



UFOP



PCC104

Projeto e Análise de Algoritmos

Joubert de Castro Lima

joubertlima@gmail.com

Professor Adjunto – DECOM

UFOP 2010/1

Figuras retiradas do livro
Introduction to parallel Computing

Programar em paralelo

Imagine duas matrizes A e B de ordem $m \times n$.
Como realizar a soma $C=(A+B)$ de forma paralela.
Elabore um algoritmo e descreva a técnica utilizada de maneira informal.

Programar em paralelo

- Imagine um arquivo txt com os seguintes dados:

a1 b1 c1 d1 e1 f1	+ Id da linha
a2 b2 c2 d2 e2 f2	+ Id da linha
....	
an bn cn dn en fn	+ Id da linha

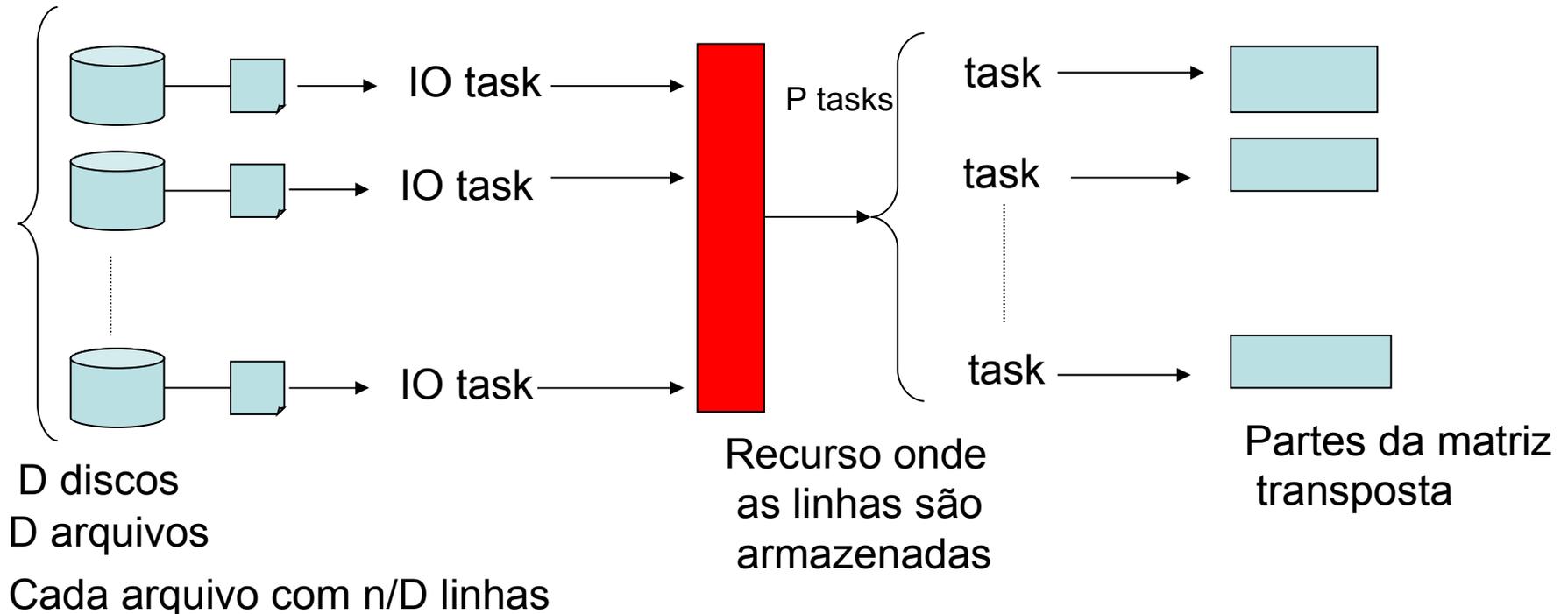
Você possui uma máquina com D discos e P processadores, onde $P > D$.

