Nick Shekelle

Writeup

8 GET requests to a 190 MB file

Single thread time: 0m35.857s

Multithread time: 0m21.265s

Observed speedup with multithreading = 1.69x speedup.

What is likely to be the bottleneck in your system? How much concurrency is available in various parts, such as dispatch, worker, logging? Can you increase concurrency in any of these areas and, if so, how?

Logging is likely the bottleneck in my system. My logging thread could slow down other threads if it is overworked. The dispatch, worker, and logging threads all run concurrently and do not block each other, so I can not increase concurrency between the dispatch, worker, and logging threads.

For this assignment you are logging the entire contents of files. In real life, we would not do that. Why?

There are multiple reasons not to do this in real life. A few off the top of my head are:

1. Performance
2. Security (logging sensitive information)
3. Readability

As I said in the last question, logging impacted my performance. It would impact my performance less if I was not logging the whole file contents.

Another issue is that the log file would hold the file contents of all files, even sensitive ones. An attacker would not need to access the original file to steal sensitive data in this case, they would only need the log file.

Finally, logging the whole file doesn't make the log file very human readable. One of the reasons to have a log file is to hopefully traceback issues, but that does not help much if the log file isn't readable.

Testing:

Nick Shekelle

I tested this program extensively. I unit tested logging, healthcheck, and multithreading separately. I also ran bash scripts with concurrent requests to make sure the whole system was working properly and make sure that my server could handle more requests than there are threads.