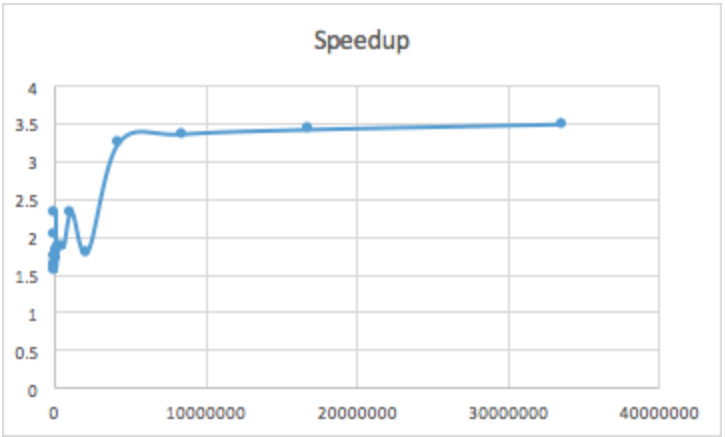
SR Kanna
Parallel HW5

Multiplication

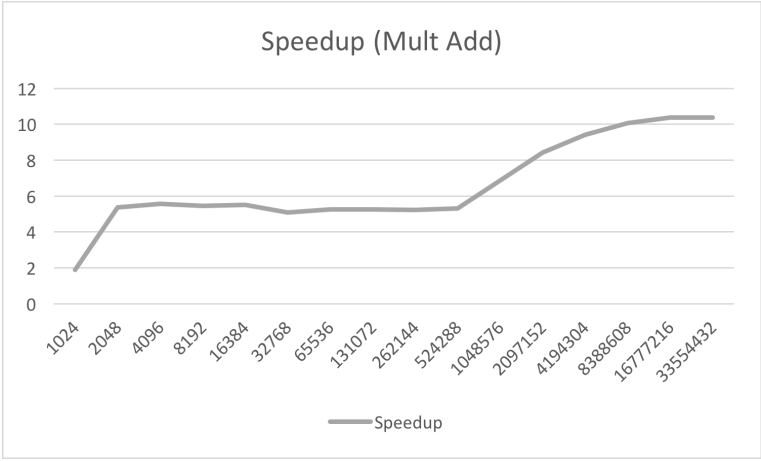
Array Size	SIMD	CPP	Speedup
1024	191444152.313760	124119391.293786	1.542419
2048	507681693.536188	219129892.683491	2.316807
4096	449376367.743333	221787519.470701	2.026157
8192	385054687.702676	221579404.674329	1.737773
16384	352125421.225300	215253522.144143	1.635864
32768	366319762.086576	226456471.205415	1.617617
65536	392529392.411803	232031787.232794	1.691705
131072	421533480.013053	231907812.536457	1.817677
262144	425483797.241072	228095498.024271	1.865376
524288	424482811.382664	227719232.991769	1.864062
1048576	535363094.675029	230544580.856623	2.322167
2097152	644353830.829563	230376391.572707	2.796961
4194304	740473499.986825	228402437.123103	3.241968
8388608	763424887.532274	227701692.221746	3.352741
16777216	783737630.314495	229570565.788152	3.413929
33554432	799614268.532508	229321973.717185	3.486863



Array Multiplication & 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	91	10.47531
33554432	9.83E+08	92042652	10.67614

Chart Area



I ran this on flip. The speedup increased in correlation with the arraysize. It was close to 1.5 when the array size was 1024, 1.9 near arraysize 262144 and 3.5 for array size 33554432. Overall the speedup increases (quite rapidly around arraysize 2048), before dropping down and rising at a more steady rate. We can see that there was a drop to 1.8 around array size 8192, before beginning to increase to 3.48 again. This increase probably occurred because eventually the hardware overtook the communication overhead. On the other hand, for the multiplication and summation there is also an increase, but there is minimum fluctuation in values. It jumps from a speedup of one to five, before steadily increasing to ten by an arraysize of 33554432.

This time the speedup was a lot larger than the approximated value, most likely because the assembly code optimized better the code. I'm not entirely sure why the speedup didn't remain consistent across the array size for either the multiplication or add/multiplication. It seems logical that it would stagnate after a certain array size, or maybe I didn't test for a large enough size.