Elabore um algoritmo paralelo que leia o arquivo e monte a matriz transposta do arquivo original. Utilize idéias para paralelizar IO, dados e tasks

Programar em paralelo

Considere um sistema de endereçamento global com todos os processadores idênticos!!

UM EXEMPLO....



No exemplo acima houve paralelização de tasks, dados e IO ? Justifique fundir a matriz faz sentido??

Antes de começarmos:

Crie tasks

Elabore o grafo de dependabilidade das tasks

Agrupe as tasks em processos usando o grafo de dependabilidade

Escolha uma estratégia que minimize a comunicação

Busque paralelizar IO e processamento

Foque na localidade dos dados

FAÇAM A DISCIPLINA BCC 447 (Programação Paralela)

EM INGLÊS MESMO...

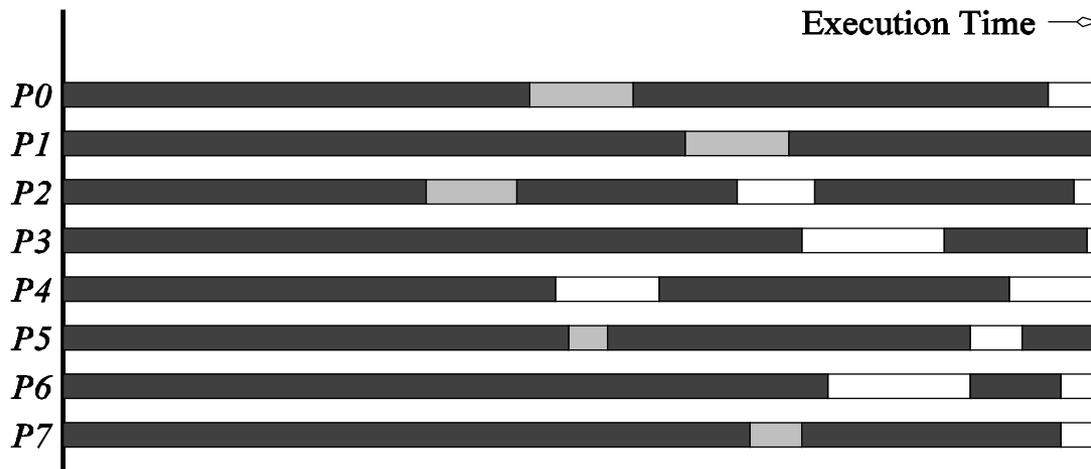
- A sequential algorithm is evaluated by its runtime (in general, asymptotic runtime as a function of input size).
- The asymptotic runtime of a sequential program is identical on any serial platform.
- The parallel runtime of a program depends on the input size, the number of processors, and the communication parameters of the machine.
- An algorithm must therefore be analyzed in the context of the underlying platform.
- A parallel system is a combination of a parallel algorithm and an underlying platform.

Modelagem analítica

USANDO DUAS VEZES MAIS HARDWARE É DE SE ESPERAR QUE O PROGRAMA RODE DUAS VEZES MAIS RÁPIDO

NA TEORIA ATÉ PODE SER....

Programas paralelos normalmente possuem



- 1) Computação extra
- 2) Comunicação
- 3) Espera

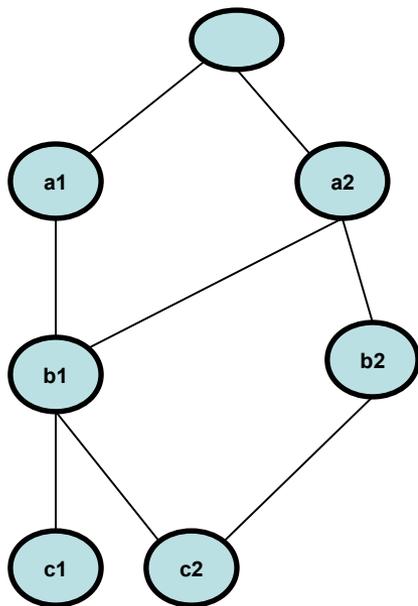
■ Essential/Excess Computation ■ Interprocessor Communication
□ Idling

