Miles Van de Wetering

Parallel Project 5

I ran my code on flip2, my data is included below.The speedup was very low when the array size was small, but rapidly caught up, presumable as the hardware advantages overcame the communication overhead. The speedup spiked again as the array sizes became very large. Interestingly, my speedup in both cases was greater than the expected value of 4. This is most likely because the provided assembly code was simply much more efficient than my own function calls. Interesting the speedups did not remain consistent, as one might expect. I'm not entirely sure why, since I thought that the speedup efficiency would be relatively flat after reaching a reasonably large array size, say 2k. They did, however, show consistent patterns across the sum and the sum/mult operations, and over multiple runs.

Array Multiplicaton & Summation

Array Size	SIMD	CPP	Speedup
1024	1.08E+08	59886254	1.801059
2048	5.17E+08	89961678	5.746121
4096	4.91E+08	88552461	5.544937
8192	4.66E+08	86333543	5.40017
16384	4.52E+08	85161254	5.305229
32768	4.27E+08	81415529	5.239231
65536	4.24E+08	83331254	5.08963
131072	4.31E+08	82059278	5.247192
262144	4.33E+08	81964318	5.280242
524288	4.26E+08	82232066	5.178677
1048576	5.76E+08	86841489	6.633363
2097152	7.52E+08	89387996	8.409945
4194304	8.59E+08	90837967	9.45453
8388608	9.25E+08	91505475	10.11226
16777216	9.62E+08	91819773	10.47531
33554432	9.83E+08	92042652	10.67614

Array Multiplication

Array Size	SIMD	CPP	
1024	1.08E+08	67900428	1.593799
2048	3.36E+08	1.19E+08	2.814973
4096	2.83E+08	1.2E+08	2.351172
8192	2.48E+08	1.21E+08	2.048628
16384	2.24E+08	1.21E+08	1.853314
32768	2.24E+08	1.21E+08	1.848934
65536	2.18E+08	1.21E+08	1.79949
131072	2.17E+08	1.16E+08	1.871552
262144	2.17E+08	1.2E+08	1.814925
524288	2.15E+08	1.2E+08	1.795216

1048576	3.09E+08	1.19E+08	2.587968
2097152	3.98E+08	1.19E+08	3.330891
4194304	4.57E+08	1.19E+08	3.843272
8388608	4.99E+08	1.18E+08	4.209196
16777216	5.2E+08	1.19E+08	4.37609
33554432	5.31E+08	1.3E+08	4.095221