
Written Assignments (50 points)

1.1. Amdahl's Law (25 points)

1.1.1. $\frac{1}{q_1+\frac{q_2}{2}}$

1.1.2. q_1 is the fraction of WORK that must be done sequentially, so the max speedup of this alone would be $\frac{1}{\frac{q_1}{1}+\frac{1-q_1}{\infty}} = \frac{1}{q_1}$, as lecture 13 discusses. However, we must also account for the fraction of WORK that can use at most 2 processors, q_2. This would be $\frac{1}{\frac{q_1}{1}+\frac{q_2}{2}+\frac{1-q_1-q_2}{\infty}} = \frac{1}{q_1+\frac{q_2}{2}}$ because q_1 is the fraction of WORK that is bounded by 1 processor, q_2 is the fraction of WORK that is bounded by 2 processors, and $1 - q_1 - q_2$ is the fraction of WORK that is unbounded. In the case that $q_1 = 0$ and $q_2 = 0$, we get $\frac{1}{0+0+\frac{1-q_1-q_2}{\infty}} = \frac{1}{0}$ = DNE, which is reasonable because we could never have an infinite number of processors. As another example, if $q_1 = 1$, then the max speedup would be 1 according to my solution, which makes sense because we would only be able to do WORK sequentially, using one processor at a time. Likewise, if $q_2 = 1$. the max speedup would be 2 according to my solution, which makes sense because now we would be able to do WORK using two processors at a time. For these reasons, I believe my solution is correct.

1.2. Finish Accumulators (25 points)

*count*1 can only have the value 0 because a.get() is called on line 12 before `finish` is done, so it returns the initial value of accumulator a. On the other hand, the possible values for both *count*0 and *count*2 are Q such that $0 \leq Q \leq \frac{N}{M}$. This is because *count*0 increases every time we find a match of the pattern array in the text array, and because a.get() is called after `finish` completes on line 15, so *count*2 gets the final sum of occurrences of the pattern array in the text array, as computed using accumulator a. The minimum bound is 0 because of the event that the pattern array is not in the text array at all. However, the max bound of Q is $\frac{N}{M}$ because N is the size of the text array, and M is the size of the pattern array, so the maximum number of instances we could find of the pattern array in the text array is $\frac{N}{M}$.