VAMOS AOS DETALHES:

Computação extra

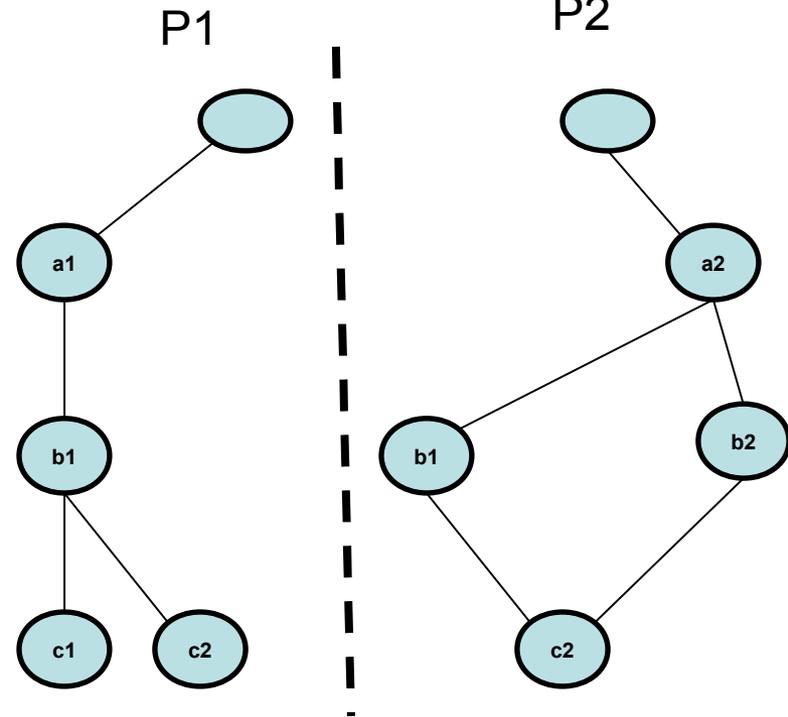
Às vezes algoritmos paralelos possuem maior carga de trabalho do que as versões sequenciais

EX: Vamos a nossa velha Estrutura de Dados



sequencial

CUSTO
EXTRA
RELACIONADO
A DUPLICAÇÃO
DOS NÓS
b1 E c2



Um particionamento para 2 processos

Espera

Normalmente, causada por falta de balanceamento da carga de trabalho entre os processos

PROJETO RUIM

SINCRONIZAÇÃO

DEPENDENDO NÃO TEM JEITO!!!!

Execution time: assim como o sequencial, é o tempo gasto entre o início do programa paralelo e seu término (quando o último processo termina).

Parallel Overhead:

T_s é o tempo sequencial (**a melhor versão sequencial possível**)

$p T_p$ é a soma do tempo gasto por cada processo

$$T_o = p T_p - T_s$$

SpeedUp: é a razão entre o tempo gasto para se resolver um problema usando um processo e o tempo gasto com n processos

