1 Results: Graphs

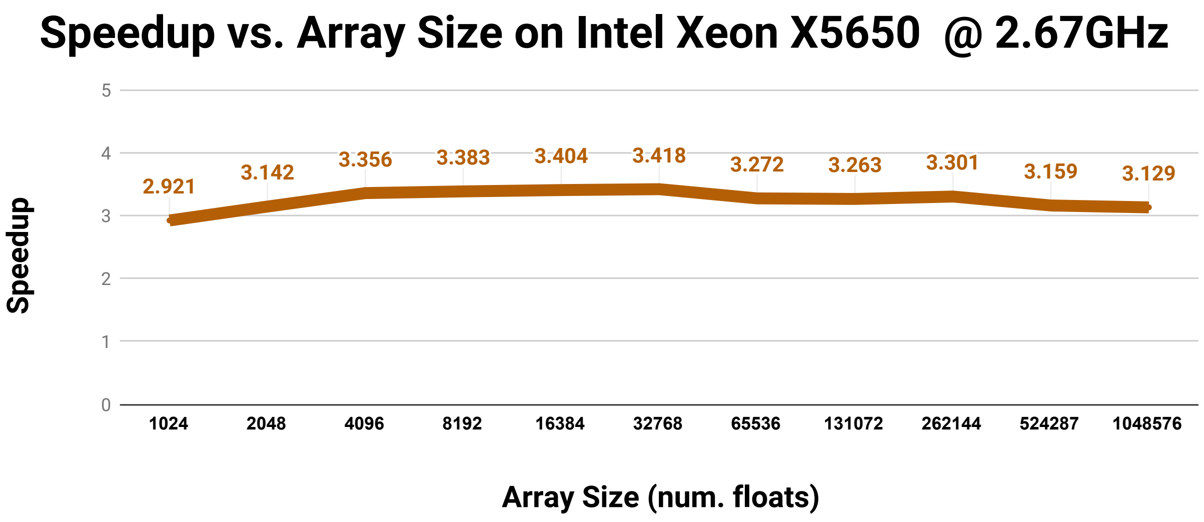

Figure 1: A sample run on the Flip2 server.

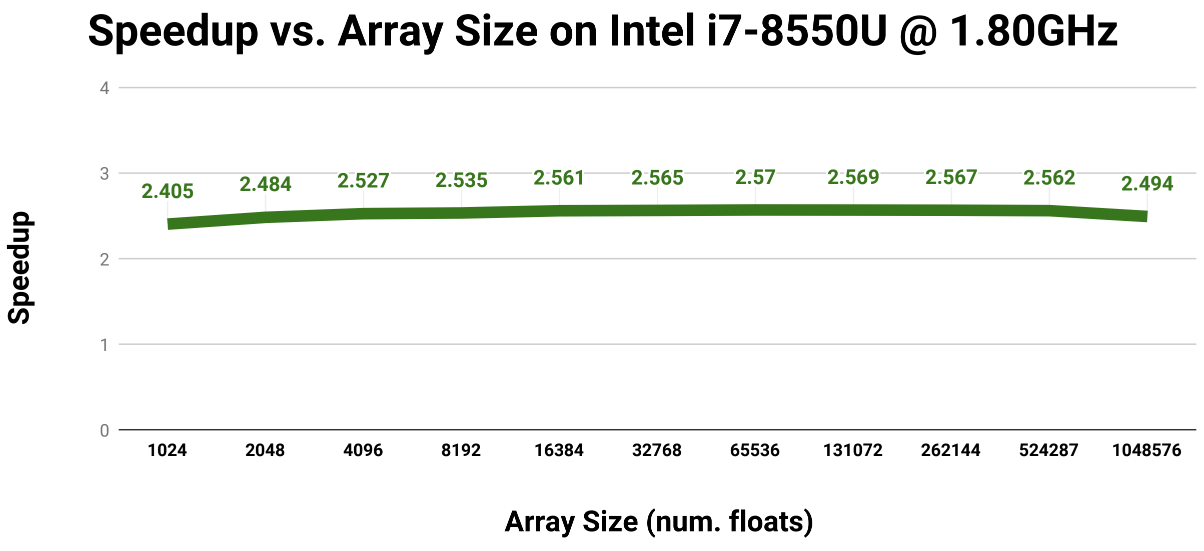

Figure 2: A sample run on my laptop.

2 Results: Tables

Array Size	Speedup
1024	2.921
2048	3.142
4096	3.356
8192	3.383
16384	3.404
32768	3.418
65536	3.272
131072	3.263
262144	3.301
524287	3.159
1048576	3.129

Table 1: Table of the Array Size/Speedup relationship when testing on Flip2.

Array Size	Speedup
1024	2.405
2048	2.484
4096	2.527
8192	2.535
16384	2.561
32768	2.565
65536	2.570
131072	2.569
262144	2.567
524287	2.562
1048576	2.494

Table 2: Table of the Array Size/Speedup relationship when testing on my laptop.

3 Discussion

I ran this project on OSU's `flip2` server as well as my laptop. The `flip2` run yielded the best results (speedup) for this project. The title of the graph indicates what CPU the computer has, and can be seen in Fig. 1 and Fig. 1. Performance wise, SSE has about the same speedup per device across the board. We see a small dip in the beginning and ends of both graphs. A dip in the beginning can probably be attributed the fact that the array size is so small that the performance boost of SSE doesn't have room to shine yet. A dip in the end? It's probably not a prefetching issue, because we would have seen that from the start. It might be that the repeated trips to cache (cache line is 16 floats) are starting to add up over time, but you would see that on the non-SIMD side of things, too. I don't have a definitive answer for why we would see performance trend downwards at the upper ends of the tests.

Why is the `flip2` about 1.5x faster than my laptop? The SIMD features of the Intel Xeon X5650 and Intel i7-8550U are the same:

```
MMX instructions
SSE / Streaming SIMD Extensions
SSE2 / Streaming SIMD Extensions 2
SSE3 / Streaming SIMD Extensions 3
SSSE3 / Supplemental Streaming SIMD Extensions 3
SSE4 / SSE4.1 + SSE4.2 / Streaming SIMD Extensions 4
```

But you do have the Xeon running faster as well as...

i7 Cache

```
Level 1 cache size:     4 x 32 KB 8—way set associative instruction caches
                        4 x 32 KB 8—way set associative data caches
Level 2 cache size:     4 x 256 KB 4—way set associative caches
Level 3 cache size:     8 MB 16—way set associative shared cache
```

Xeon Cache

```
Level 1 cache size:     6 x 32 KB 4—way set associative instruction caches
                        6 x 32 KB 8—way set associative data caches
Level 2 cache size:     6 x 256 KB 8—way set associative caches
Level 3 cache size:     12 MB 16—way set associative shared cache
```

1.5x as many L1 and L2 caches vs my laptop, plus a 1.5x bigger L3 cache. I actually estimated the speedup laptop vs `flip2` ratio before finding out how much bigger the cache was so you can imagine how nice the 1.5x lined up!

As a side note, the company I'm working for is trying to get a lot of parallel Kalman filters on an ARM Cortex-A9 working, and working fast. I looked up if there was an equivalent to Intel's Intrinsics and it looks like there is! ARM SIMD ISAs