Performance of Parallel Programs

Lecture Coverage

Performance Measures

- Speed-up
- Scalability
- Isoefficiency

Performance Issues

- Computation Time
- Communication Time
- Wait Time

Models and Formulas

- Amdahl's Law
- Contention Free Models
- Operation counts and Asymptotic Analysis

Amdahl's Law

Amdahl's Law states that potential program speedup is defined by the fraction of code (P) which can be parallelized:

$$\text{speedup} = \frac{1}{1 - P}$$

If none of the code can be parallelized, P = 0 and the speedup = 1 (no speedup).

If all of the code can be parallelized, P = 1 and the maximum speedup is infinite (in theory).

If 50% of the code can be parallelized, maximum speedup is 2, meaning the code will run twice as fast. (Assuming an infinite number of processors and no communication or wait time.)

Amdahl's Law - Continued

Introducing the number of processors performing the parallel fraction of work, N, the relationship can be modeled by:

$$\text{speedup} = \frac{1}{P/N + S}$$

where P = parallel fraction, N = number of processors and S = serial fraction.

There are limits to the scalability of parallelism. For example, at P = .50, 0.90 and 0.99 (50%, 90% and 99% of the code is parallelizable):

	speedup		
N	**P = .50**	**P = .90**	**P = .99**
10	1.82	5.26	9.17
100	1.98	9.17	50.25
1000	1.99	9.91	90.99
10000	1.99	9.91	99.02