Sempre consideramos a melhor versão sequencial como baseline

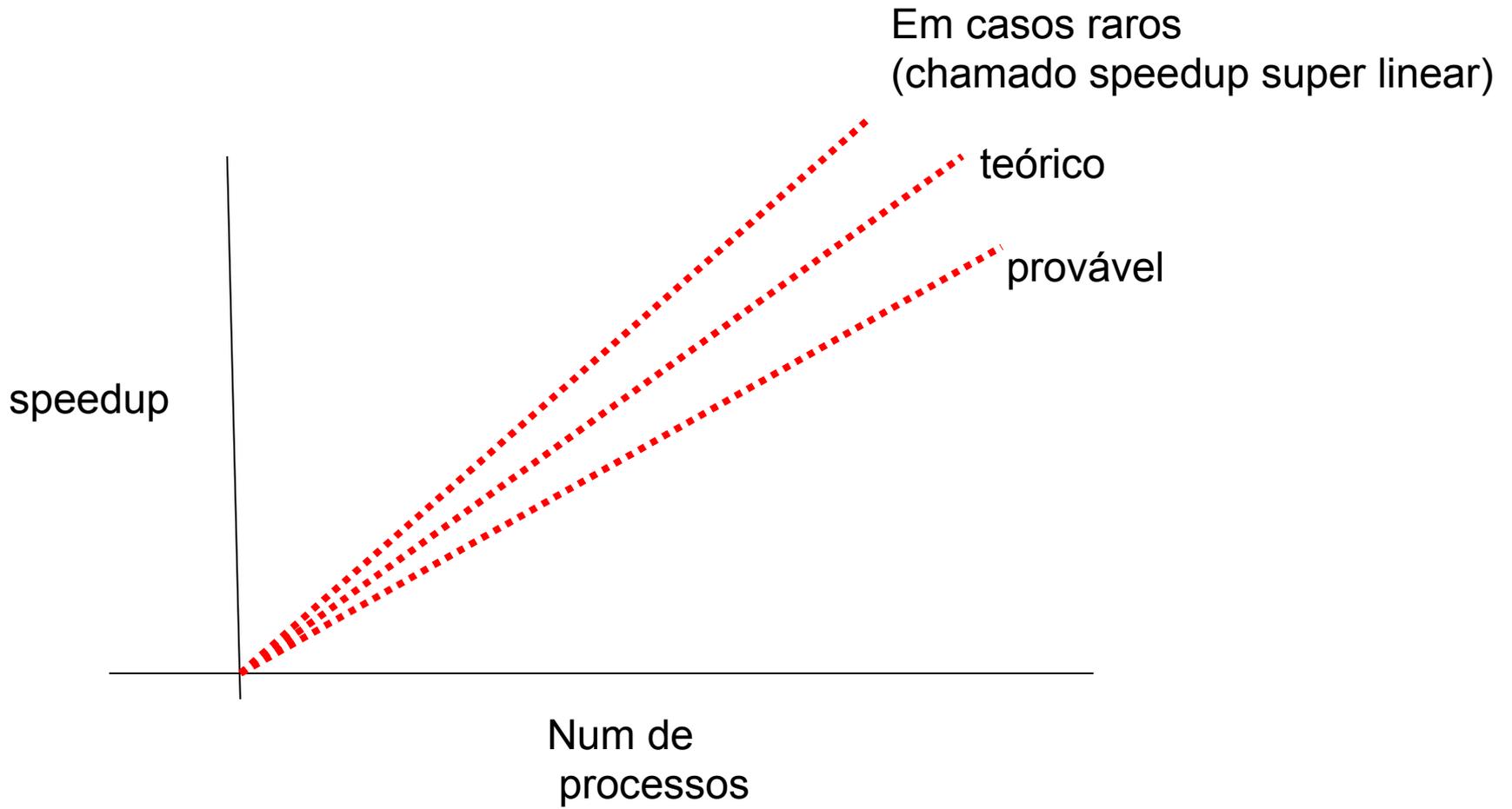
Para ilustrar: Vamos verificar o speedup do Bubble sort paralelo

Sabemos que o sequencial levou 150 segundos

Sabemos que o paralelo levou 40 segundos com 4 processos

Speedup: $150/40 = 3.75$ **ÓTIMO SPEEDUP**

Não é bem assim: se sabemos que o quicksort leva 30 segundos, então speedup: $30/40 = 0.75$ BEM MAIS REALISTA, PORÉM



SUPER LINEAR, COMO ASSIM!!

