WRITEUP: More Improvements to the RPC Server

December 11, 2020

1 Testing

The testing for this program was done mostly by scripts, as the scale we're getting to makes it difficult to test things by running commands. I stress tested all of my server commands by running multiple variations of them in parallel, which exposed multiple flaws in the foundation of my server. Additionally, I tried starting the server with multiple different variations of the opening arguments. With the scale that this program has grown to, I found myself oftentimes creating multiple bugs when trying to solve them - something that, had I had a better design in asgn1/2, could've been easily avoided. Honestly, making this server has been a very educational experience, as far as keeping functions separate and easily testable.

2 Questions

2.1 What did you do to increase concurrency in your server? How does it help improve concurrency?

As the math operations are almost entirely parallelized, my efforts for concurrency went mostly into my load operation, with some effort being put into read/write. In load, I attempted to have everything except inserting values into the variable storage be concurrent (for obvious reasons). However, due to the neccesity to edit values when being loaded, each element must be verified to not be on a list already - which carries an O(n) cost, where n is the size of the list in the respective hash bucket. As more variables are loaded in, the load operation slows down more and more.

These efforts helped, in that I can run multiple load operations simultaneously without crashing, but due to the slowness of my load operation, I'm still failing speed tests. At this point, I think concurrency would be improved most of all by efficiency work.

2.2 How is concurrency affected by recursive name lookup, if at all?

Recursive name lookup slightly hurts concurrency, as the varstore is locked when a value is being recursively searched, and all other threads that want to lookup a variable need to wait. However, the effect is minimal, as recursive lookup is fairly fast.

2.3 What additional tests did you do because performance and scalability matter?

In addition to the scripts mentioned in section 1, I timed everything, and used valgrind excessively. Other than my load operation, everything runs at under 20μs, which I'm vaguely proud of (without any context of what fast should actually look like). I attempted a lot of tests with various algorithmic changes in order to improve speed, but most were not fruitful. However, a lot of these changes exposed flaws in my code, which did help improve scalability significantly - a large change from asgn2 is the server's ability to handle many, many variables.