CS 5220 – Aug. 27 Preclass Questions

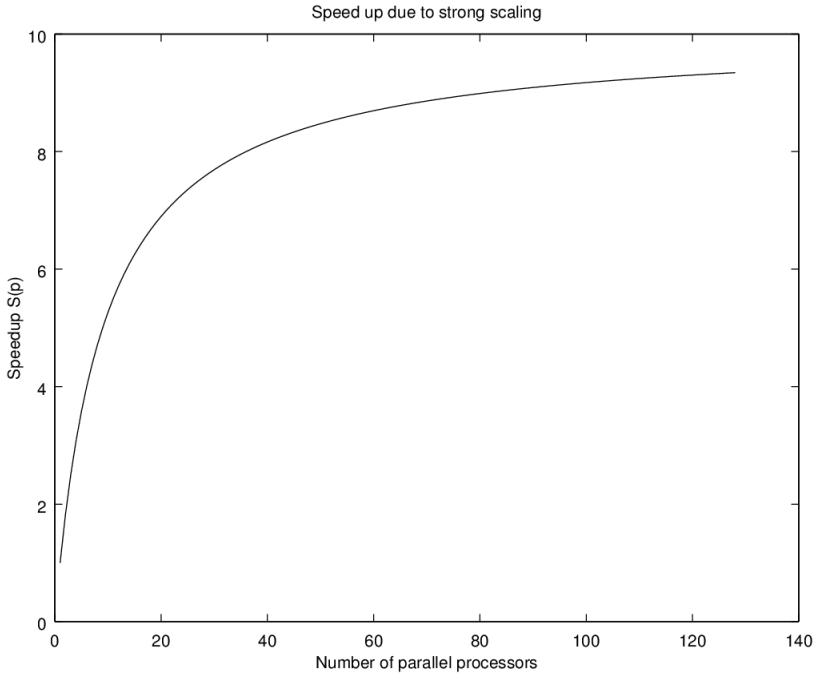
Eric Gao – emg222

Question 1

Amdahl's law tells us that the speed up is bounded from above by:

$$S(p) \leq \frac{1}{\alpha(1-\alpha)/p}$$

Using this relation, the following plot was produced:



The code to produce this image was:

```
function [ speedups ] = q1( alpha, p )
% This function will produce a plot of speedup as a function of number
    of cores (S(p))
% alpha is the fraction of serial work
% p is the number of total cores

```

```
speedups = zeros(1, p);
for i = 1:p
    speedups(i) = 1 / ( alpha + ( 1 - alpha ) / i );
end

plot(speedups, 'k-');
title( 'Speed up due to strong scaling' );
xlabel( 'Number of parallel processors' );
ylabel( 'Speedup S(p)' );

end
```

Question 2

You can have up to p workers working at any one time. So the amount to schedule p tasks is: αp time. Therefore, the time to perform p tasks is bounded from above by $\alpha p + \tau$. Therefore, our theoretical maximum throughput is:

$$\frac{p}{\alpha p + \tau}$$

Question 3

There are a few circumstances under which it would not be best to tune:

1. When you are not running the code more than once or a few times.
2. Performance tuning can reduce the readability of code. When it makes code unnecessarily complex, it may not be worth it.
3. When the code is already running at close to peak performance. Or when you could tune another part of the code that would result an even better speedup.

Question 4

We have 60 cores per accelerator and 15 Xeon Phi accelerator boards. Each core on the Xeon Phi board has a clock speed of 1.053 GHz (http://ark.intel.com/products/71992/Intel-Xeon-Phi-Coprocessor-5110P-8GB-1_053-GHz-60-core). Therefore, assuming we can perform the theoretical maximum of 4 FLOPs per cycle (https://en.wikipedia.org/wiki/FLOPS), we would have the following FLOP rate on just the 15 accelerator boards:

$$15 \text{ accelerators} \times 60 \frac{\text{cores}}{\text{acclerator}} \times 1.053 \times 10^9 \frac{\text{cycles}}{\text{second core}} \times 4 \frac{\text{FLOPs}}{\text{cycle}} = 3790.8 \text{ GFLOPs}$$

We then also have 8 nodes, with 12 cores per node. We also have 2 threads per core due to Intel Hyperthreading. Any core on these 8 nodes has a theoretical maximum clock rate of

3.5 GHz (http://ark.intel.com/products/75283/Intel-Xeon-Processor-E5-2697-v2-30M-Cache-2_70-GHz). Assuming these specs, the FLOP rate of our 8 compute nodes is given by:

$$8 \text{ nodes} \times 12\frac{\text{cores}}{\text{node}} \times 2\frac{\text{threads}}{\text{core}} \times 3.5 \times 10^9 \frac{\text{cycles}}{\text{second thread}} \times 4\frac{\text{FLOPs}}{\text{cycle}} = 2688 \text{ GFLOPs}$$

This gives us a total and theoretical maximum FLOP rate of: $3790.8 + 2688 = 6.4788$ TFLOPs, or about 6.5 TeraFLOPs per second.

Question 5

My machine has a Intel Core i7 running clocked at 4.0 GHz. There are 4 cores, with 2 threads per core due to Intel Hyperthreading. Assuming again 4 FLOPs per clock cycle, my machine has a theoretical maximum FLOP rate of:

$$4 \text{ cores} \times 2\frac{\text{threads}}{\text{core}} \times 4.0 \times 10^9 \frac{\text{cycles}}{\text{second thread}} \times 4\frac{\text{FLOPs}}{\text{cycle}} = 128 \text{ GFLOPs}$$