Imagine que os dados de um problema (carregar um arquivo com n tuplas) são maiores do que a cache disponível

Os dados quando particionados podem diminuir, ficando porções de dados compatíveis com a capacidade de cache de cada nó de processamento

Neste exemplo o speedup da versão paralela poderia ser super linear (PODERIA.....)

SUPER LINEAR, COMO ASSIM!!

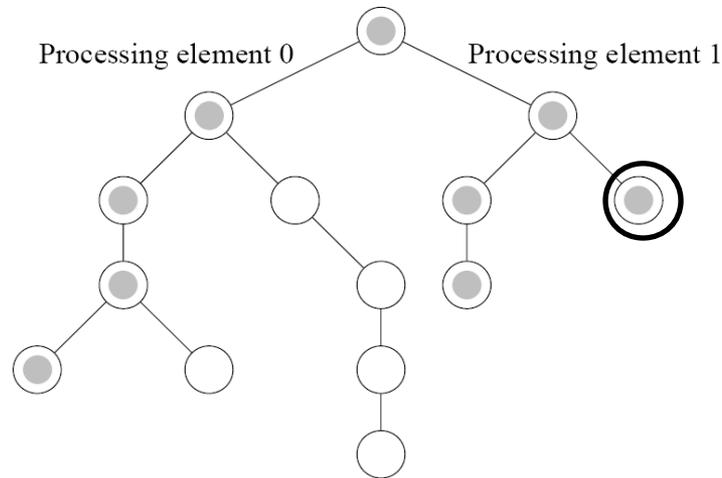


Figure 5.3 Searching an unstructured tree for a node with a given label, 'S', on two processing elements using depth-first traversal. The two-processor version with processor 0 searching the left subtree and processor 1 searching the right subtree expands only the shaded nodes before the solution is found. The corresponding serial formulation expands the entire tree. It is clear that the serial algorithm does more work than the parallel algorithm.

$$T_s = 14t_c$$

$$T_p = 5t_c$$

SPEEDUP: 2,8 para dois processos

Eficiência

Fica quase sempre entre 0 e 1

$$E = \frac{S}{p}$$

Speedup
Número de processos

Mundo ideal: speedup
igual a p e eficiência
igual a 1, mas.....

