EXAM 2

JASON MEDCOFF

1. Describe concepts and techniques

1. A cost-optimal parallel system is a system in which solving a problem has the same asymptotic growth in parallel as the best known serial solution.

2. Problem size is the number of basic computations in the best serial solution to some problem.

3. Serial factor is the amount of a parallel program that must be executed in serial.

4. An isoefficiency function gives the growth rate of the problem size necessary to keep efficiency of a parallel system fixed as the number of processors increases.

5. A parallel system is scalable if efficiency is fixed when problem size and processing elements are increased.

2. Short answer questions

1. Typical sources of overhead may include communication such as message passing, or idling from thread synchronization.

2. Say that a problem of size W has a serial component W_s. Then we write the parallel execution time as

$$T_p = \frac{W - W_s}{p} + W_s$$

where p is the number of processing elements. Then

$$S = \frac{W}{\frac{W-W_s}{p} + W_s}$$

and it is clear that

$$\lim_{p\to\infty} \frac{W}{\frac{W-W_s}{p} + W_s} = \frac{W}{W_s}.$$

3. Superlinear speedup is exhibited when a speedup occurs by a factor greater than p. This is usually caused by work done by each processing element in a parallel solution to be less when combined than one serial solution. An example is one processor not having enough cache to fit the entire problem data, while distributing the problem data among many processors alleviates this issue.

4. The graphs illustrate exactly Amdahl's law; as the number of processors increases, efficiency tends toward some limit asymptotically, and speedup asymptotically approaches some upper bound.

5. A system is cost optimal if and only if the isoefficiency function is linear in the problem size as the system is scaled up.

6. For an ideal system, the isoefficiency function is a linear function of p.

7. We know that T_0 must grow with p, since every program has some serial component. Then while one processing element is performing the serial component, say in time t_s, the other $p-1$ processors are idle such that

$$T_0 \geq t_s(p-1)$$

Therefore, T_0 is at least linear in p due to this unavoidable overhead.

8. The isoefficiency function is optimal if and only if the degree of concurrency of the parallel algorithm is a linear function of the problem size W.

3. Calculation

1. Parallel runtime is given by

$$T_p = \Theta(n/p + \log p)$$

and cost is given by

$$C = O(n + p\log p).$$

Since speedup is given as parallel runtime over serial runtime, we have

$$S = \frac{n/p + \log p}{n}$$

and efficiency is

$$E = \frac{S}{p}.$$

So we can calculate with $n = 512$ and $p = 16$ that parallel runtime is 36 units, cost is 576 units, speedup is $36/512 \approx .0703$, and efficiency is 0.0044. As long as n is on the order of $p\log p$, the cost is linear in n and we are cost optimal. This is shown here, since 512 is “close” to 576.

2. If we assume we have at least eight processing elements, we can at best compute groups of tasks according to a diagonal line from bottom left to top right, crossing through nodes at the same distance from the starting node. This yields an execution time of 15 units.

3. The corresponding speedup equation is

$$\frac{t_c n^3}{t_c\frac{n^3}{p} + (t_s + 2t_w)\log p}$$

For memory constrained scaling, we assume memory grows linearly with p. We have memory $m = n^2$, so $n^2 = cp$ for some constant c. Then the scaled speedup is

$$\frac{t_c(cp)^{1.5}}{t_c\frac{(cp)^{1.5}}{p} + (t_s + 2t_w)\log p} = O(p)$$

For the time constrained problem, we force $T_p = O(n^3/p)$ constant, giving $n^3 = cp$ for some constant c. Then speedup is

$$\frac{t_c(cp)}{t_c\frac{(cp)}{p} + (t_s + 2t_w)\log p} = O(p).$$