Josh Huff, huffj@oregonstate.edu
Professor Bailey
CS 475 – Spring 2017
June 13, 2017

Project 7B: Commentary

---

1. What machine you ran this on

OpenMP and SIMD:

I knew from experience that SIMD caused strange errors and segfaults on my personal machine, so I ran both iterations of the OpenMP implementation and the SIMD implementation all on flip2 for the sake of consistency and meaningful comparison.

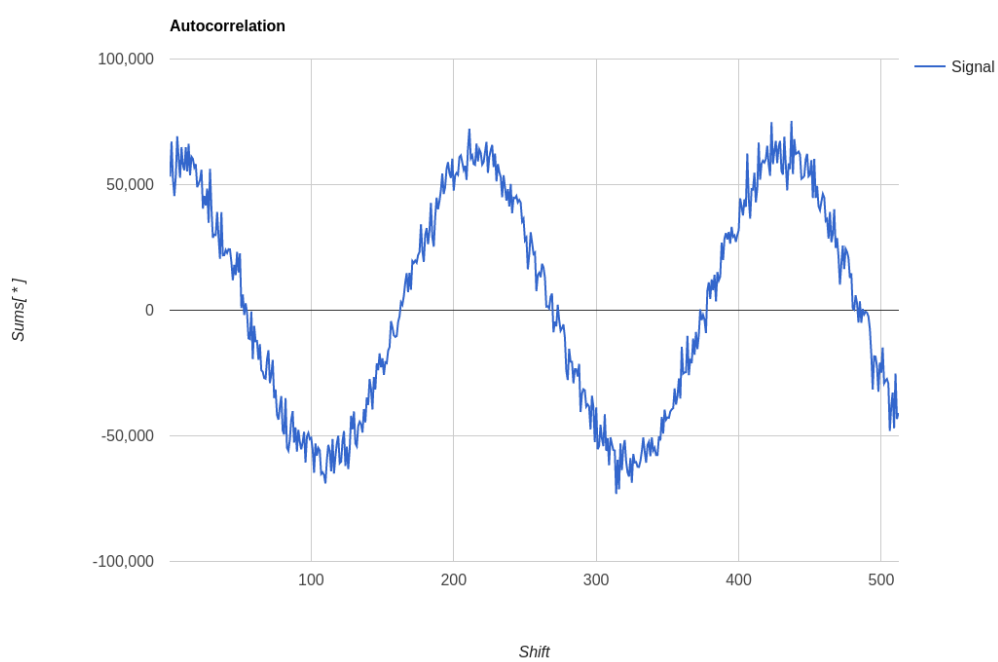
Load average for OpenMP runs: 3.28, 3.65, 3.63
Load average for SIMD runs: 3.44, 3.37, 4.03

OpenCL:

My main workhorse does not have a GPU, and neither does flip. Therefore, my only option was rabbit.

The load averages were: 0.49, 0.22, 0.13 – like having the Autobahn to myself.

