Operating Systems

Project 1 Report

For part C of the project, the elapsed times of different experiments have been calculated with child processes and threads. The experiments are listed below:

Experiment 1:

First run:
- calculate average request (avg)
- One file
- File size: 100

Second run:
- calculate average request (avg)
- One file
- File size: 100

Third run:
- calculate average request (avg)
- One file
- File size: 1000

Fourth run:
- calculate average request (avg)
- One file
- File size: 10,000

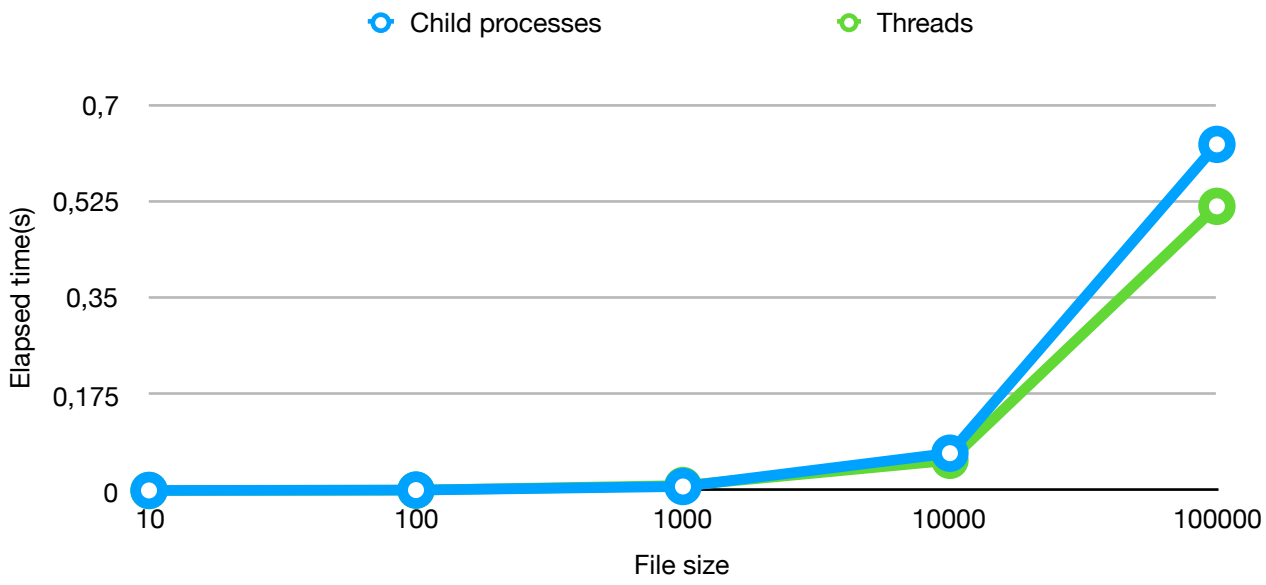
Fifth run:
- calculate average request (avg)
- One file
- File size: 100,000

In this experiment, we aim to observe the difference in elapsed time of child processes and threads as the file size increases and determine which option performs better for smaller file sizes and which performs better for larger file sizes.

	File size	Elapsed time (sec)
Child processes	10	0,000322
	100	0,001009
	1000	0,007481
	10,000	0,068053
	100,000	0,629292

	File size	Elapsed time (sec)
Threads	10	0,000255
	100	0,000907
	1000	0,009250
	10,000	0,055067
	100,000	0,516683

Table 1: Elapsed time of child processes and threads while calculating the average in files with different sizes



Graph 1: Elapsed time with different file sizes

Experiment 2:

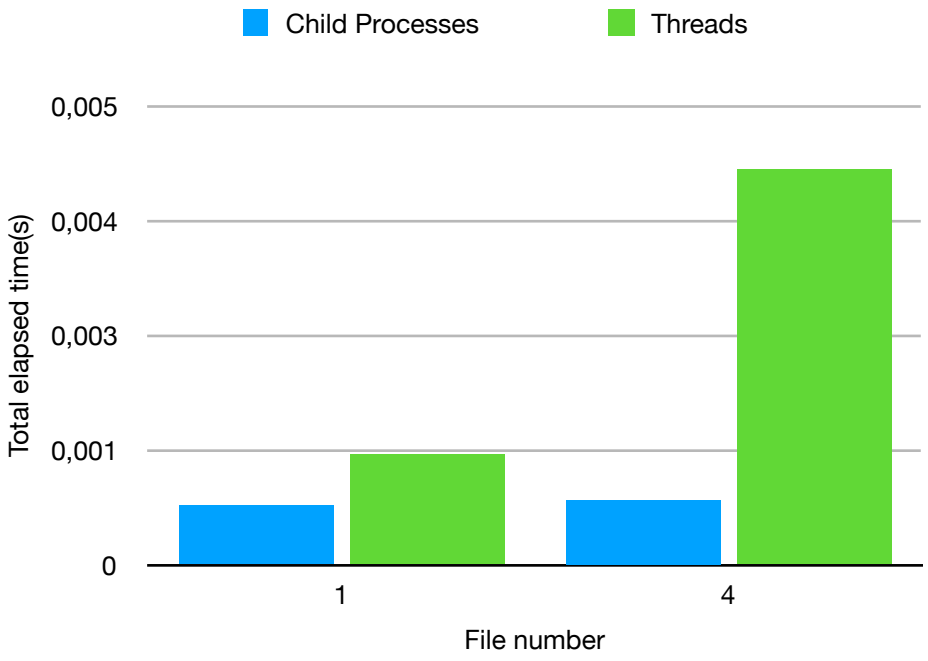
First run:
- calculate max request (max)
- One file
- File sizes: 1000

Second run:
- calculate max request (max)
- Four files
- Each file size 1000

In this experiment, we aim to observe which option performs better as the file number increases. For the sake of making a controlled experiment, the file size is kept constant for this experiment while with each execution the file number increased.

	File number	Elapsed time (sec)	Total elapsed time(sec)
Child processes	1	0,000655	0,000655
	4	0,000648	0,0007145
		0,000697	
		0,000647	
		0,000866	
Threads	1	0,001206	0,001206
	4	0,003660	0,004309
		0,004012	
		0,004522	
		0,005042	

Table 2: Elapsed time of child processes and threads while calculating the max in different numbers of files



Graph 2: Elapsed time with different file numbers

It can be seen from the first experiment that while with small file sizes the elapsed time difference between thread usage and child process usage is extremely small and negligible, the difference increases as the file sizes increase. With larger files, threads seem to perform better according to our experiments. Similarly, with the second experiment, we can observe that threads perform better as the number of files increase. It should be noted that file number is equal to the number of threads/child processes created. It should also be mentioned that the elapsed times in experiments are taken starting from the start of the child process or thread to the moment they terminate.

Therefore, any overhead during initialization of the process/thread are not included in the elapsed time. In addition, the elapsed time actually represents the overall time that thread/process takes from start to finish, including any waiting time.
From the results, we can see that overall threads perform better no matter the file size or number. However, the performance difference becomes more significant as the file size or the file number increases. We can make a few observations to understand why threads are performing better than processes. First of all, each process has its own memory space while threads share a memory space. Therefore, while trying to send back the computations to parent process or main thread, there is a difference in the time the communication takes. A child process can send the data back to the parent by using a pipe, while a thread can simply assign values to a global variable, that is already in the same memory space as the main thread. This difference in communication can create a time difference in the elapsed time. In addition, during context switches, threads take less time than processes as they are already in the same memory space and don't require as much time for copying certain values to the registers. Therefore, the waiting time for a thread can be less than a child process and since as we mentioned the waiting time is also included in elapsed time, the threads can perform faster than processes.