Performance of Parallel Programs

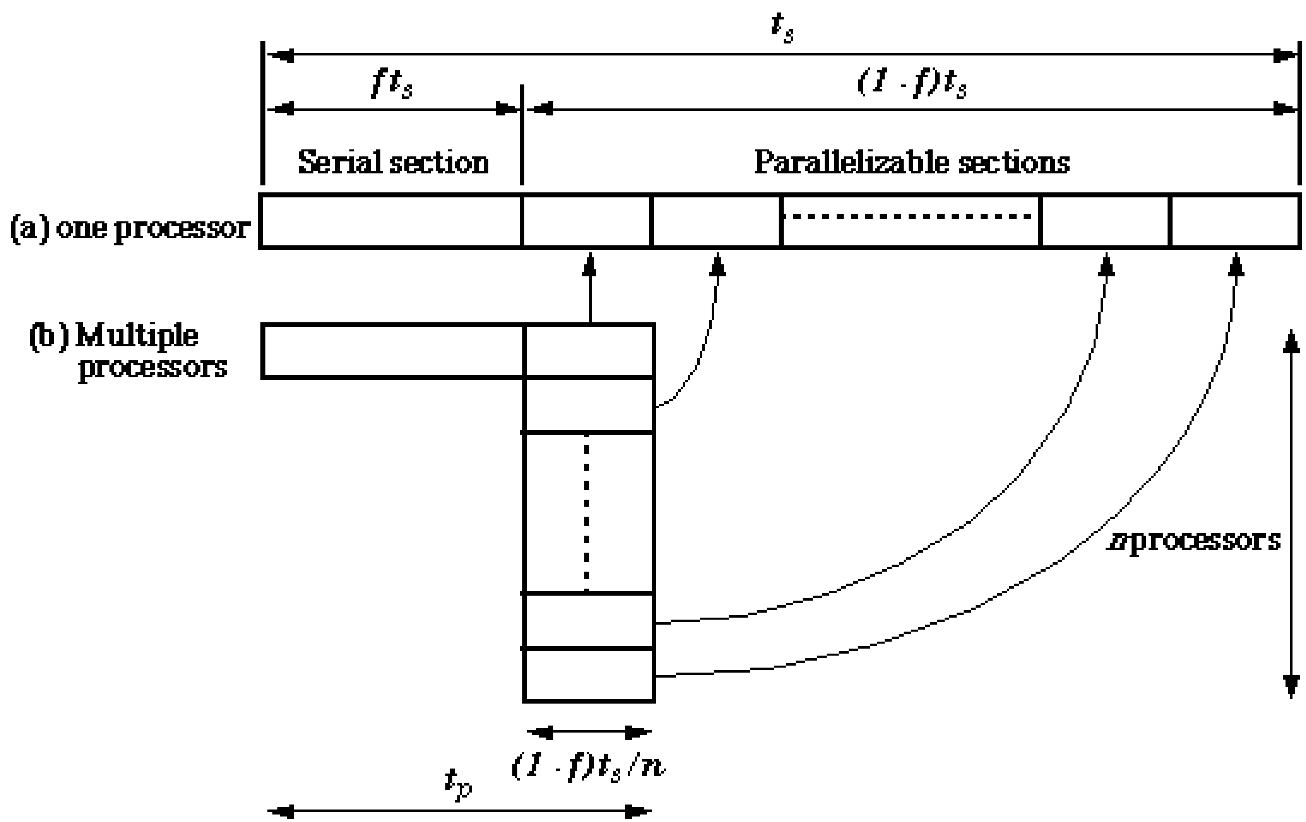
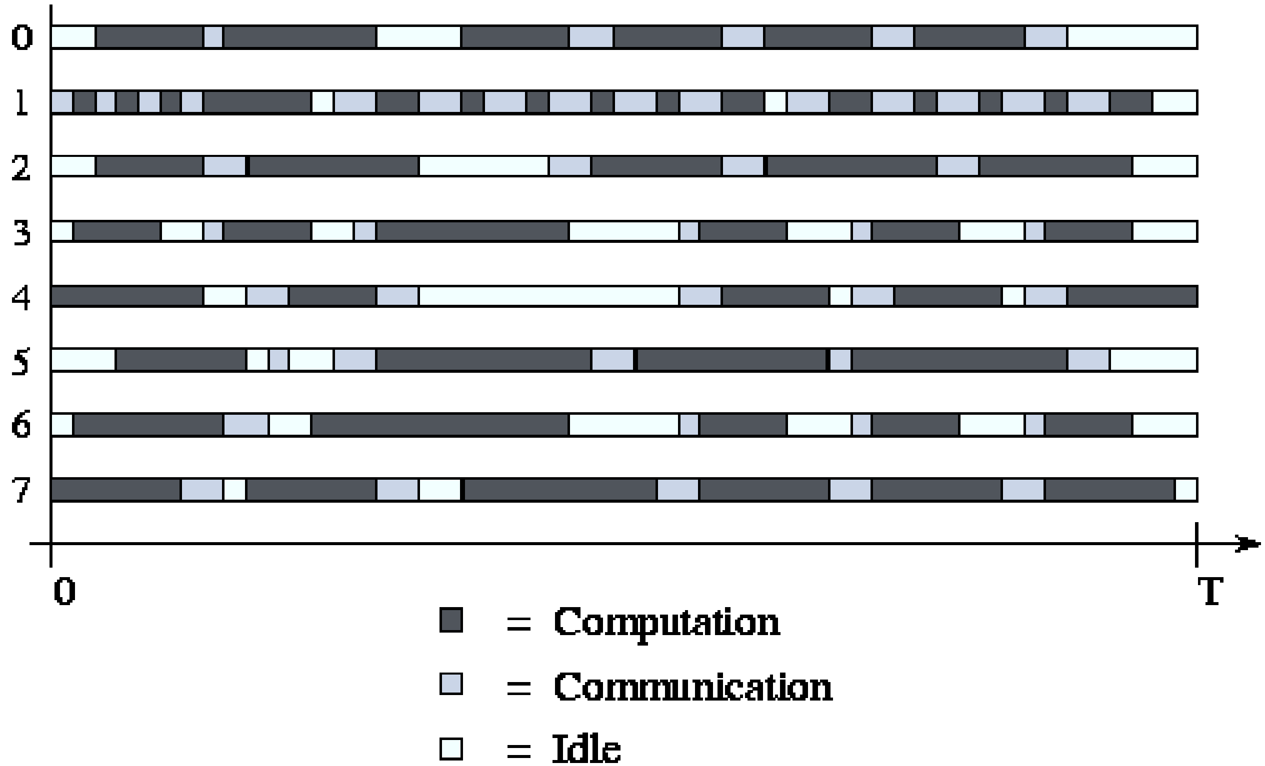


