2 Amdahl's and Gustafson's Law

Assuming a program consists of 50% non-parallelizable code.

a) Compute the speed-up when using 2 and 4 processors according to Amdahl's law.

$$S_2 = \frac{1}{0.5 + \frac{1-0.5}{2}} = \frac{1}{\frac{3}{4}} = \frac{4}{3} \approx 1.33$$

$$S_4 = \frac{1}{0.5 + \frac{1-0.5}{4}} = \frac{1}{0.625} = \frac{8}{5} = 1.6$$

b) Now assume that the parallel work per processor is fixed. Compute the speed-up when using 2 and 4 processors according to Gustafson's law.

$$S_2 = 0.5 + 2(1-0.5) = 1.5$$

$$S_4 = 0.5 + 4(1-0.5) = 2.5$$

c) Explain why both speed-up results are different.

Gustafson is more optimistic because it assumes that runtime is constant and there is no asymptotic bound for speedup (i.e. sequential part is not limiting factor).

3 Amdahl's and Gustafson's Law II

a) The analysis of a program has shown a speedup of 3 when running on 4 cores. What is the serial fraction according to Gustafson's law?

$$S_4 = 3 = f + (1-f) \cdot 4$$

$$\Rightarrow 3 = -3f + 4 \Rightarrow -1 = -3f \Rightarrow \underline{\underline{f = \frac{1}{3}}}$$

b) The analysis of a program has shown a speedup of 3 when running on 4 cores. What is the serial fraction according to Amdahl's law (assuming best possible speedup)?

$$S_4 = 3 = \frac{1}{f + \frac{1-f}{4}} \Rightarrow \frac{1}{3} = f + \frac{1-f}{4}$$

$$\Rightarrow \frac{4}{3} = 3f + 1 \Rightarrow \frac{1}{3} = 3b \Rightarrow \underline{\underline{f = \frac{1}{9}}}$$

4 Task Graph

Assuming you want to add eight numbers, then two options to do this are

$$\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{\underbrace{1+2}_{+}+3}_{+}+4}_{+}+5}_{+}+6}_{+}+7}_{+}+8}_{+}$$

and

$$\underbrace{\underbrace{\underbrace{1+2}_{+}+\underbrace{3+4}_{+}}_{+}+\underbrace{\underbrace{5+6}_{+}+\underbrace{7+8}_{+}}_{+}}_{+}$$

a) Given those two variants, determine the length of the critical path for both computations.

first: 7

second: 3

b) For a sequence of length n, determine the length of the critical path using the two approaches from above (accumulator method and Divide and Conquer).

accumulater: $n-1$

divide and conquer: $\lceil \log_2(n) \rceil$