Performance Analysis

Kieran Lieberg, Jacob Coenen, Grant Udstrand

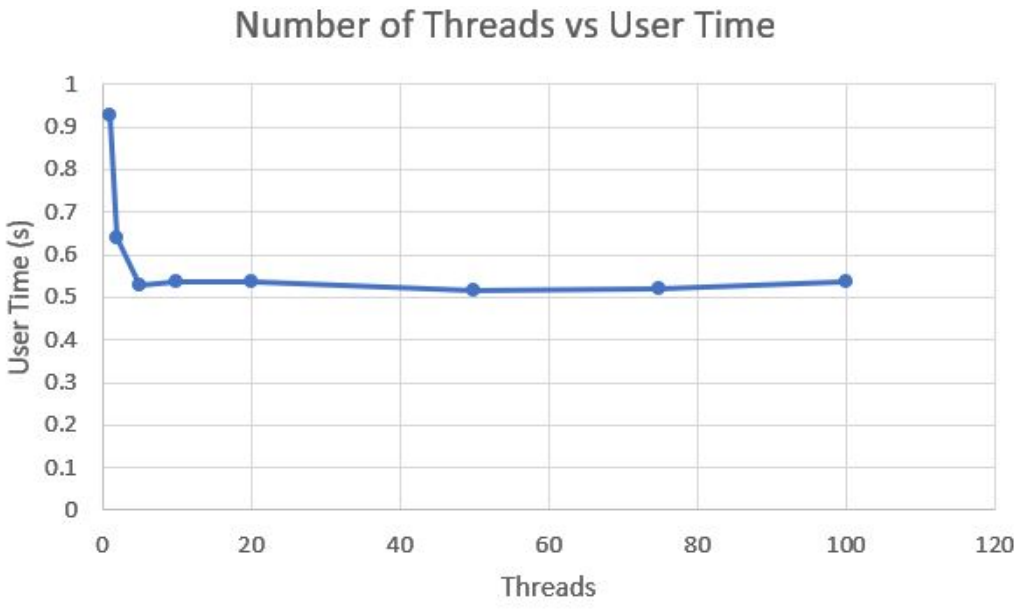
December 12, 2018

1 Cache Efficiency

In a web server, data is constantly being transfered. Reading directly from disc takes a lot of time, so a cache was implemented. The first cache access attempt will always be a MISS, as nothing is in the cache at that moment. Afterwards, repeated calls to retrieve the same piece of data will result in a HIT. Since our cache has no limit to size, there will be very few cache misses, as they only occur upon the first access to a file. However, as the cache fills up, there is the possibility that accessing the cache may become so time consuming that it loses its advantage over directly reading from disc. In our case, the cache usually is around 10 times faster than reading from disc.

2 Graph

When calling xargs with bigurls, the server will get to around 100 out of 140 texts copied, then will exit saying the disc quota has been exceeded. This is predicted to be a problem with vole, or the CSE Labs account that was used to test the server, and not with the code itself. Instead, a new text file called bigjpgs was used, which contained the address for 144 copies of the jpeg file found in the testing folder instead.



3 Analysis

As to be expected, the time taken does decrease as the number of threads goes up. However, it stops after around 10 threads. This is to be expected however, since wget is getting called 8 times simultaniously. It is likely that the worker threads are finishing their data retrieval in time for the next wget to be called, so increasing the thread count past 8 won't have too much of an effect on performance.