Lecture 4: Parallel Speedup and Amdahl’s Law

Mack Joyner
mjoyner@rice.edu

http://comp322.rice.edu
As before, WORK = 26 and CPL = 11 for this graph
T_2 = 15, for the 2-processor schedule on the right
We can also see that \( \text{max}(\text{CPL}, \text{WORK}/2) \leq T_2 < \text{CPL} + \text{WORK}/2 \)
There are 4 idle slots in this schedule — can we do better than \( T_2 = 15 \) ?
Parallel Speedup

• Define \( \text{Speedup}(P) = \frac{T_1}{T_P} \)
  — Factor by which \( P \) processors speeds up execution time relative to 1 processor, for fixed input size
  — For ideal executions without overhead, \( 1 \leq \text{Speedup}(P) \leq P \)
  — You see this with abstract metrics, but bounds may not hold when measuring real execution times with real overheads
  — Linear speedup
    - When \( \text{Speedup}(P) = kP \), for some constant \( k \), \( 0 < k < 1 \)

• Ideal Parallelism  = \( \text{WORK} / \text{CPL} = \frac{T_1}{T_{\infty}} \)
  = Parallel Speedup on an unbounded (infinite) number of processors
Assume greedy schedule, input array size $S$ is a power of 2, each add takes 1 time unit

- $\text{WORK}(G) = S-1$, and $\text{CPL}(G) = \log_2(S)$
- Define $T(S,P)$ = parallel execution time for Array Sum with size $S$ on $P$ processors
- Use upper bound $T(S,P) \leq \text{WORK}(G)/P + \text{CPL}(G)$ as a worst-case estimate

$$T(S,P) = \frac{\text{WORK}(G)}{P} + \text{CPL}(G) = \frac{S-1}{P} + \log_2(S) \Rightarrow \text{Speedup}(S,P) = \frac{T(S,1)}{T(S,P)} = \frac{S-1}{((S-1)/P + \log_2(S))}$$
How many processors should we use?

Define $\text{Efficiency}(P) = \frac{\text{Speedup}(P)}{P} = \frac{T_1}{P \times T_P}$

—Processor efficiency --- figure of merit that indicates how well a parallel program uses available processors

—For ideal executions without overhead, $\frac{1}{P} \leq \text{Efficiency}(P) \leq 1$

—$\text{Efficiency}(P) = 1$ (100%) is the best we can hope for
How many processors should we use?

What should be the minimum efficiency to determine how many processors we should use?
Array Sum: Speedup as a function of array size $S$ and number of processors $P$

- $\text{Speedup}(S,P) = \frac{T(S,1)}{T(S,P)} = \frac{(S-1)}{((S-1)/P + \log_2(S))}$
- Asymptotically, $\text{Speedup}(S,P) \rightarrow \frac{(S-1)}{\log_2 S}$, as $P \rightarrow \infty$
Amdahl’s Law

If \( q \leq 1 \) is the fraction of WORK in a parallel program that must be executed sequentially for a given input size \( S \), then the best speedup that can be obtained for that program is \( \text{Speedup}(S,P) \leq \frac{1}{q} \).
Amdahl’s Law

- Observation follows directly from critical path length lower bound on parallel execution time
  - CPL >= q * T(S,1)
  - T(S,P) >= q * T(S,1)
  - Speedup(S,P) = T(S,1)/T(S,P) <= 1/q

- Upper bound on speedup simplistically assumes that work can be divided into sequential and parallel portions
  - Sequential portion of WORK = q
    - also denoted as $f_S$ (fraction of sequential work)
  - Parallel portion of WORK = 1-q
    - also denoted as $f_p$ (fraction of parallel work)
Illustration of Amdahl’s Law:
Best Case Speedup as function of Parallel Portion
Announcements & Reminders

• No lab tomorrow
• Quiz #1 available today, due Friday, Jan. 31st at 11:59pm
• HW #1 due on Wednesday, Jan. 29th at 11:59pm
• IMPORTANT: Watch video & read handout for topic 2.1 for lecture on Friday