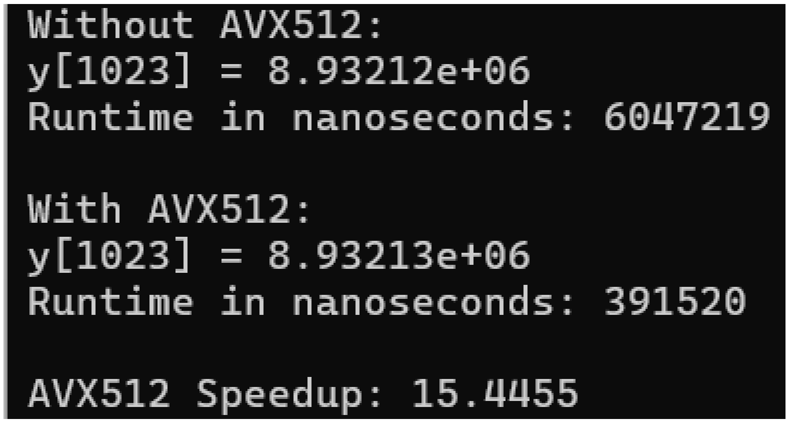
Justin Bahr
NUID: 001592707

High Performance Computing
Homework 3 - Question 2

I ran my program on the Explorer Cluster with an AVX512 node (--constraint=cascadelake) (Intel ® Xeon ® Platium 8276 CPU @ 2.20GHz, 28 cores per socket, 2 sockets, with 196090404 KB of memory, and Linux 5.14.0).

a. I noticed a wide range of speedup times between runs ranging from 6x to 15x. The maximum speedup time I observed when adding AVX512 support compared to a non-vectorized approach was 15.44 (seen in figure 1 below). The theoretical maximum speedup would be about 16x, since the 512 bit SIMD bus can fit 16 floats (512 bit vector / 32 bits per float = 16 floats per vector). Since no random data was used and my script file requests exclusive use of the node, I predict that the variation in run time is caused by shared memory latency variations due other processes on the socket.



```
Without AVX512:
y[1023] = 8.93212e+06
Runtime in nanoseconds: 6047219

With AVX512:
y[1023] = 8.93213e+06
Runtime in nanoseconds: 391520

AVX512 Speedup: 15.4455
```

Figure 1. Maximum Speedup Obtained using AVX512

b. I found the following three AVX assembly instructions:

vmulss %xmm0, %xmm2, %xmm0

- This is a vectorized single-precision scalar floating point multiplication instruction

vaddss %xmm0, %xmm1, %xmm0

- This is a vectorized single-precision floating point addition instruction

vmovss (%rdx,%rax,4), %xmm1

- This instruction loads a single-precision floating point number to an xmm register

The xmm registers are 128-bit registers