Homework #3
Due May 16, 2005 in Discussion Section
Revised May 12, 2005

Scalability, Speedup and Measurement

Problem 1. Speedup and Scalability

You have heard in lecture about speedup and scalability. In this problem, we will explore these terms and their utility for reasoning about parallel applications and systems.

a. Consider the following measurements for the execution time of three applications on a parallel system of various sizes.

	1-node	8-nodes	16-nodes	32-nodes	64-nodes
App#1	139s	17s	9s	6s	8s
App#2	309s	40s	25s	20s	18s
App#3	555s	70s	38s	25s	20s

For each of the three applications, compute the speedup and present it in a table, and plot it. Which application has the best speedup at 8, 16, 32, and 64-nodes. Is it the same application for each size system?

b. Using Amdahl's law as a model, compute the serial fraction (in seconds) for App#2 and App#3. Is it possible to compute it for App#1? Explain what would be the best estimate.
c. Using the serial fractions computed in part b, what is the maximum speedup possible for each application based on Amdahl's Law?

Problem 2. Scaled Speedup

An alternative view of application scaling with parallelism was proposed by John Gustafson. Here the idea was that its fair to consider scaling the application size up, as we increase the number of processors in the system.

a. Consider an application with a serial fraction of 0.15 and a 1-node running time of 320 seconds. Using Amdahl's law, what is the maximum possible speedup for this application? What is the speedup for 64 processors? 128 processors?
b. Consider the same application as on part a of this question. However, now let's consider scaled speedup, where we can increase the size of the application linearly with the number of processors. For this application, as the size of the application increases, the serial work increases proportional to the problem size. However, the parallel portion increases as n^2 with the problem size. What speedup is possible for 64 processors? 128 processors? 1024 processors? What is the maximum possible speedup?

c. Consider the same application data as on part a. However, in this case, we will assume a different scaling. We still assume that we can increase the size of the application linearly with the number of processors. This time, as the size of the application increases, the serial work increases proportional to the problem size. However, the parallel portion increases as n * log n with the problem size. What speedup is possible for 64 processors? 128 processors? 1024 processors? What is the maximum possible speedup?
d. Finally, consider the same application data as in part a, and the application scaling as described in part b (relative growth of the serial and parallel portions of the application). Now, we are going to consider "Fixed Time Speedup". What is the maximum fixed time speedup achievable for 64 processors? 128 processors? 1024 processors?

Problem 3

a. Assume a *p*-processor PRAM machine. The shared memory contains *n* distinct items ($p<n$). Describe an efficient algorithm to find the largest value from all these *n* items. How many steps will your algorithm take for an EREW and COMMON CRCW PRAM.
b. Repeat the above problem for the scenario where P_0 has a value *x* and we have to find out the number of elements that are equal to *x*.

Lab Assignment

Go through the Proactive tutorial given in
http://www-csag.ucsd.edu/teaching/cse160s05/homeworks/proactive-tutorial.pdf.

Problem 4

Write a program that integrates the area of the curve 1/x between the range [1, 10^6]. This can be calculated by doing the following:

```
x = 1; limit = 10^6; inc = 0.000125;

while( x < limit)
{
        area += (1/x)*inc;
        x += inc;
}
```

where inc is the interval at which you calculate the intermediate areas. For our problem we set inc = 0.000125.

a. Discuss the scope of parallelism and the way you would divide the task.
b. Assuming an Amdahl's law (*seq part* + *parallel part*), what is the maximum achievable speedup? How many nodes are required to achieve it?

c. Using Gustafson's scaled speedup repeat the node *scaleup* increasing the data set size as you do so.
d. Run the experiment by varying the number of nodes from 1:20 (in steps of 3) and plot performance (1/run_time) vs. # of nodes.
e. Declare the speedup achieved? What could be achieved for 1000 node machine? (*extrapolate*) what are the real limitations?
f. As you scale to 20 nodes, can you maintain linear scalability? If so, why? If not, what are the limiting factors?

Problem 5

Write a program to perform Bucket Sort. Use the two datasets provided in /home/cse160/hw3/data/ as your input values.

a. Discuss the scope of parallelism and the way you would divide the task.
b. Assuming an Amdahl's law (*seq part* + *parallel part*), what is the maximum achievable speedup? How many nodes are required to achieve it?
c. Using Gustafson's scaled speedup repeat the node *scaleup* increasing the data set size as you do so.
d. Run the experiment by varying the number of nodes from 1:20 (in steps of 3) and plot performance (1/run_time) vs. # of nodes.
e. Declare the speedup achieved? What could be achieved for 1000 node machine? (*extrapolate*) what are the real limitations?
f. As you scale to 20 nodes, can you maintain linear scalability? If so, why? If not, what are the limiting factors?