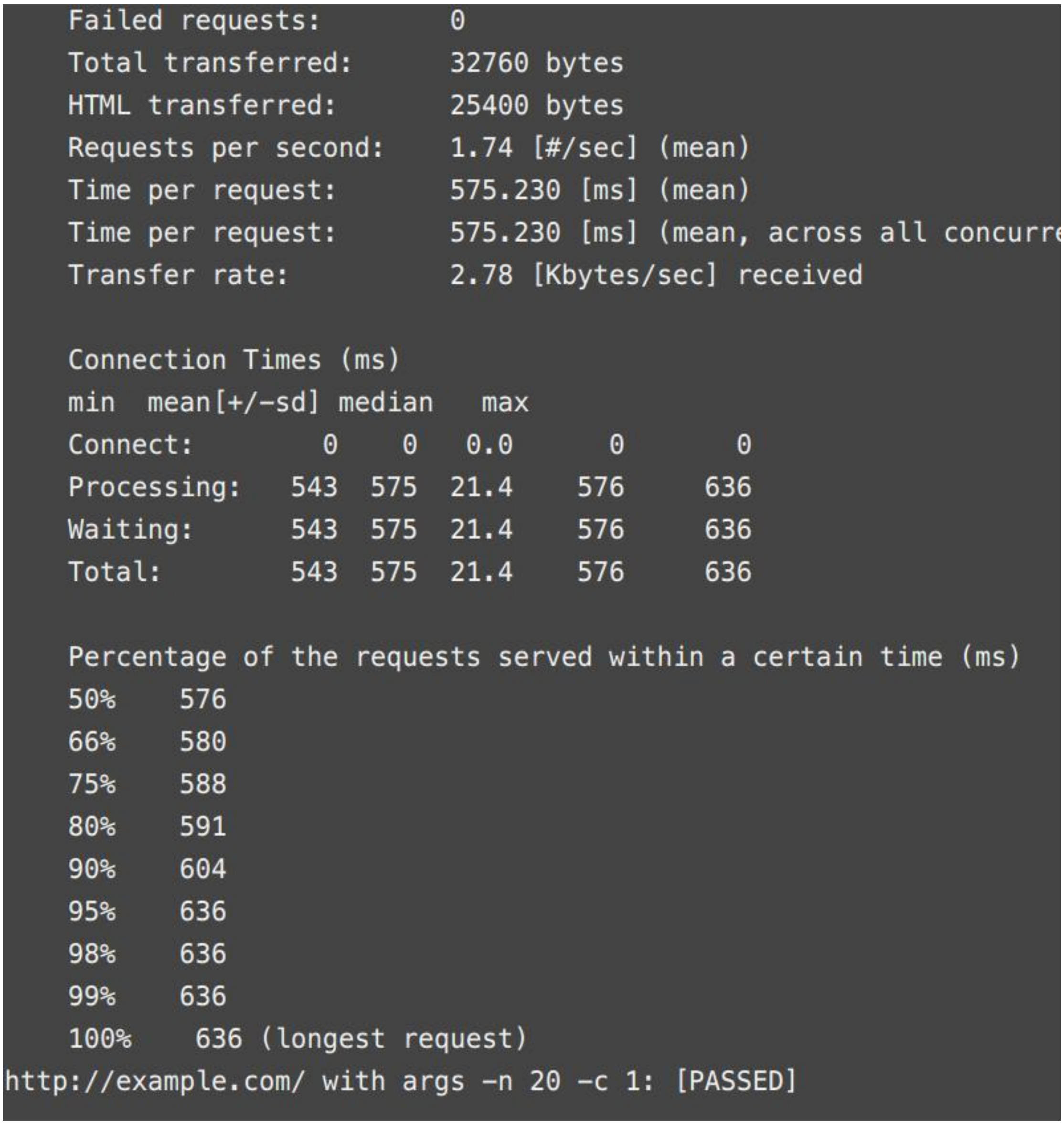
4. Concurrent requests are gracefully handled



```
### Testing 2 concurrent connects to http://example.com/
Connect to http://example.com/, 2 concurrently: [PASSED]

### Testing 10 concurrent connects to http://example.com/
Connect to http://example.com/, 10 concurrently: [PASSED]

### Testing 2 concurrent fetches to http://example.com/
Fetch to http://example.com/, 2 concurrently: [PASSED]

### Testing 10 concurrent fetches to http://example.com/
Fetch to http://example.com/, 10 concurrently: [PASSED]

### Testing 2 concurrent split fetches
Fetch to http://example.com/, 2 concurrently: [PASSED]

### Testing 10 concurrent split fetches
Fetch to http://example.com/, 10 concurrently: [PASSED]
```