2. Show the Sums[1] … Sums[512] vs. shift scatterplot

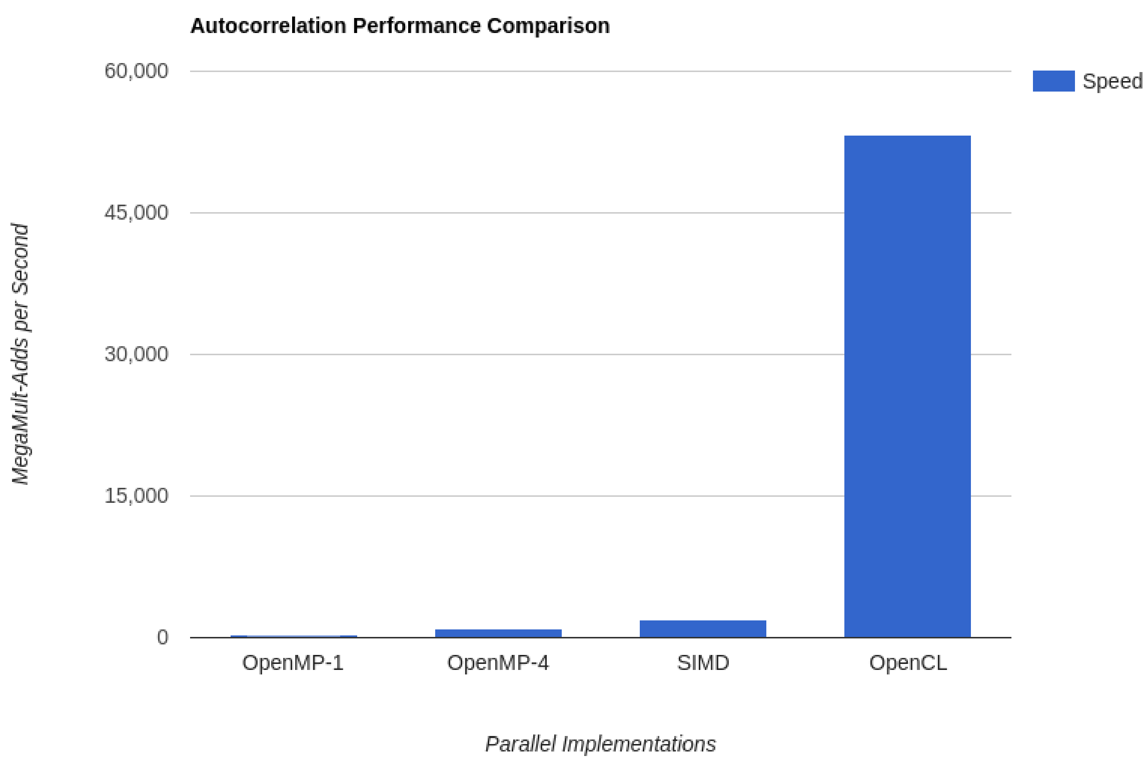


3. State what the hidden sine-wave period is, i.e. at what multiples of shift are you seeing maxima in the graph?

The graph seems to indicate that the sine-wave reaches a maxima with a very small shift, then again 200 shifts later with the same shift plus a little extra distance from the verticle gridline, then a third time plus 400. By outputting the program results to a text file and opening in vim using shift+G with number lines set, I was able to see higher products occurring around multiples of 210-211 (with some variation).

Visual inspection of the output file showed actual maxima at 211, 423, 650, 858, and so on. Multiples of 211 coincide conveniently at 422, 633, 844, and so on.

4. What patterns are you seeing in the performance bar chart? Which of the four tests runs fastest, next fastest, etc. By a little, or by a lot?



The OpenMP-1 (that is, the serialized run) is barely visible.
The OpenMP-4 isn't much better.
The SIMD is substantially larger, but still looks tiny on the graph.
The OpenCL is Brobdingnagian. It's so huge, I had to pull out a two-dollar word for it.

5. Why do you think the performances work this way?

As always, the OpenMP result is a given. It provides a baseline with which to compare actual parallel performance.

The OpenMP with four threads result demonstrated that the problem is embarrassingly parallel, because it has a speedup of almost exactly 4. This was a harbinger of the sorts of results the SIMD and OpenCL implementations would produce.

The SIMD implementation enjoys the benefit of assembly language and an FMA without the burden of an array size that would demonstrate a hardware limitation as in Project 5. As such, the speedup is better than four as in OpenMP, but it still doesn't hold a candle to OpenCL.

The OpenCL implementation shows what GPU raw computing power has. I was admittedly lazy about toying with workgroup sizes and have likely not seen the full potential – but the fireworks show was plenty to prove the point. It's almost like comparing humans to primates. We have better fine motor skills and fair strength (like the CPU's logic and control abilities and fair processing power), while they have poor motor skills and fantastic strength. Perhaps rabbit should be nicknamed "silverback"?