Figure 1.19 Parallelizing sequential problem – Amdahl's law

Simple Analytic Models - Comp+Comm+Idle

Execution behavior of a program



Effect of Communication and Idle Time

$T = T\ (N,P,U,\text{-----})$

$T_j = \quad T^j_{comp} + \quad T^j_{comm} + \quad T^j_{idle}$

If all processors take the same length of time to complete

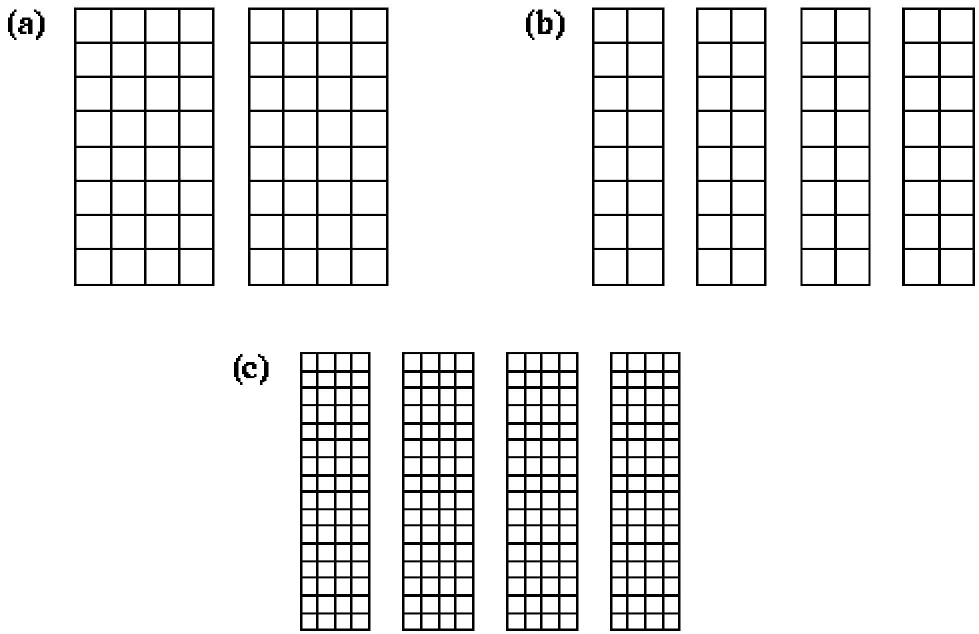
$T = (\sum_{i=1}^{P} T^j_{comp} + \sum_{i=1}^{P} T^j_{comm} + \sum_{i=1}^{P} T^j_{idle})$

But if all processors don't take the same time then

$T = \max\ (T^j_{comp} + T^j_{comm} + T^j_{idle})$

$\quad = \max\ (T_j)$

Parallel efficiency in partitioning of a 2-D grid.



Efficiency as a function of communication cost.

1-D partitioning of a 2-D grid

a) Partition among two processors. E decreases with respect to a)

b) Partition 128 points among four processors. E is the same as for a)

Efficiency and Speed-up

Let P be the number of processors.

$E_{relative} = T_1 / (P\ T_p)$

$S_{relative} = P * E_1 = T_1 / T_p$

$E_{absolute} = T_1$ (best sequential) $/ (P\ T_p)$

Scalability Analysis

What will be the speed-up or efficiency on P processors for N = M?

$$S = f(N,P),\ E = L(N,P)$$

What size problem can I reasonably solve on P processors?

$$T \propto g(N,P)$$

$$E = T_1 / (T_{comp} + T_{comm} + T_{idle})$$

For constant efficiency then T_1 must increase at the same rate as the parallel execution time.

