Ivan Prskalo
11/30/20
Lab 5 Report
CS351 Systems Programming

FLOPS = (sockets) x (cores per socket) x (cycles per second) x (FLOPS per cycle)

73.68 = 1 x 6 x 3.07 x 4

Mode	Type	Size	Threads	Measured Time	Measured Throughput	Theoretical Throughput	Efficiency
flops	single	small	1	0.58	17.27	73.68	0.234
flops	single	small	2	0.29	34.48	73.68	0.468
flops	single	small	4	0.15	66.20	73.68	0.898
flops	single	medium	1	5.78	17.29	73.68	0.235
flops	single	medium	2	2.94	33.97	73.68	0.461
flops	single	medium	4	1.54	65.14	73.68	0.884
flops	single	large	1	57.86	17.28	73.68	0.235
flops	single	large	2	29.27	34.16	73.68	0.464
flops	single	large	4	15.19	65.84	73.68	0.894
flops	double	small	1	0.58	17.25	73.68	0.234
flops	double	small	2	0.31	32.46	73.68	0.441
flops	double	small	4	0.15	66.32	73.68	0.900
flops	double	medium	1	5.85	17.10	73.68	0.232
flops	double	medium	2	3.01	33.23	73.68	0.451
flops	double	medium	4	1.51	66.34	73.68	0.900
flops	double	large	1	58.47	17.10	73.68	0.232
flops	double	large	2	30.09	33.24	73.68	0.451
flops	double	large	4	19.71	50.73	73.68	0.756

Note: The largest multiple of 1024 that didn’t produce a seg fault for both single and double matrix multiplication on fourier was 6144 so that was used as the size for “large”

Mode	Type	Size	Threads	Measured Time	Measured Throughput	Theoretical Throughput	Efficiency
matrix	single	small	1	0.48	2.09	73.68	0.028
matrix	single	small	2	0.25	4.03	73.68	0.055
matrix	single	small	4	0.12	8.11	73.68	0.110
matrix	single	medium	1	30.41	2.10	73.68	0.029
matrix	single	medium	2	15.86	4.03	73.68	0.055
matrix	single	medium	4	7.89	8.11	73.68	0.110
matrix	single	large	1	106.72	2.02	73.68	0.027
matrix	single	large	2	55.76	3.87	73.68	0.053
matrix	single	large	4	27.57	7.83	73.68	0.106
matrix	double	small	1	0.47	2.12	73.68	0.029
matrix	double	small	2	0.24	4.20	73.68	0.057
matrix	double	small	4	0.12	8.08	73.68	0.110
matrix	double	medium	1	30.15	2.12	73.68	0.029
matrix	double	medium	2	15.35	4.17	73.68	0.057

matrix	double	medium	4	7.92	8.08	73.68	0.110
matrix	double	large	1	105.80	2.04	73.68	0.028
matrix	double	large	2	53.53	4.03	73.68	0.055
matrix	double	large	4	27.70	7.80	73.68	0.106

We can see that for both flops and matrix multiplication the throughputs were mostly unaffected by size. The measured time did increase with size but that is expected as there is significantly more data to be processed per size increase. Threads had the largest impact on throughput in both flops and matrix multiplication. The more threads running in parallel, the higher the throughput. This makes sense since threads have less overhead than processes so running them concurrently resulted in higher throughputs. Matrix multiplication has significantly lower throughput than flops. I used loop unrolling for flops and didn’t use it for matrix multiplication. This is probably the main reason why flops has a significantly larger throughput than matrix multiplication. There doesn’t seem to be much difference between single and double. We do allocate more memory for double than single for matrix multiplication. This caused the max size fourier could handle for matrices to be 6144 instead of 7168.