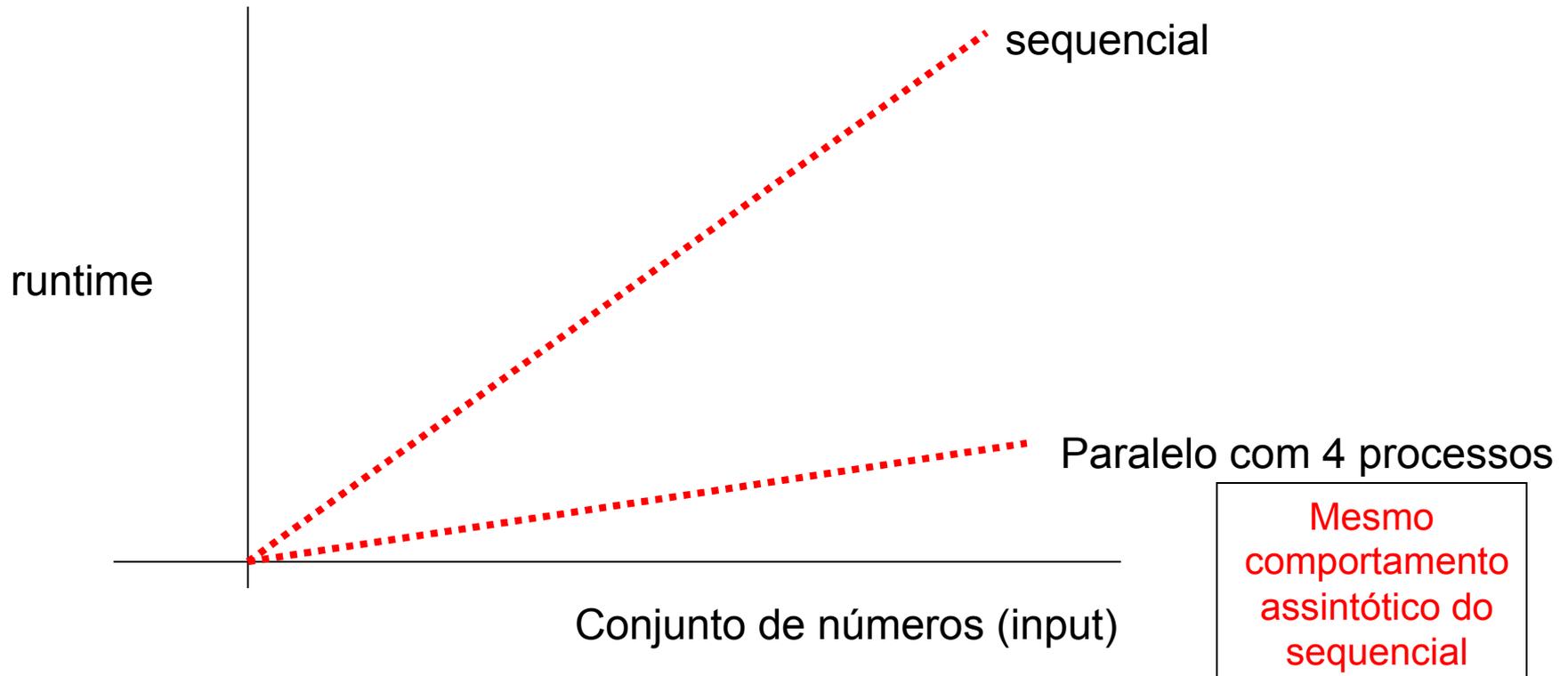
Custo

$$p \times T_p$$

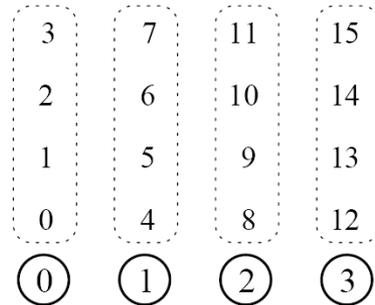
Custo ótimo: o programa paralelo deve ter o mesmo comportamento assintótico do sequencial (variando de acordo com o input)

Custo ótimo

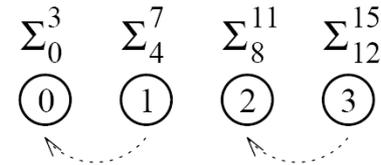
Somar 16 números



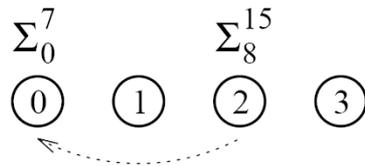
Custo ótimo



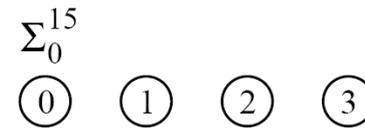
(a)



(b)



(c)



(d)

Figure 5.6 A cost-optimal way of computing the sum of 16 numbers using four processing elements.