Scalability Analysis

Isoefficiency metric - Establish a relationship between the amount of work, W, to be accomplished and the number of processors,P, such that E remains constant as P increases

Let

$$T_{comm} + T_{idle} = T_{overhead}$$
$$T_{overhead} = T - T_{comm} - T_{idle} = T_O$$
$$T_O = T_O\,(W,P)\ ,\ T_{comp} = T_{comp}(W)$$
$$W = W(\text{problem size})$$

For simple matrix multiply,

problem size $\sim N^3$

Isoefficiency metric

$T_P = (W + T_o (W, P))/P, W = T_1$

$S = W/T_p = WP/(W + T_o)$

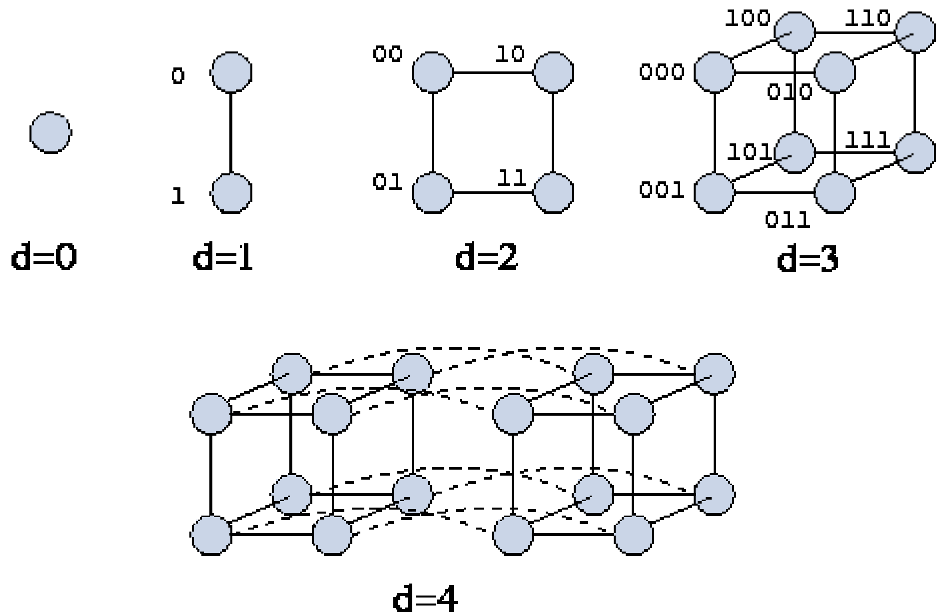
$E = S/P = W/(W + T_o(W,P))$

$T_o(W,P)$ is an increasing function of P so,
if W is constant and P increases then E decreases.
E will remain constant if $T_o(W,P)/W$ is constant
To obtain constant E, W must increase as P is increased.

