Homework 2

Josh Bradt

February 1, 2016

1 Array copy performance

I ran this test on my Late-2012 Mac Mini with quad-core Intel Core i7-3615QM processor. This processor has 64 kB L1 cache per core, 256 kB L2 cache per core, and 6 MB L3 cache which is shared between the cores. The computer has 16 GB of RAM installed. The operating system is Mac OS X El Capitan (10.11.2).

All codes were compiled using the default Apple-provided Clang version 7.0.2 with optimization level -O2.

The results for this section are shown in Table 1. Note that I was unable to run the test for an array size of 5 000 000 since OS X sets a hard limit of 65 532 kB on the stack size which cannot be bypassed with `ulimit -s`.

The time results are plotted in Fig. 1, and the data rates are plotted in Fig. 2.

2 STREAM benchmark

I ran the C version of the STREAM benchmark on the same system as above, compiled as above with `clang -O2`. The array size was set to 3 000 000, which produced an array size of 22.9 MiB. This is more than 4 times the size of the 6 MB L3 cache on this system.

N	Time for Loop (s)	Rate (MB/s)
250	4.76×10^{-8}	8.40×10^{4}
1000	1.80×10^{-7}	8.89×10^{4}
5000	1.85×10^{-6}	4.33×10^{4}
10 000	3.63×10^{-6}	4.41×10^{4}
50 000	2.28×10^{-5}	3.51×10^{4}
100 000	4.52×10^{-5}	3.54×10^{4}
500 000	3.68×10^{-4}	2.17×10^{4}
1 000 000	8.78×10^{-4}	1.82×10^{4}
5 000 000	(Failed)	(Failed)

Table 1: Results from Part 1

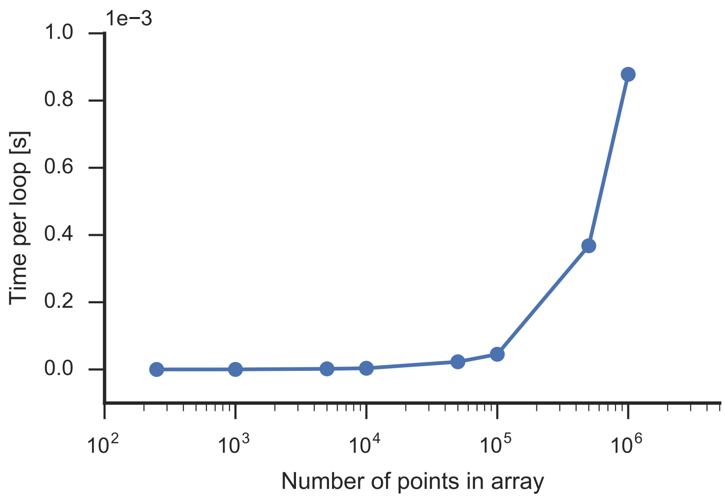

Figure 1: Time per loop for Part 1.

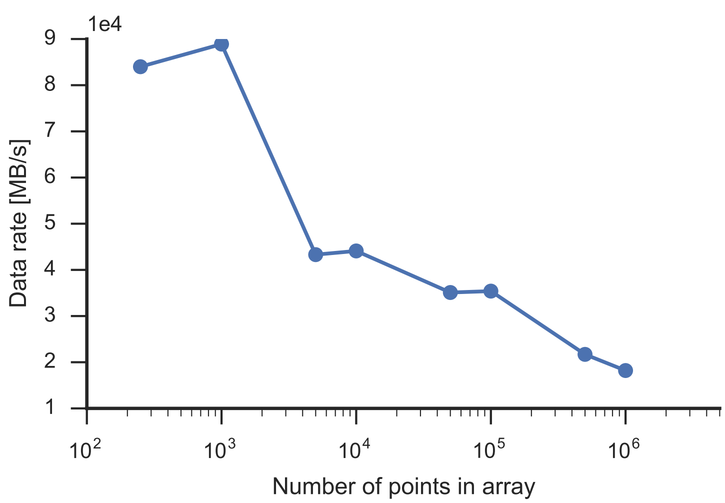

Figure 2: Data rate for Part 1.

Function	Best rate [GB/s]	Avg time [ms]	Min time [ms]	Max time [ms]
Copy	16.5	3.14	2.89	4.12
Scale	12.3	4.25	3.92	5.28
Add	12.7	5.99	5.67	6.41
Triad	12.7	6.09	5.66	6.62

Table 2: STREAM results

The results of the benchmark are shown in Table 2.

3 Sparse matrix-vector multiply performance estimate

If we ignore the effects of the cache, then the estimate given on slide 8 in lecture 5 is:

$$\text{time} = n_{NZ}(2c + 2.5r) + n(0.5r + w) \tag{1}$$

where n is the number of rows, n_{nz} is the number of nonzero elements, c is the time per operation, r is the time per read, and w is the time per write.

The STREAM copy operation performs one read and one write, so take

$$r = w = \frac{t_{copy}}{2N} = \frac{3.14\,\text{ms}}{2 \cdot 3\,000\,000} = 0.523\,\text{ns}.$$

If the processor can perform one floating point operation per clock cycle, then the processor clock speed suggests that $c = 1/(2.3 \times 10^9\,\text{Hz}) = 0.435\,\text{ns}$. Thus, the time should be

$$\text{time} = n_{NZ}(2.18\,\text{ns}) + n(0.785\,\text{ns}). \tag{2}$$

For an example, assume we have a $1\,000\,000 \times 1\,000\,000$ tridiagonal matrix. This has approximately $3\,000\,000$ nonzero elements. Therefore, the time for the sparse matrix-vector multiply should be approximately

$$\text{time} = 3\,000\,000(2.18\,\text{ns}) + 1\,000\,000(0.785\,\text{ns}) = 7.33\,\text{ms}.$$