CSCI 4061
Ziqian Qiu
Guangyu Yan

Performance Analysis Report

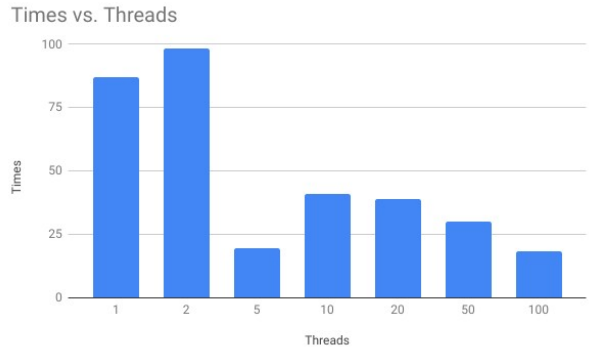
a. I've tried commands requesting files of different content-types with a mixture of HIT and MISS the cache. The results were as the following.

```
qiuxx283@csel-kh4250-42:/home/qiuxx283/Desktop/p3 $ ./web_server 10000 /home/qiuxx283/Desktop/WEB/testing 100 10 0 100 100
[0][1][5][image/jpg/29.jpg][17772][4023 us][MISS]
[1][1][6][image/jpg/29.jpg][17772][1 us][HIT]
[2][1][5][image/gif/29.gif][72507][440496 us][MISS]
[3][1][6][image/gif/30.gif][72507][256994 us][MISS]
[4][1][5][text/gif/30.gif][404 Not Found][100417 us][MISS]
[5][1][7][text/html/30.html][6280][349410 us][MISS]
[6][1][5][text/plain/30.txt][139][346600 us][MISS]
[7][1][6][text/plain/30.txt][139][3 us][HIT]
```

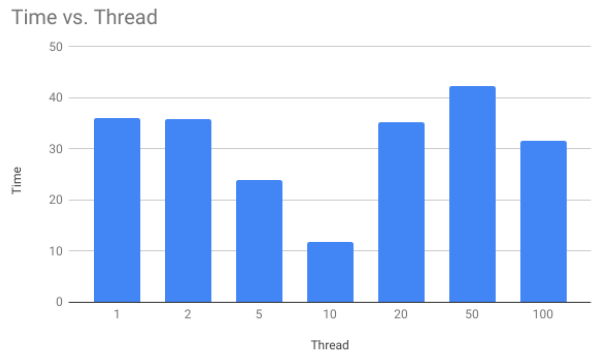
As the figure shows, requests for files of different content-types worked well. If the file didn't exist in the corresponding directory, it would prompt out the error message of "404 Not Found" .

When the request was not in the cache, it showed MISS and if the request was in the cache, it showed the information of HIT. We notice that the service time of getting results from cache is much faster than getting results from disk.

b. The first figure is the running time versus the number of threads for fixed-sized thread-pool.



The second figure is the running time versus the number of threads for dynamic-sized thread-pool.



As seen in the first picture, the running time of one or two thread is much longer than the others. The reason of it is that the workload of the threads are too heavy with few numbers of thread. The threads should finish the current work before moving to the next, which limits the possibility of working concurrently. The running time is the shortest at around 5 threads when the threads are fully used with proper load and are reused efficiently. When the number of threads exceed the number, many threads are in idle. In this condition, the exceeded number of threads will consume extra system resources, which also reduce the performance.

To solve the problem of having too few or too many number of threads. We adapted the dynamic-sized thread-pool. We increase the number of threads by twice if the working threads is more than 80% of the total number of thread which indicates the treads are busy. On the other hand, we reduce the number of thread if the working threads is less than 20% of the total number of thread, which indicates most of the threads are in idle. According to our implementation, when the number of worker threads is set to be 1, it will increase to 8. When the number of worker threads is 2, it increases to 8; 5 to 10; 10 is unchanged; 20 decreases to 10; 50 to 7; 100 to 7. The number of thread will concentrate on the amount around 7 that has fair performance.