ECE 459: Programming for Performance
Assignment 2

Daniel Robson

February 18, 2022

1 Message Passing

To begin, the alphabet letters are evenly split among the number of cpus on the system. After that, an unbounded channel is created, which will hold the found secret. These senders and receivers are cloned and passed to the threads. Once a solution has been found, the secret will be sent to the shared channel. Each resulting thread checks the channel each iteration, and will stop and halt when a message exists in the channel.

2 Shared Memory

The shared memory solution is similar to the message passing solution. Several aspects were able to be reused. This solution also begins by dividing the letters between the threads. Instead of a channel, an atomic bool is created and set to false. The threads are spun up, each containing a mutable reference to a clone of the atomic bool. The check_all function is modified to take a mutable reference of the bool. When this bool is set to true, that indicates that a secret is found. Since each iteration this bool is checked, the function cancels and returns.

3 Valgrind Errors

There are none!

4 Helgrind Errors

There are some Helgrind errors, however they are linked to the imported libraries.

5 Benchmark Results

When we run the command to get an output of saaaa. Below are the benchmarks:

Original implementation: 26.75s
Message Passing: 1.76s
Shared Memory: 0.01s
Given a different command where the expected key is 01234:

Original implementation: 40.26s
Message Passing: 3.54s
Shared Memory: 2.26s
Overall, it's much quicker with parallelization!