Performance Metrics

Parallel Computing - Theory and Practice (2/e)
Section 3.6
Michael J. Quinn
Performance Metrics

Speedup = \frac{(\text{Time on 1 CPU})}{(\text{Time on } p \text{ CPUs})}

Efficiency = \frac{\text{speedup}}{p}

What do we compare against?
- parallel program on 1 CPU?
- equivalent sequential program?
- best sequential program?
Example: sorting

• Suppose a parallel bubble sort program obtains 100% efficiency

• Efficiency relative to quicksort:

\[
\frac{N \log N}{N^2/p} = \frac{\log N}{N}
\]
How to Cheat with Speedups…

Program with hardware floating point support

- Sequential program: 100 sec
- Parallel program on 10 CPUs:
  - computation: 10 sec
  - communication: 10 sec
- Speedup = $\frac{100}{(10 + 10)} = 5$

Without hardware floating point support

- Sequential program: 400 sec
- Parallel program on 10 CPUs:
  - computation: 40 sec
  - communication: 10 sec
- Speedup = $\frac{400}{(40 + 10)} = 8$
Can Speedup be Super linear?

Super linear speedup: speedup > p

Negative search overhead (for search problems)
  Applies to certain input problems, not to average case
  Sequential program could simulate parallel program to get same behavior

Better caching behavior on parallel systems
Amdahl's Law

Let $f =$ fraction of code that must be performed sequentially

$$\text{Speedup} = \frac{T(1)}{T(p)} \leq \frac{T}{(f + \frac{1-f}{p}) \times T} = \frac{1}{(f + \frac{1-f}{p})} < \frac{1}{f}$$

Example $f = 10\% \implies \text{speedup} \leq 10$, even if $p$ is infinite!

So much for parallel computing ??
Amdahl's Law - the True Story

In practice, $f$ depends on the problem size

Goal of parallel computing is to solve *large* problems fast

If problem size increases, maximum speedup increases
How to Measure Performance

Use wall clock time

```c
int start, stop;
start = get_time();
do computations ... 
stop = get_time();
print(stop-start);
```

Exclude initialization

- Measures I/O performance

Avoid debugging statements
Try to understand performance behavior
  Just attributing poor performance to 'communication overhead' is unacceptable

Determine where the time is spent
  Measure number of messages per second, data volume per second, idle time