IE 495 Lecture 6

September 14, 2000

Reading for This Lecture

- Primary
  - Paper by Kumar and Gupta
  - Paper by Gustafson
- Secondary
  - Roosta, Chapter 5

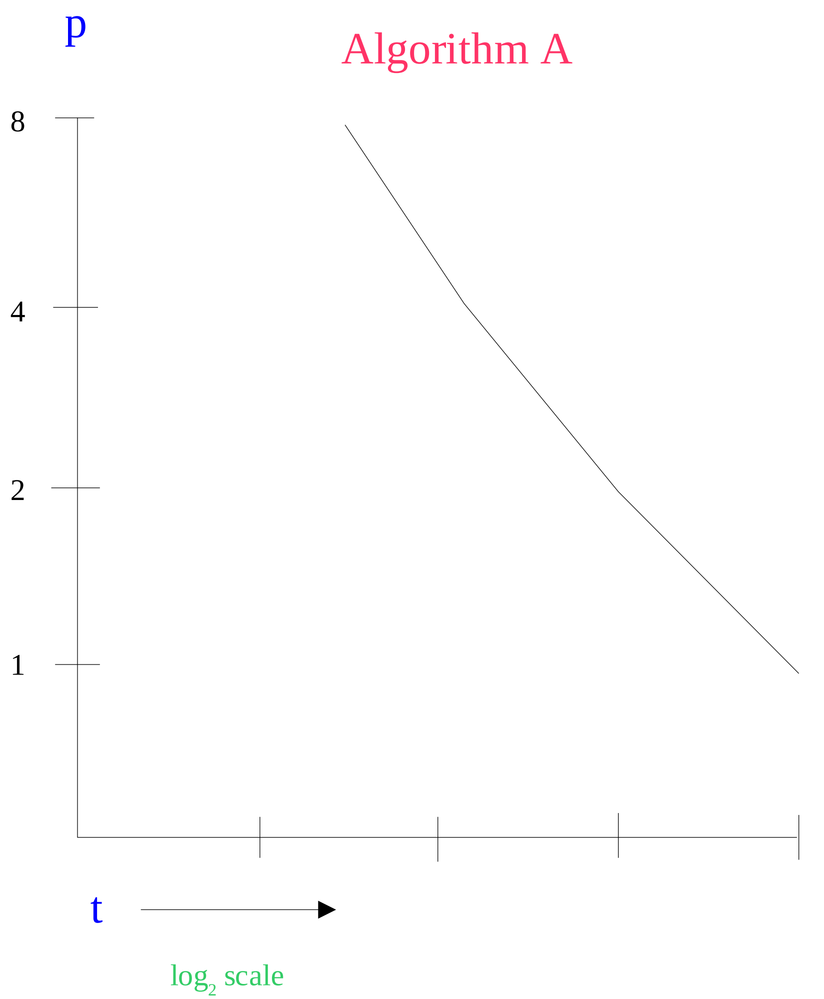
Analyzing Parallel Algorithms

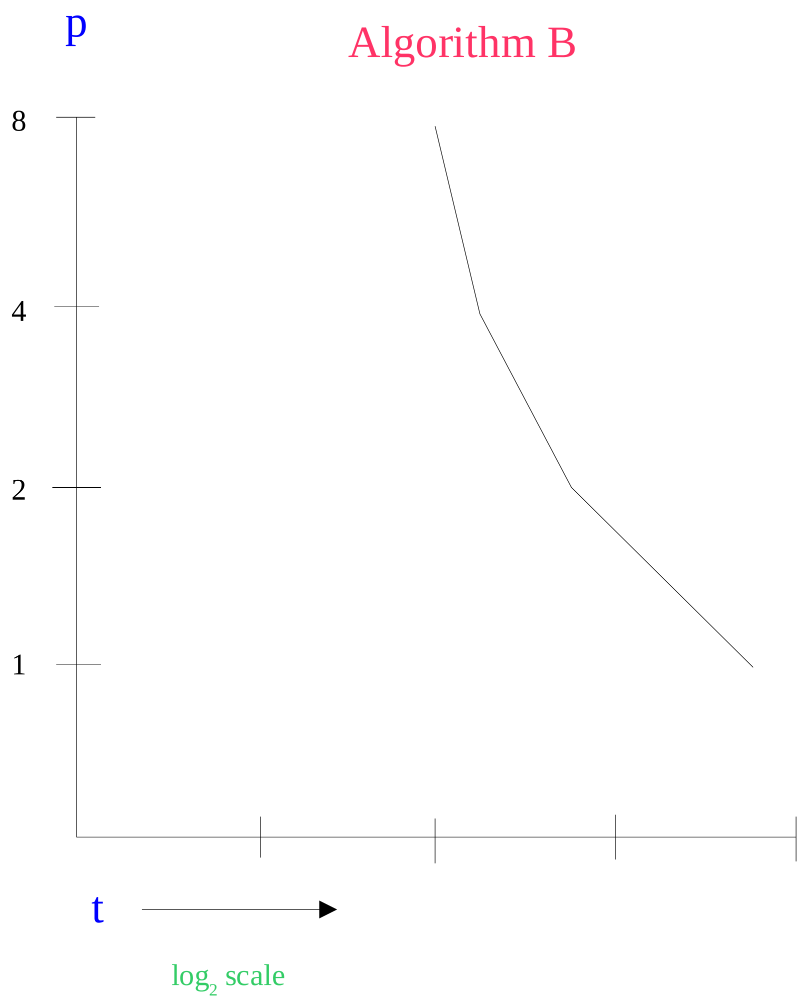
Parallel Systems

- A parallel system is a parallel algorithm plus a specified parallel architecture.
- Unlike sequential algorithms, parallel algorithms cannot be analyzed very well in isolation.
- One of our primary measures of goodness of a parallel system will be its scalability.
- Scalability is the ability of a parallel system to take advantage of increased computing resources (primarily more processors).

Scalability Example

Which is better?





Terms and Notations

Sequential Runtime	T_1
Sequential Fraction	s
Parallel Fraction	$p = 1 - s$
Parallel Runtime	T_N
Cost	$C = NT_N$
Parallel Overhead	$T_o = C - T_1$
Speedup	$S_N = T_1 / T_N$
Efficiency	$E = S_N / N$

Definitions and Assumptions

- The sequential running time is usually taken to be the running time of the best sequential algorithm.
- The sequential fraction is the part of the algorithm that is inherently sequential (reading in the data, splitting, etc.)
- The parallel overhead includes all additional work that is done due to parallelization.
  - communication
  - nonessential work
  - idle time

Cost, Speedup, and Efficiency

- These three concepts are closely related.
- A parallel system is cost optimal if $C = T_1$.
