# Report for Multithreaded Web server

HASSAAN FAYYAZ AHMED, Lahore University of Management Sciences Opposite Sector U, DHA Lahore Punjab 54810 Pakistan
MUHAMMAD QASIM HUNAIN, Lahore University of Management Sciences Opposite Sector U, DHA Lahore Punjab 54810 Pakistan

**ACM Reference format:**
Hassaan Fayyaz Ahmed and Muhammad Qasim Hunain. 2017. Report for Multithreaded Web server. *ACM Trans. Web* 0, 4, Article 00 (March 2017), **??** pages.
DOI: 0000001.0000001

 1559-1131/2017/3-ART00 $15.00
DOI: 0000001.0000001

## 1 EXPERIMENTATION

- **Does your server fail when overloaded?**
  It does not exactly fail. If overloaded by the requests, server checks for the availability of a worker thread in the threadpool. If no worker thread is free, server queues these requests. As soon as any worker thread gets free, the boss thread assigns this request to the recently free'd worker thread.

- **Does it just die or does it gracefully refuse connections?**
  We have introduced exception handling on the server side. So, in our testing we have observed delay in the response rather then just dying.

- **What are the limits of server(how much files/bytes client has retrieved from server before server failure etc.)?**
  We are getting a delayed response when the server is heavily bombarded with requests. The server is not failing.
  Note that this bombardment was done for a file of small size.

- **What is the throughput of the server at different number of threads of client and server?**
  Number of threads : 10000
  Number of requests per thread : 10000
  The time taken is 1292 seconds. This time only includes the processing time of the worker threads . Each time a worker execute completely the stop watch puased and when a worker thread is started the execution time is resumed from the exact time. The System was 4 cores processor and 8 GB of ram. It was not a physical system but a virtual machine which is connected through remote desktop. The OS we are using is Windows Server 2012 R2.

- **Vary the number of files (from few tens to thousands) of different sizes (from kBs to MBs) and experiment with throughput. Does the file caching affect the throughput?**
  We tested server from 450 bytes file and the Response time was above given.
  We also tested the server with 16 KB file with 10,000 Number of thread and 1,000 Number of requests. The total time taken by Server exceeded to 463 seconds.

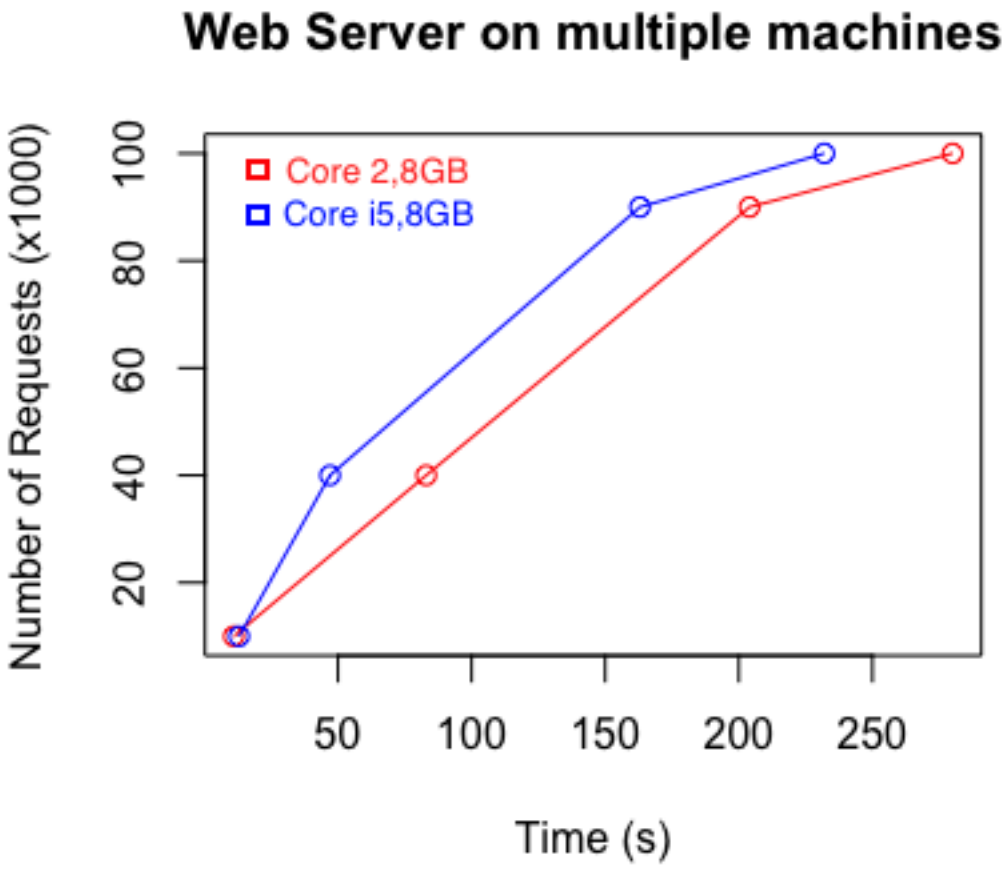
- **What is the affect of thread scheduling schemes?**
  **First Come First Serve :** The default scheduling scheme that we used is FCFS.

