/\* CSci4061 Fall 2018 Project 3

\* Group ID: 111

\* Name: Michael Nguyen, Berni Duong, Catherine Ha

\* X500: nguy2571, duong142, haxxx214 \*/

**Performance Analysis Report**

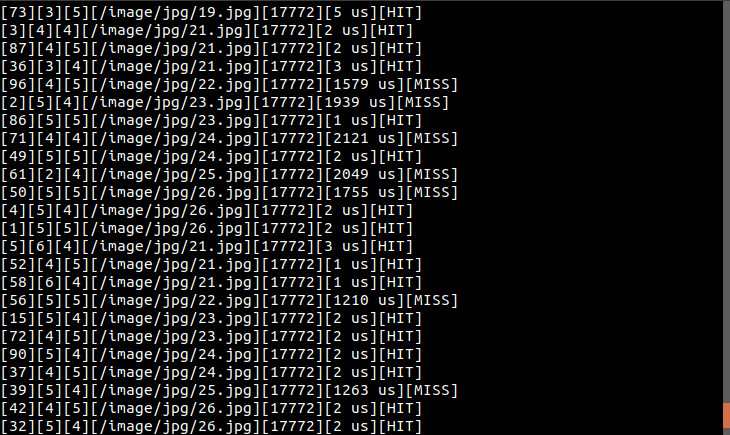
Cache HIT/MISS Example

*How to Run:*

In one terminal >>> ./web\_server 9000 <path\_to\_testing>/testing 100 100 0 100 100

In another terminal >>> wget -i <path-to-urls>/urls -O myres

*Output:*



*Explanation of Results:*

Almost every time the cache hits a MISS, it takes a lot longer to load in terms of us compared to when it is a HIT in cache. The HITs show the amount of time it takes to retrieve the request from cache whereas the time for the MISSes is the time it takes to read from the disk.

Running the Server with 100 Dispatcher Threads

*Note*: We used 100 for the queue length and cache entries.

*How to Run:*

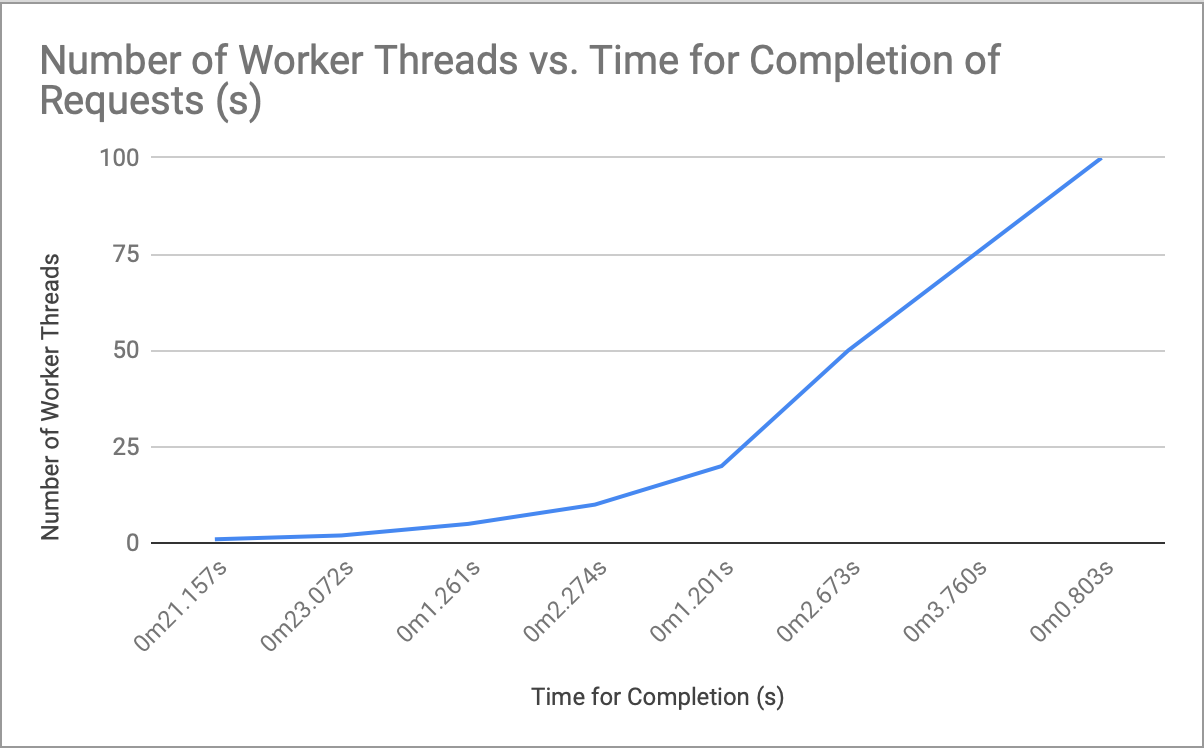
In terminal >>> ./web\_server 9000 <path\_to\_testing>/testing 100 n 0 100 100

\*\*\*where n is the num\_workers which changes per run as shown below in the data table

In another terminal >>> time cat <path \_to\_urls\_file> | xargs -n 1 -P 1 wget

*Data:*

|  |  |
| --- | --- |
| Number of Worker Threads | Time for Completion |
| 1 | 0m21.157s |
| 2 | 0m23.072s |
| 5 | 0m1.261s |
| 10 | 0m2.274s |
| 20 | 0m1.201s |
| 50 | 0m2.673s |
| 75 | 0m3.760s |
| 100 | 0m0.803s |

*Data Represented in a Graph:* 

*Observations:*

As shown in the graph, it takes less time to complete requests as the number of worker threads increases. This is because there are more workers available to handle requests and return its result to the client, so of course it’s completion time is faster. In our implementation, the dispatcher adds these requests to the queue and the worker threads searches for the requests in cache. Whenever it is a hit, it gives the result of the request to the client. With a miss on the other, it reads from the disk and stores it in cache. With asynchronization on these treads, it gets the request results faster without disrupting the other threads working on other requests.