Escalabilidade

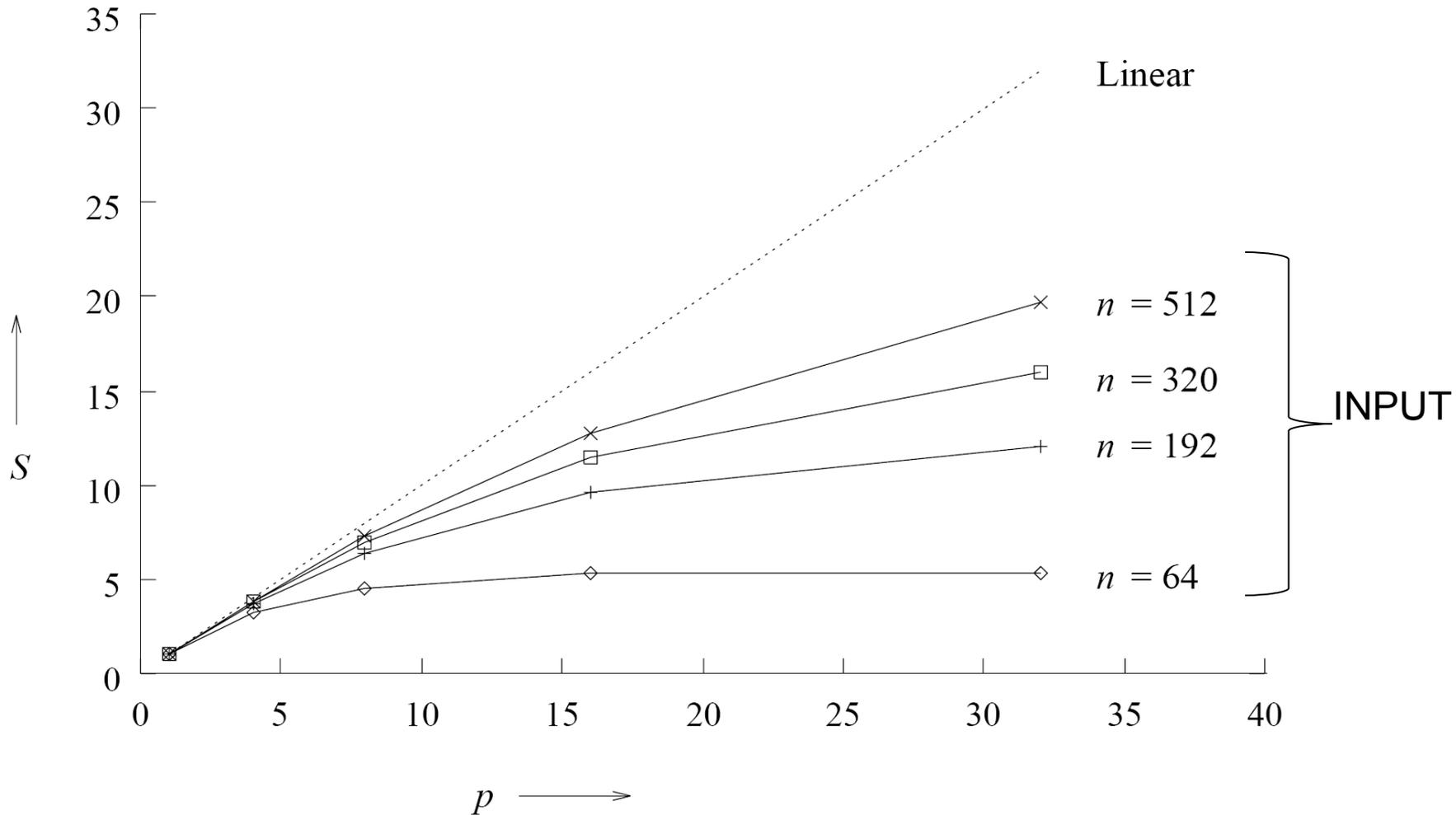


Figure 5.8 Speedup versus the number of processing elements for adding a list of numbers.