- **Experiment on different machines with dual/quad core, uni-processor (if available) and provide explanation for your observations. Also clearly specify the configuration of your ma- chine(s).**
  We experimented on 2 different machines
  (1) 2 Core processor with 8 GB ram it is an remote server it executes 100 threads and each thread has 100 requests it t took 11 second.
  (2) 2 Core processor with 8 GB ram it is an remote server it executes 200 threads and each thread has 200 requests it t took 83 second.
  (3) 2 Core processor with 8 GB ram it is an remote server it executes 300 threads and each thread has 300 requests it t took 204 second.

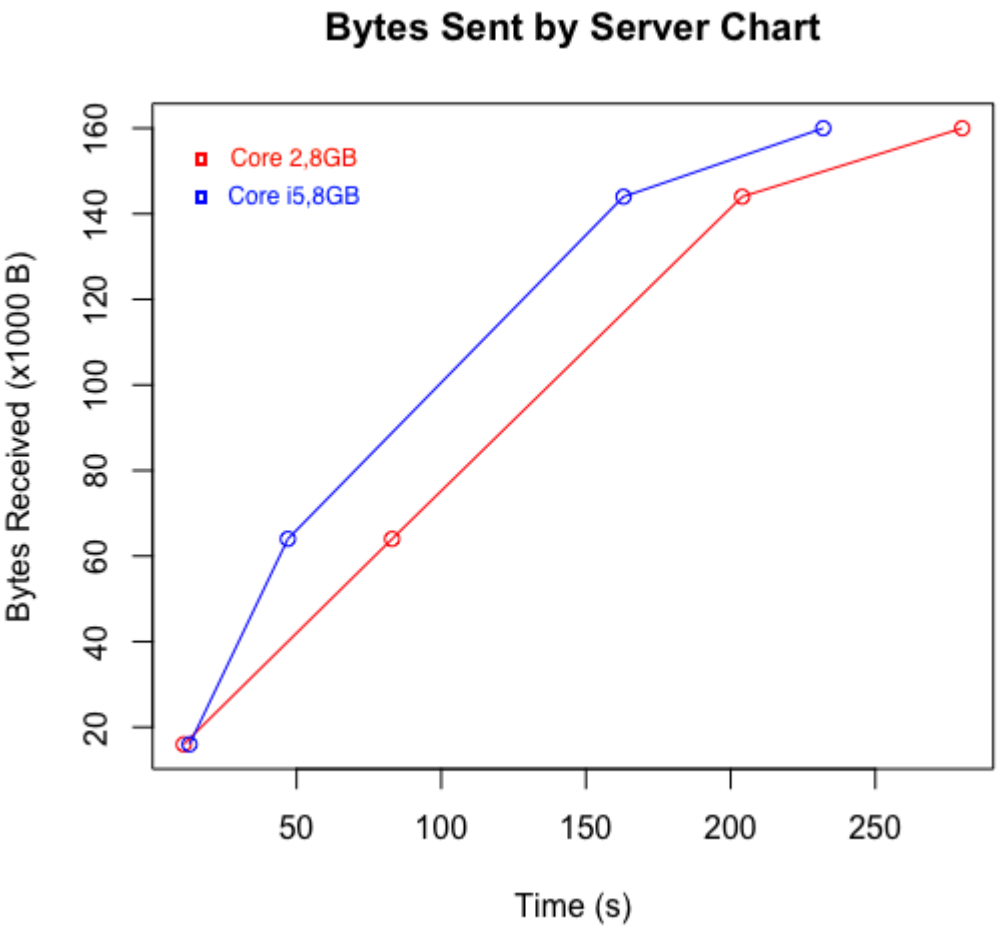
(4) 2 Core processor with 8 GB ram it is an remote server it executes 1000 threads and each thread has 100 requests it t took 280 second.
(5) Core i5 processor with 8 GB ram it is an remote server it executes 100 threads and each thread has 100 requests it t took 13 second.
(6) Core i5 processor with 8 GB ram it is an remote server it executes 200 threads and each thread has 200 requests it t took 47 second.
(7) Core i5 processor with 8 GB ram it is an remote server it executes 300 threads and each thread has 300 requests it t took 163 second.
(8) Core i5 processor with 8 GB ram it is an remote server it executes 1000 threads and each thread has 100 requests it t took 232 second.

## 2 GRAPH

### 2.1 Performance



### 2.2 Data Transfer Grapph




Received March 2017; revised March 2017; accepted March 2017