

4061 Project 3 Performance Analysis

Kadin Schermers (scher528), Nicole Walker (walk0760), Samuel Ball (ballx188)

HIT/MISS Test:

Below is a series of requests that resulted in a series of HIT or MISSES

Our arguments for this test were as follows:

of dispatchers: 1

of workers: 1

dynamic flag: 0

queue length: 4

cache length: 4

Requests were made one at a time so that dispatcher/worker/queue length amounts would not effect time take and the only measurement being made was against a request already in cache or that had to be retrieved from disk

/text/plain/2.txt	3008	MISS
/text/plain/3.txt	1146	MISS
/text/plain/4.txt	830	MISS
/text/plain/5.txt	765	MISS
/text/plain/6.txt	1128	MISS
/text/plain/7.txt	779	MISS
/text/plain/4.txt	5	HIT
/text/plain/5.txt	5	HIT
/text/plain/8.txt	1082	MISS
/text/plain/8.txt	5	HIT
/text/plain/6.txt	5	HIT
/text/plain/3.txt	408	MISS
/text/plain/6.txt	3	HIT
/image/jpg/5.jpg	6657	MISS

Average HIT time: 5ms Average MISS time: 1756 ms

Conclusion:

The difference here between a HIT and MISS is quite significant. When we already have the content stored by our program, the retrieval time is much faster than having to go find and read the content from the disk. Notice also the much slower time to retrieve a jpg file with a MISS over a txt with a MISS due to the jpg being much larger and taking longer to read in from the disk than the txt file.

Worker Thread Testing:

Our arguments for this test were as follows:

of dispatchers: 100

of workers: variable

dynamic flag: 0

queue length: 100

cache length: 100

Execution times for bigurls file:

-Time for 1 worker thread: 8.032s

-Time for 2 worker threads: 7.144s

-Time for 5 worker threads: 5.683s

-Time for 10 worker threads: 5.104s

-Time for 20 worker threads: 4.639s

-Time for 50 worker threads: 4.844s

-Time for 75 worker threads: 4.647s

-Time for 100 worker threads: 4.729s

Conclusion:

As you can see by the executions times, our program does show improvement when we increase the number of worker threads. We also say that our program had diminishing returns after a certain number of threads was created. This is due to the time that actual thread creation and switching takes. It appears that the optimal number of threads is around 20.

Note: our times do not scale perfectly, that is due to the inherent 'randomness' of things such as CPU timings, system calls, etc. This may be alleviated if many tests were ran for each measurement and all their times were averaged.