Notas sobre escalabilidad

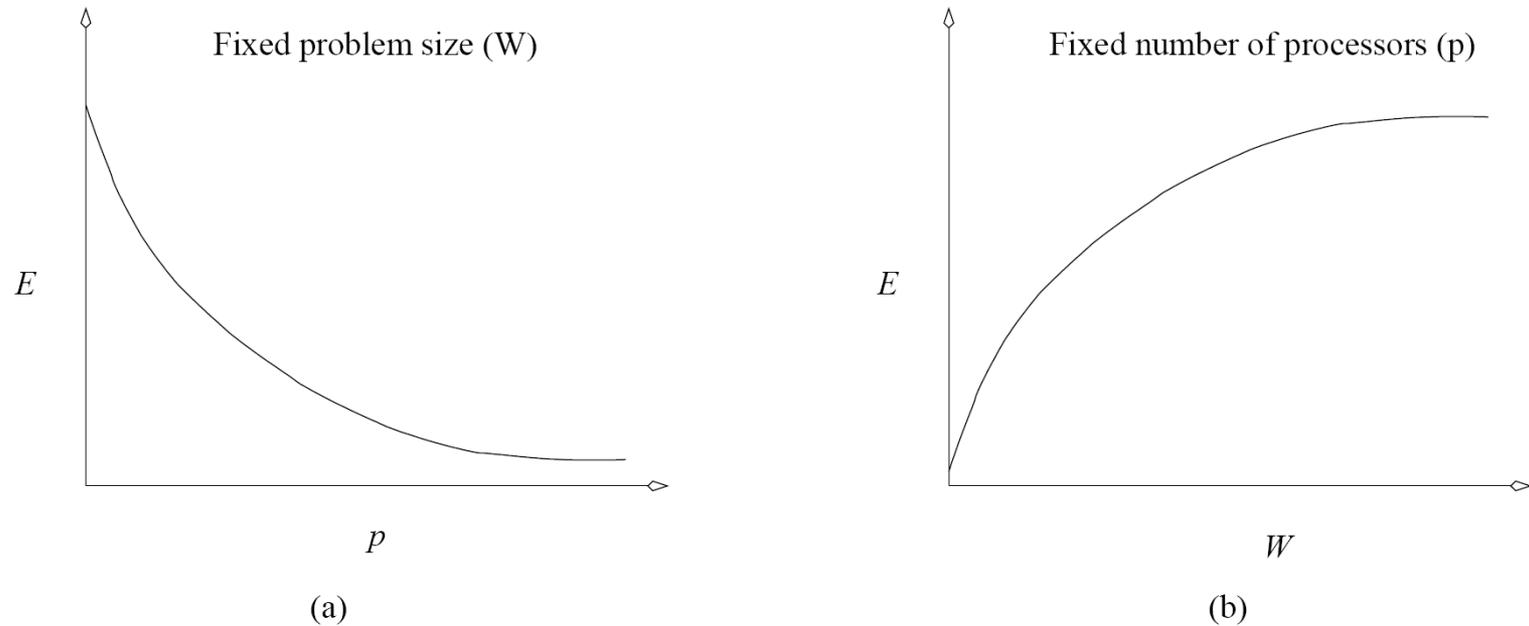


Figure 5.9 Variation of efficiency: (a) as the number of processing elements is increased for a given problem size; and (b) as the problem size is increased for a given number of processing elements. The phenomenon illustrated in graph (b) is not common to all parallel systems.

Fazendo uma análise

Table 5.2 Comparison of four different algorithms for sorting a given list of numbers. The table shows number of processing elements, parallel runtime, speedup, efficiency and the pT_P product.

	Algorithm	A1	A2	A3	A4
Número de processos	p	n^2	$\log n$	n	\sqrt{n}
Tempo de 1 processo	T_P	1	n	\sqrt{n}	$\sqrt{n} \log n$
speedup	S	$n \log n$	$\log n$	$\sqrt{n} \log n$	\sqrt{n}
eficiência	E	$\frac{\log n}{n}$	1	$\frac{\log n}{\sqrt{n}}$	1
custo	pT_P	n^2	$n \log n$	$n^{1.5}$	$n \log n$

Quem é o melhor??

Próxima aula

Implementar um exemplo e calcular todas as métricas de desempenho aprendidas nesta aula

Apresentar o trabalho sobre este tema

Local : lab do DECOM ou pavilhão?