Christian Penick
CPTS360
Dr. Corrigan
04/10/2024

Part 1

Comparing Bash Time and Program Time

When only timing the matrix multiplication, my program's reported real time differed from bash's *time* function by 2.392s (2.263s vs 4.655s). Perhaps this is due to matrix initialization and scheduling of the process start up and close down.

When starting my timers at the beginning of main, I would receive answers from the two timing methods that matched to the hundredths place.

For the rest of part 1, results will be gathered by only timing the matrix multiplication function in main.

Doubling Command Line Argument

Doubling the parameter *n* (number of rows of *c* and *a*) approximately doubled the CPU time (4.457105s vs 2.184308s).

Comparing Times Across Trials

Trial	CPU Time (s)	Real Time (s)
1	4.457105	4.457121
2	4.251207	4.251239
3	4.520551	4.520726
4	4.426102	4.426136

CPU Time Range: 4.520551 - 4.251207 = 0.269344

Real Time Range: 4.520726 - 4.251239 = 0.269487

The differences in Real Time across the trials might come from scheduling causing the CPU to work on other processes differently across trials.

The differences in CPU Time across the trials might come from CPU performance being affected by CPU temperature and how it tried to regulate power consumption.

Christian Penick
CPTS360
Dr. Corrigan
04/10/2024

Part 2

Comparing Bash Time and Program Time

When only timing the matrix multiplication, my program's reported real time differed from bash's *time* function by 2.392s (2.355143s vs 9.009s). This is most likely due to matrix initialization and scheduling of the process start up and close down.

When starting my timers at the beginning of main, I would receive answers from the two timing methods that matched to the tenths place.

For the rest of part 2, results will be gathered by only timing the matrix multiplication function in main.

Trial	CPU Time (s)	Real Time (s)
1	9.376376s	2.344466s
2	9.494640s	2.374265s
3	10.438114s	2.610023s
4	12.791460s	3.198282s

CPU Time Range: 12.791460 - 9.376376 = 3.415084

Real Time Range: 3.198282 - 2.344466 = 0.853816

I think the CPU time might be varying because of other processes using the cache causing the matrix multiplication to have cache misses. It could also be the CPU clock speed varying with CPU temperature and power consumption considerations.

I think the real time might be varying because of other processes being scheduled to run causing the matrix multiplication to be interrupted.

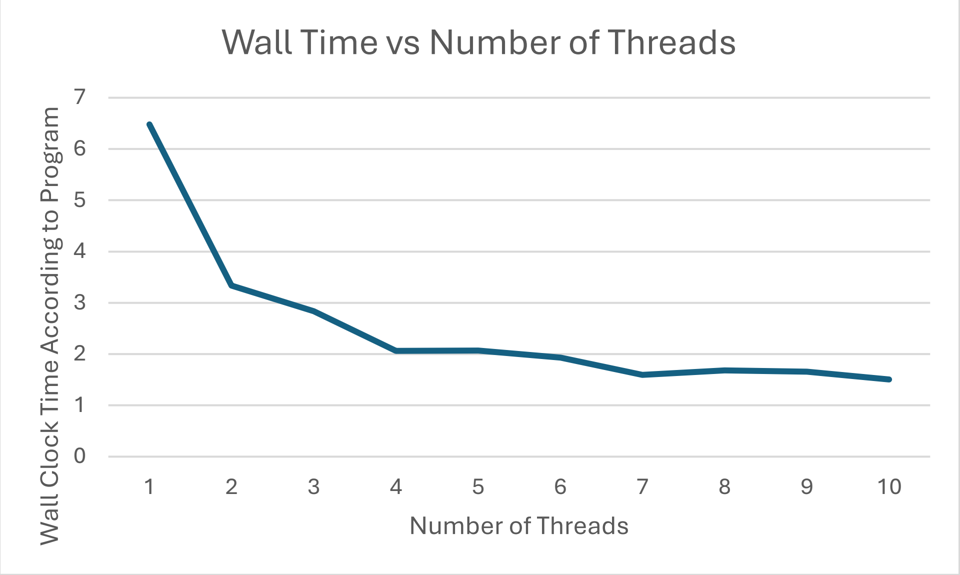
Individual Thread Statistics

All threads got about the same amount of time (within 0.001s of each other).

All threads get roughly the same number of rows to multiply (within 80 rows of each other).

Christian Penick
CPTS360
Dr. Corrigan
04/10/2024

Number of Threads vs Wall Clock Time



Wall time goes down dramatically between 1 and 4 threads with diminishing returns occurring afterwards.

The total CPU time increased by up to a second as I increased the number of threads. Perhaps more cache misses occurred during this portion of the experiment, forcing RAM to be accessed rather than the cache.