Performance Metrics

$$\text{Speedup} = \frac{Time\ on\ 1\ CPU}{Time\ on\ P\ CPUs}$$

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

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: $\dfrac{\dfrac{N \log N}{N^2/p}}{p} = \dfrac{\log N}{N}$

How to Cheat with Speedups (1)

Program with hardware floating point support

```
Sequential program: 100 sec
Parallel program on 10 CPUs:
     computation: 10 sec
     communication: 10 sec
Speedup = 100 / (10 + 10) = 5
```

How to Cheat with Speedups (2)

Without hardware floating point support

```
Sequential program: 400 sec
Parallel program on 10 CPUs:
     computation: 40 sec
     communication: 10 sec
Speedup = 400 / (40 + 10) = 8
```

Can Speedup be Superlinear?

Superlinear speedup: speedup $\geq$ 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

Speedup $\leq \frac{1}{f+(1-f)/p}$

Example: f=10% $\Rightarrow speedup \leq 10$, 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

Scaled speedup $= \frac{time\ \ sequential\ \ program\ \ would\ \ have\ \ taken}{time\ \ parallel\ \ program}$

How to Measure Performance

Use wall clock time

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

Exclude initialization

  Measures file I/O performance

Avoid debugging statements

Performance Debugging

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