CS 5220 Applications of Parallel Programming	Fall 2015
Kenneth Lim (kl545)	Performance Analysis Basics

1. Given $\alpha = 0.1$, we have:

$$S(p) \leq \frac{1}{0.1 + 0.9/p}, \quad 1 < p < 128$$

2. The total time T taken to complete k tasks is:

$$T = k(\tau + \alpha)$$

   Since $k\alpha$ represents the serial portion of the task which does not benefit from parallelization, the throughput becomes $k/(k\alpha) = 1/\alpha$.

3. Tuning should be avoided when: (a) the man hour cost of tuning exceeds the computation time saved; (b) it potentially converts the codebase to spaghetti *al dente*; (c) and when it does not target or significantly affect the code bottleneck.

4. The peak double precision FLOPS for an Intel Xeon Phi coprocessor 5110P is 1010.88 GF/s[1]. where the performance is obtained by:

$$16 \, \frac{\text{FLOPS}}{\text{clock}} \times 60 \, \text{cores} \times 1.053 \, \text{GHz} = 1010.88 \, \text{GF/s}$$

   The peak double precision FLOPS for a single Intel E5-2620 v3 with Turbo Boost is 120 GF/s[2]. This came straight from Intel's data sheet without specifying how it is calculated, though it *probably* uses only AVX2 instructions.

   Since we have a total of 96 cores and 15 accelerators, the overall peak FLOPS for the system is:

$$96(120) + 15(1010.88) = 26683.2 \, \text{GF/s} = 26.683 \, \text{TF/s}$$

5. My machine is a Late 2013 15" Apple Retina Macbook Pro. The processor is an Intel Core i7-4850HQ with a maximum clock speed of 3.5 GHz, 4 cores, 16 vector instructions per cycle, and a 64-bit wide AVX instruction set (which works out to 1 double-precision operand). This gives:

$$3.5 \times 4 \times 16 \times 1 = 224 \, \text{GF/s}$$

[1] https://www-ssl.intel.com/content/www/us/en/benchmarks/server/xeon-phi/xeon-phi-theoretical-maximums.html

[2] http://download.intel.com/support/processors/xeon/sb/xeon_E5-2600.pdf

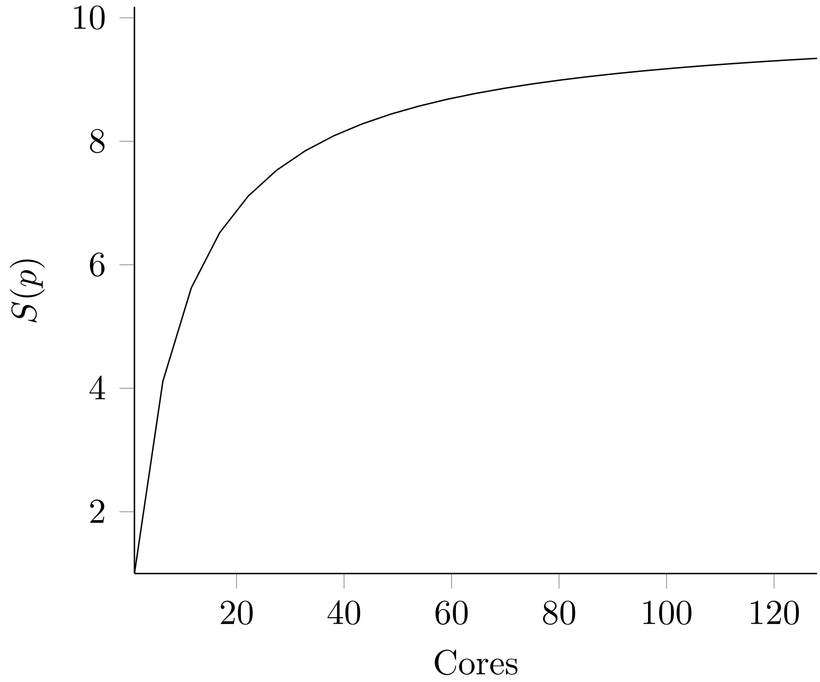

Figure 1: Plot of idealized speedup vs. number of cores used for $p = 1$ to $p = 128$ and $\alpha = 0.1$. The graph asymtotes at $S(p) = 1/\alpha = 10$.

```
\begin{tikzpicture}
  \begin{axis}[
    enlarge x limits=false,enlarge y limits=upper,
    xlabel=Cores,ylabel=$S(p)$,tick align=outside,
    axis lines*=left,axis on top]
    \newcommand{\ALPHA}{0.1}
    \addplot[black,domain=1:128]
    {1/(\ALPHA + (1-\ALPHA)/x)};
  \end{axis}
\end{tikzpicture}
```

Figure 2: Graph Generation Code, using LaTeX's TikZ and PGFPlots packages

In addition, the machine also has an integrated GPU as well as a dedicated GPU, both of which we assume can serve as accelerators. The former is an Intel Iris Pro 5200 with peak FLOPS of 832 GF/s[3], and the latter is an NVIDIA GeForce GT 750M, with peak FLOPS of 722.7 GF/s[4]. Thus the total theoretical peak flop rate for my machine is:

$$224 + 832 + 722.7 = 1778.7\,\mathrm{GF/s} = 1.78\,\mathrm{TF/s}$$

[3] http://www.anandtech.com/show/6993/intel-iris-pro-5200-graphics-review-core-i74950hq-tested/2
[4] https://www.techpowerup.com/gpudb/2224/geforce-gt-750m.html