ENCE360 Assignment

Harrison Camm - 59049620

Introduction

This report concerns the observed performance improvements made from modifications made to a given program. The program, serial.c is a c program that performs the menial task of integrating a given function. Since this task can be considered ‘embarrassingly parallel’, it can be optimized via threading. By utilizing many of the inbuilt synchronization primitives in c, three different programs have been produced. The first program, thread.c, multi-threads the integration. The second, process.c, spawns children to compute the integration such that multiple processes can run at the same time. The third, processThread.c combines these together.

Methodology

To investigate the difference in performance of these different programs, the following steps were taken:

Firstly, in all three programs, the system call clock_gettime() was inserted appropriately into the different programs, both at the beginning and end of each program respectively. Both calls write to two struct timespec: start and finish which are then used to compute the time taken by finding the difference.

All programs were run on the lab computers in Jack Erskine which have the following specifications:

- Operating System: Linux Mint 21.1 Cinnamon (Ubuntu-based)
- Linux Kernal: 5.15.0-91-generic
- Processor: Intel© Core™ i7-10700 CPU @ 2.90GHz × 8
- Graphics Card: Intel Corporation CometLake-S GT2 (UHD Graphics 630)
- Graphics Card: NVIDIA Corporation TU106 (GeForce RTX 2060 SUPER)

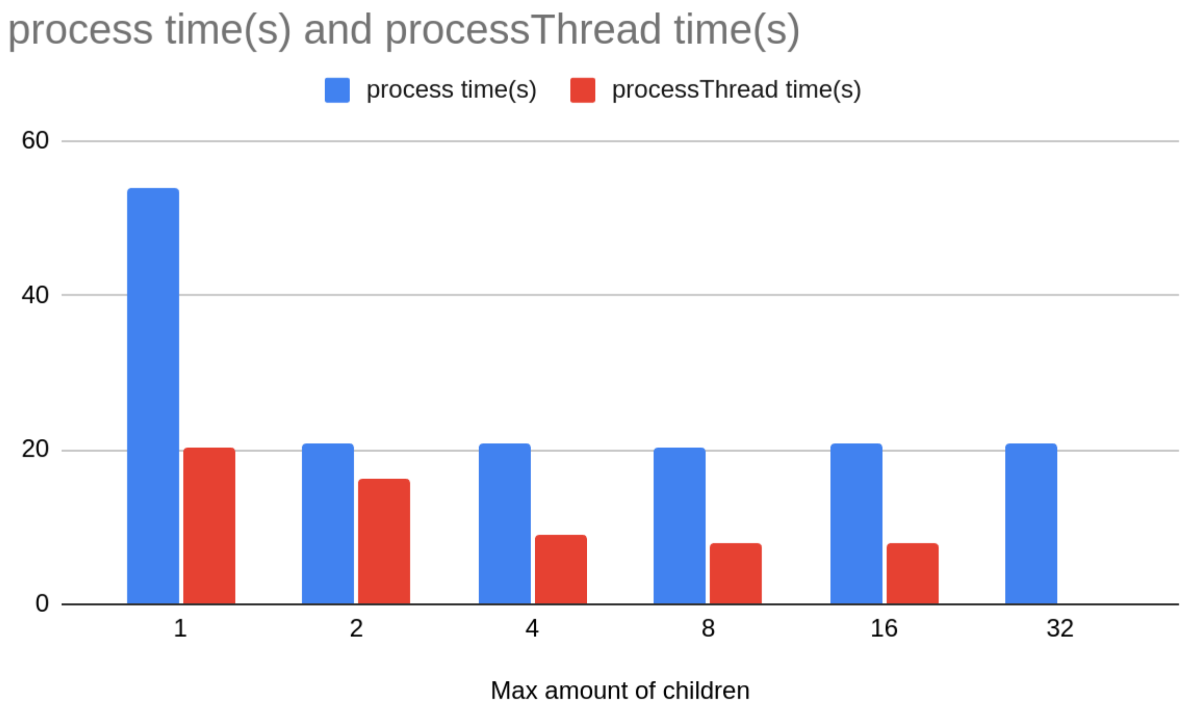
Three different experiments were run to investigate the effect of changing different variables on the performance of these programs.
Firstly, the effect that changing NUM_THREAD, the number of threads has on the performance of thread.c and processThread.c

Secondly, the effect that changing MAX_CHILDREN, the maximum amount of children has on the performance of process.c and processThread.c

Program Descriptions

Results

Maximum Children



The results for the maximum number of threads clearly show that having at least more than one parent available significantly improves the performance of both process and processThread. This is because when a parent process creates a child process, they are concurrently running. When only one child is allowed, then the parent has to wait for that particular child to complete running before it can create a new one [1].

Number Of Threads

References

[1] A. Silberschatz, P. B. Galvin, and G. Gagne, *Operating System Concepts*, 10th ed. Hoboken, NJ, USA: Wiley, 2018.