Parallel Performance

Steve Lantz
Computing and Information Science 4205
www.cac.cornell.edu/~slantz

Parallel Speedup

- **Parallel speedup: how much faster does my code run in parallel compared to serial?**
- **Max value is *p*, the number of processors**

$$\text{Parallel speedup} = \frac{\text{serial execution time}}{\text{parallel execution time}}$$

Parallel Efficiency

- **Parallel efficiency: how much faster does my code run in parallel compared to *linear speedup from* serial?**
- **Max value is 1**

$$\text{Parallel efficiency} = \frac{\text{Parallel speedup}}{p}$$

Model of Parallel Execution Time

- **Let n represent the problem size and p be the number of processors**
- **Execution time of the inherently sequential portion of the code is $\sigma(n)$**
- **Execution time of the parallelizable portion of the code $\phi(n) / p$**
- **Overhead is due parallel communication, synchronization is $\kappa(n,p)$**

$$T(n,p) = \sigma(n) + \varphi(n) / p + \kappa(n,p)$$

Model of Parallel Speedup

- **Divide $T(n,1)$ by $T(n,p)$**
- **Cute notation by Quinn: ψ (or Greek psi) = parallel speedup**

$$\psi(n,p) = \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n)/p + \kappa(n,p)}$$

Amdahl's Law

- **Parallel overhead κ is always positive, so…**
- **Can express it in terms of sequential fraction $f = \sigma(n) / [\sigma(n) + \varphi(n)]$**
- **Divide numerator, denominator by $[\sigma(n) + \varphi(n)]$**

$$\psi(n,p) < \frac{\sigma(n) + \varphi(n)}{\sigma(n) + \varphi(n) / p}$$

Other Expressions for Amdahl's Law

$$\psi(n,p) < \frac{1}{f + (1 - f)/p}$$

$$\psi(n,p) < \frac{p}{1 + f(p - 1)}$$

Parallel Overhead: Synchronization (Example)

- **Suppose the parallelizable part of the code consists of *n* tasks each taking the same time *t***
- **What if *n* is not divisible by *p*? Let $x = \mathrm{mod}(n,p)$ = the "extra" tasks**
- **Sequential execution time: $t\, n = t\,(n - x) + t\, x$**
- **For parallel execution, only the <u>first</u> term shrinks inversely with *p***
- **For *p* processors, the <u>second</u> term takes either 0 or *t* (because $x < p$)**
- **Therefore, time on *p* processors = $t\,(n - x) / p + t\,\{1 - \delta_{0,x}\}$**

$$T(n,p) = \sigma(n) + \varphi(n) / p + \kappa(n,p)$$

$$\varphi(n) = t\,[n - \mathrm{mod}(n,p)], \quad \kappa(n,p) = t\,\{1 - \delta_{0,\mathrm{mod}(n,p)}\}$$

Aside: How Do You Code a Kronecker Delta in C?

- **Simple once you see it… not so simple to come up with it…**
- **Formula assumes 0 <= x < p**

```
Kron(0,x) = 1 - (x + (p-x)%p)/p)
```

Steve Lantz
Computing and Information Science 4205
www.cac.cornell.edu/~slantz

Parallel Overhead: Jitter (Example)

Steve Lantz
Computing and Information Science 4205
www.cac.cornell.edu/~slantz

Communication

- **Latency**
- **Bandwidth**