12/02/2024
Leigh Goetsch
CPE 2600- Systems Programming Dr. Turney

Lab 12: Multithreading

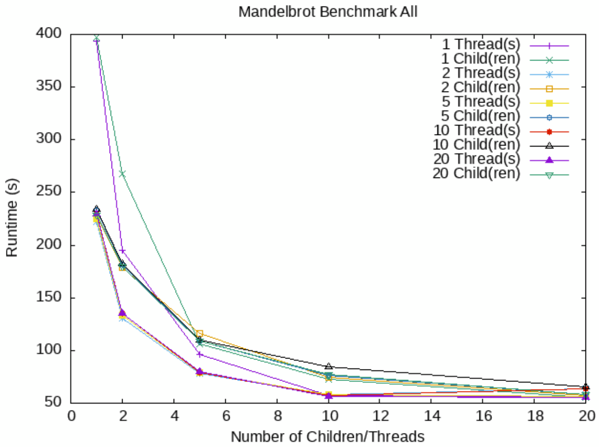
Overview of Threads

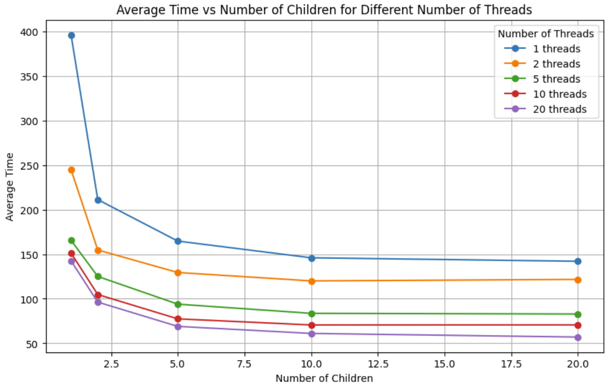
I created a struct to store the arguments for each thread, and created a method for each thread to execute to build the image.

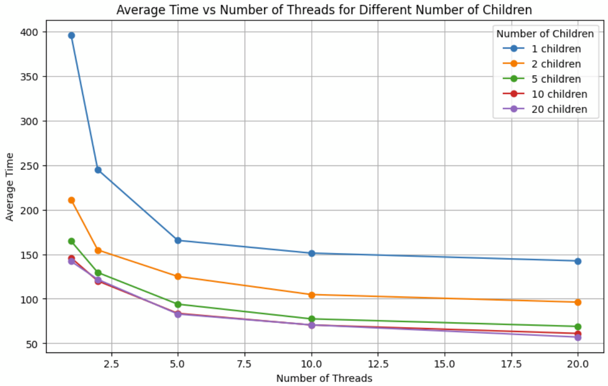
I used the ‘pthread‘ library in C to create and manage threads. The main thread waited for all child threads to complete their execution using the ‘pthread_join‘ function.

Results

num_children	num_threads	time
1	1	394.052345
2	1	194.664908
5	1	96.58268
10	1	57.393608
20	1	55.140141
1	2	222.368145
2	2	130.764116
5	2	77.738222
10	2	57.973349
20	2	63.703118
1	5	224.807174
2	5	133.744711
5	5	79.024139
10	5	57.813655
20	5	56.56309
1	10	230.097656
2	10	135.418242
5	10	79.059961
10	10	57.041502
20	10	63.883061
1	20	229.269893
2	20	135.426675
5	20	79.54736
10	20	56.744055
20	20	55.613426







Discussion

- **Which technique seemed to impact runtime more – multithreading or multiprocessing. Why do you think that is?**

  Multithreading seemed to impact runtime more. This is likely because the overhead of creating and managing threads is less than the overhead of creating and managing processes.

- **Was there a "sweet spot" where optimal (minimal) runtime was achieved?**

  Yes, the optimal runtime was achieved with 20 threads and 10 children.