- A parallel system is said to exhibit linear speedup if $S = N$.
- Hence, linear speedup $\Leftrightarrow$ cost optimal $\Leftrightarrow$ $E = 1$
- If $E > 1$, this is called super-linear speedup.

Factors Affecting Speedup

- Sequential Fraction
- Parallel Overhead
  - Unecessary/duplicate work
  - Communication overhead/idle time
  - Time to split/combine
- Task Granularity
- Degree of Concurrency
- Sychronization/Data Dependency
- Work Distribution
- "Run-up" Time

Amdahl's Law

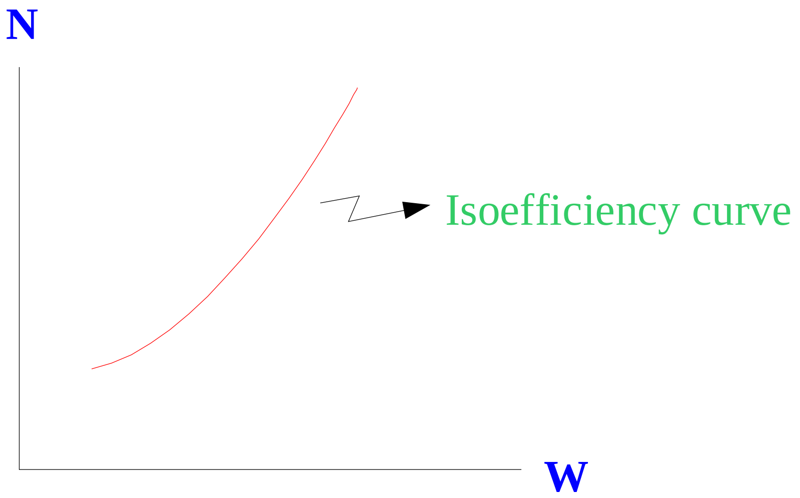
- Speedup is bounded by

$$(s + p)/(s + p/N) = 1/(s + p/N) = N/(sN + p)$$

- This means more processors $\Rightarrow$ less efficient!
- How do we combat this?
- Typically, larger problem size $\Rightarrow$ more efficient.
- This can be used to "overcome" Amdahl's Law.

The Isoefficiency Function

- The isoefficiency function *f(N)* of a parallel system represents the rate at which the problem size must be increased in order to maintain a fixed efficiency



- This function is a measure of scalability that can be analyzed using asymptotic analysis.

Gustafson's Viewpoint

- Gustafson noted that typically the serial fraction does not increase with problem size.
- This view leads to an alternative bound on speedup called scaled speedup.

$$(s + pN)/(s + p) = s + pN = N + (1-N)s$$

- This may be a more realistic viewpoint.

Example: Parallel Prefix

- This example is in Miller and Boxer, Chapter 7