5. Apached Benchmars

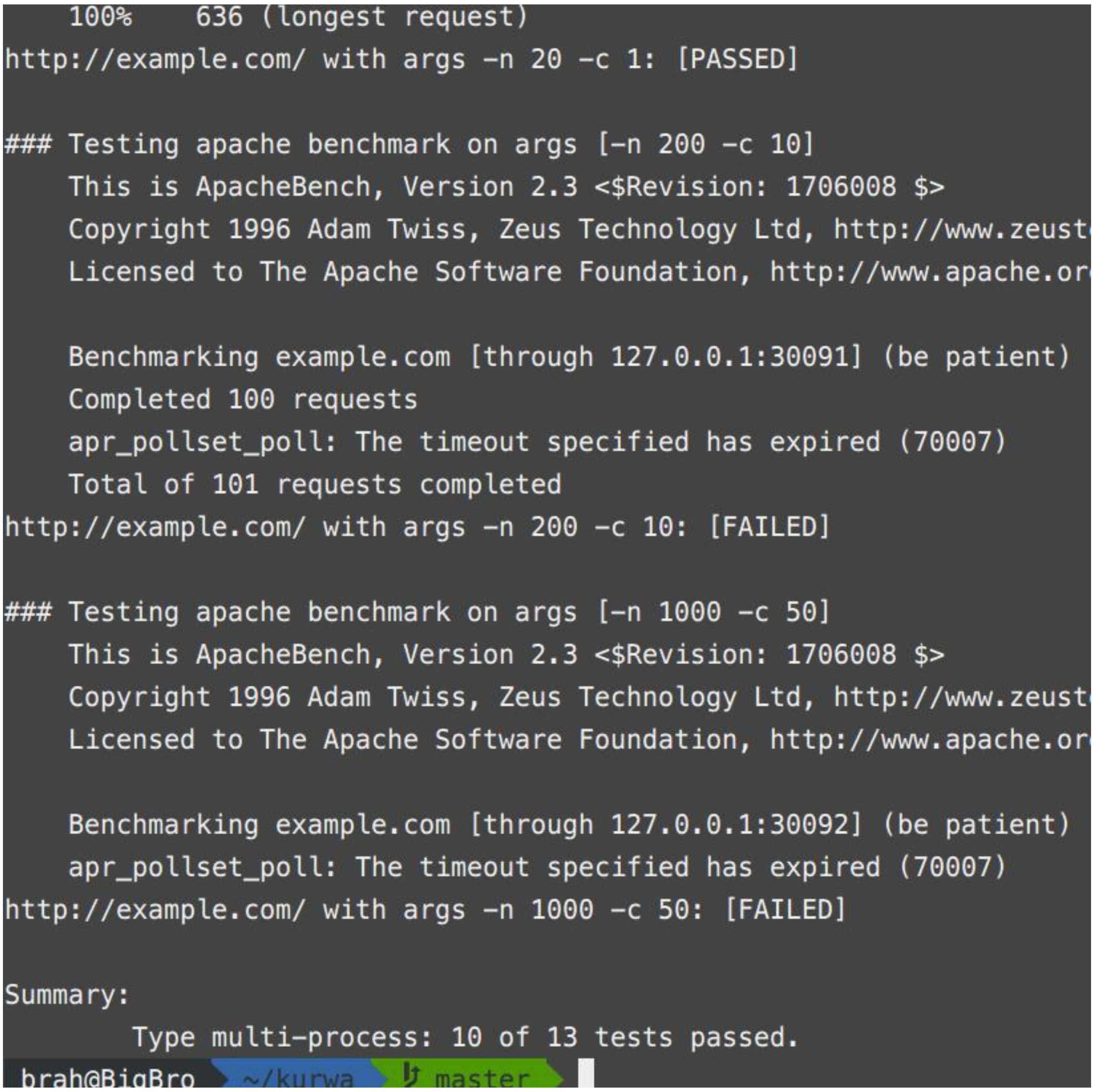
Only one out of the four benchmarks works on slower networks. In the cse network other pass out too.



```
    Failed requests:        0
    Total transferred:      32760 bytes
    HTML transferred:       25400 bytes
    Requests per second:    1.74 [#/sec] (mean)
    Time per request:       575.230 [ms] (mean)
    Time per request:       575.230 [ms] (mean, across all concurre
    Transfer rate:          2.78 [Kbytes/sec] received

    Connection Times (ms)
    min  mean[+/-sd] median   max
    Connect:        0    0   0.0      0       0
    Processing:   543  575  21.4    576     636
    Waiting:      543  575  21.4    576     636
    Total:        543  575  21.4    576     636

    Percentage of the requests served within a certain time (ms)
    50%    576
    66%    580
    75%    588
    80%    591
    90%    604
    95%    636
    98%    636
    99%    636
    100%    636 (longest request)
http://example.com/ with args -n 20 -c 1: [PASSED]
```

Anamolies

1. To pass the 4th case, adding a wait helps after sending
2. 10/13 cases pass in the 2nd script



```
    100%    636 (longest request)
http://example.com/ with args -n 20 -c 1: [PASSED]

### Testing apache benchmark on args [-n 200 -c 10]
    This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
    Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeust
    Licensed to The Apache Software Foundation, http://www.apache.or

    Benchmarking example.com [through 127.0.0.1:30091] (be patient)
    Completed 100 requests
    apr_pollset_poll: The timeout specified has expired (70007)
    Total of 101 requests completed
http://example.com/ with args -n 200 -c 10: [FAILED]

### Testing apache benchmark on args [-n 1000 -c 50]
    This is ApacheBench, Version 2.3 <$Revision: 1706008 $>
    Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeust
    Licensed to The Apache Software Foundation, http://www.apache.or

    Benchmarking example.com [through 127.0.0.1:30092] (be patient)
    apr_pollset_poll: The timeout specified has expired (70007)
http://example.com/ with args -n 1000 -c 50: [FAILED]

Summary:
        Type multi-process: 10 of 13 tests passed.
 brah@BigBro  ~/kurwa  master 
```

3.