

# August 27th, 2015 Pre-Class Questions

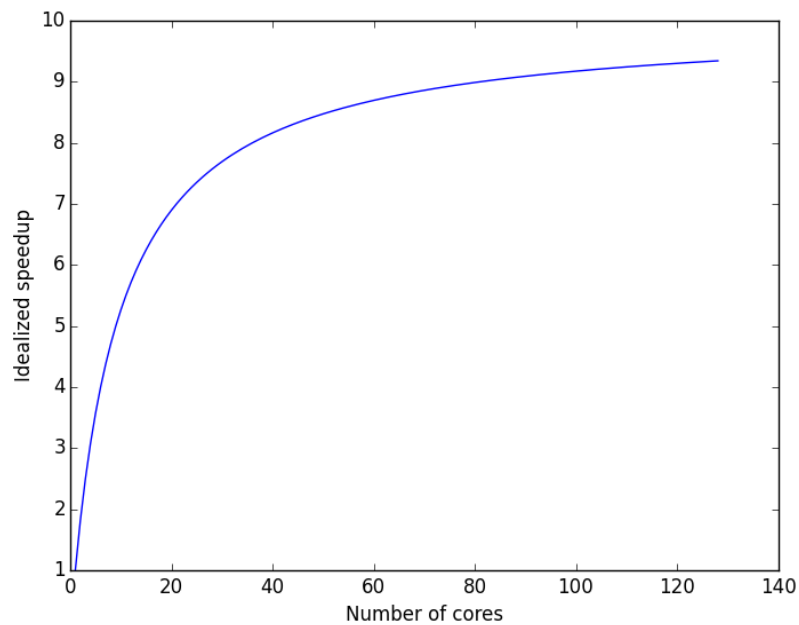
Elliot Cartee

August 27, 2015

## Question 1

This plot can be found in Figure 1.

Figure 1: Idealized speedup versus number of cores



A script to make the plot using `matplotlib` is in `parallel_plot.py` in this directory. It follows here:

```

import matplotlib.pyplot as plt

max_cores = 128
serial_fraction = 0.1

cores = range(1,max_cores+1)
speedup = range(0,max_cores) #temporary
for i in cores:
    speedup[i-1] = 1.0/(serial_fraction + (1.0-serial_fraction)/i)

plt.plot(cores,speedup)
plt.xlabel('Number of cores')
plt.ylabel('Idealized speedup')
plt.show()

```

## Question 2

We note that the serial time required to compute  $k$  tasks is  $k * \alpha + k * \tau$ . The parallelizable part is  $k * \tau$ . As the number of processors increases, the wall clock time required to complete the parallelizable part goes to 0, and thus the throughput is bounded by  $k * \tau / k = \tau$

## Question 3

It is best not to tune when the human time required to tune is more valuable than the computer time that is saved by tuning. This can be the case because implementing and maintaining better performance can become very difficult, or because a particular part of the code does not consume much of the total time

## Question 4

Each of the Xeon Phi 5110P boards has a frequency of 1.053 GHz with 60 cores

Source:

[http://ark.intel.com/products/71992/Intel-Xeon-Phi-Coprocessor-5110P-8GB-1\\_053-GHz-60-core](http://ark.intel.com/products/71992/Intel-Xeon-Phi-Coprocessor-5110P-8GB-1_053-GHz-60-core)

Each core can compute 16 double precision floating point operations per cycle

Source:

<https://software.intel.com/en-us/articles/intel-xeon-phi-core-micro-architecture>

Each of the 8 nodes has a Intel Xeon E5-2620 v3 processor, which has 12 cores at a base frequency of 2.4 GHz

Source:

[http://ark.intel.com/products/83352/Intel-Xeon-Processor-E5-2620-v3-15M-Cache-2\\_40-GHz](http://ark.intel.com/products/83352/Intel-Xeon-Processor-E5-2620-v3-15M-Cache-2_40-GHz)

The above source also says it uses the AVX 2.0 instruction set. According to the source below, this can compute 16 double precision floating point operations per clock.

Source:

<http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/performance-xeon-e5-v3-advanced-vector-extensions-paper.pdf>

(Note that I had to split the URL into two lines because it is too long for the page)

This means that the theoretical peak flop rate from the accelerators is:

$$\begin{aligned} & (\# \text{ of cores}) * (\text{cycles per second}) * (\text{flops per cycle}) \\ &= (15 * 60) * (1.053 \text{ GHz}) * (16) = 15163.2 \text{ GFlop/sec} = 15.2 \text{ TFlop/sec} \end{aligned}$$

And the theoretical peak flop rate from the nodes is:

$$\begin{aligned} & (\# \text{ of cores}) * (\text{cycles per second}) * (\text{flops per cycle}) \\ &= (8 * 12) * (2.4 \text{ GHz}) * (16) = 3686.4 \text{ GFlop/sec} = 3.7 \text{ TFlop/sec} \end{aligned}$$

Together this comes out to a theoretical peak flop rate of:

$$15.2 \text{ TFlop/sec} + 3.7 \text{ TFlop/sec} = 18.9 \text{ TFlop/sec}$$

## Question 5

My machine is a mid-2012 13-inch Macbook Air. It has a 1.8 GHz Intel Core i5 processor with two cores Source:

[http://ark.intel.com/products/64903/Intel-Core-i5-3427U-Processor-3M-Cache-up-to-2\\_80-GHz](http://ark.intel.com/products/64903/Intel-Core-i5-3427U-Processor-3M-Cache-up-to-2_80-GHz)

Since this CPU uses the AVX instruction set, it can execute 8 double precision flops per cycle, So the theoretical peak flop rate is:

$$\begin{aligned} & (\# \text{ of cores}) * (\text{cycles per second}) * (\text{flops per cycle}) \\ &= (2) * (1.8 \text{ GHz}) * (16) = 57.6 \text{ GFlop/sec} \end{aligned}$$