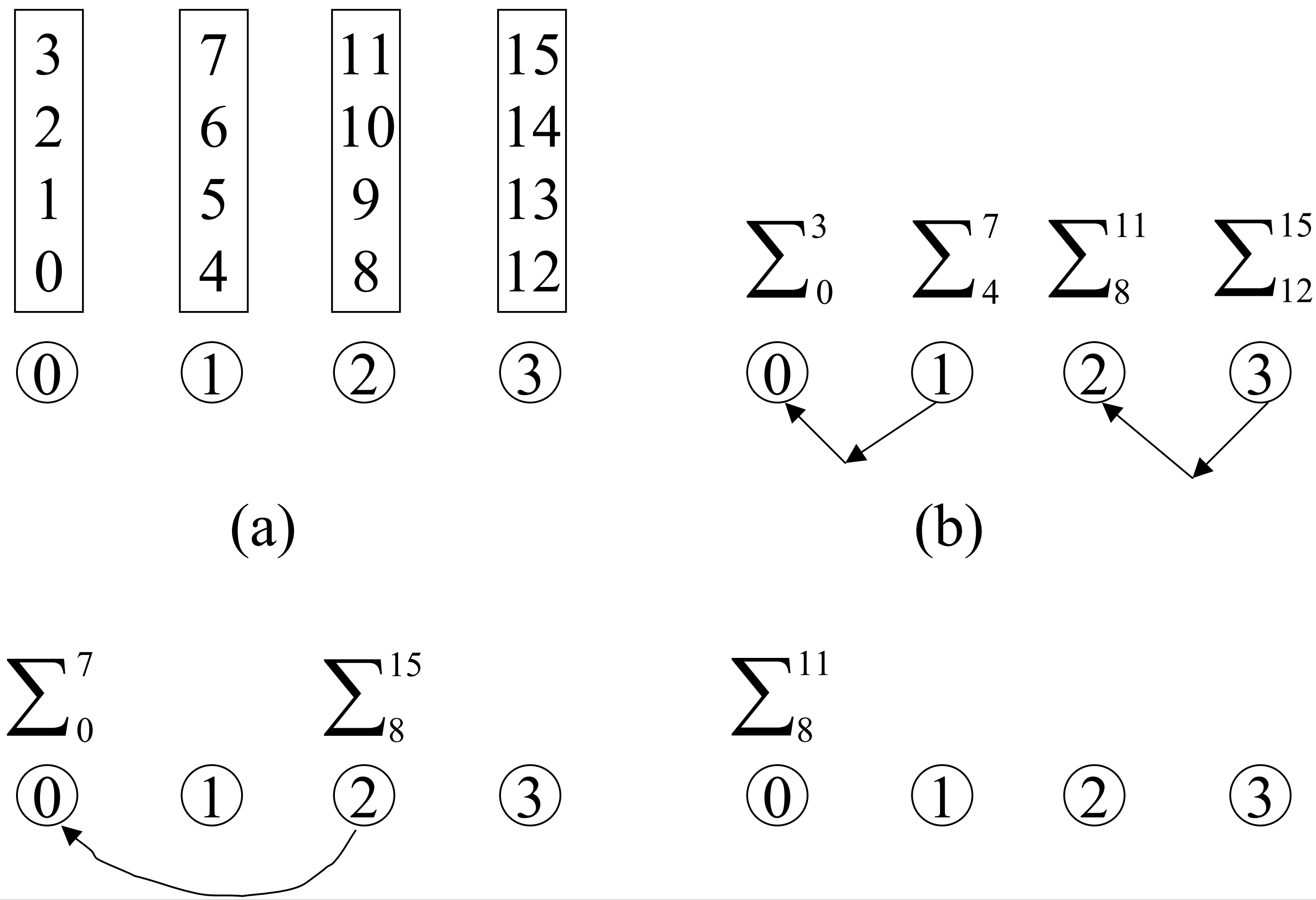
or

$W = K(N)^* T_o(W,P)$, K = isoefficiency function

Hypercube Interconnection Networks for 1,2,3,4 dimensions.



Adding Numbers on a Hypercube (4 Processors)



Adding Numbers on a Hypercube

Let an add take 1 unit of time

Let a unit communication take 1 unit of time

Adding n/p numbers => n/p + 1

$T_1 \sim W \sim n - 1$

$T_O \sim \log p$

$T_p \sim n/p + \log p$

$S \sim n/(n/p + \log p) = np/(n + p \log p)$

$E \sim S/p = n/(n + p \log p)$

$E \sim W/(W+p \log p)$

To make E constant when p is increased to p',

$W \sim n * p' \log p' / p \log p$

$K = p' \log p' / p \log p$

Scalability of Adding Numbers on a Hypercube

E = .8 for n = 64, p = 4

Then for E = .8 for 8 processors

W = n*8*3/4*2 = n * 3 = 192

E = .8 for n = 192, p = 8