

Avaliação de Desempenho de Programas Paralelos



Programação Paralela Avançada - PPA

Mestrado em Computação Aplicação – MCA
Programa de Pós-Graduação em Computação Aplicada – PPGCA
Centro de Ciências Tecnológicas - CCT
Universidade do Estado de Santa Catarina – UDESC

Profs Maurício A. Pillon e Guilherme P. Koslovski

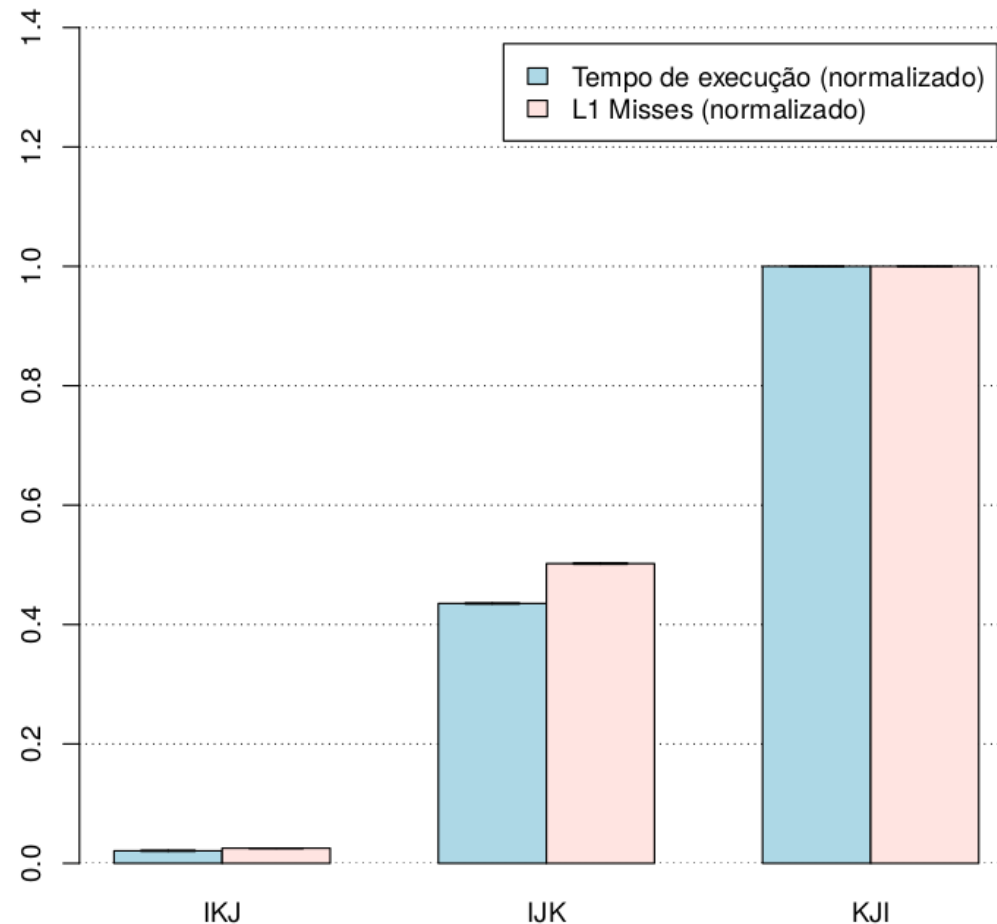
Linha de Sistemas Computacionais
Grupo de Pesquisa de Redes de Computadores e Sistemas Distribuídos
Laboratório de Pesquisa LabP2D

Agenda

- **Projeto de programas paralelos**
 - **Metodologia de particionamento**
- Exemplo: Multiplicação de matrizes
- Avaliação de desempenho de aplicações paralelas
- Considerações finais

Multiplicação de Matrizes

Impacto de Caches Misses no Desempenho (Multiplicação de Matrizes)



Avaliação de desempenho de aplicações paralelas



- Mensurar o impacto do paralelismo sob uma determinada aplicação/algoritmo
- Conceitos
 - Aceleração
 - Eficiência
 - Escalabilidade

Avaliação de desempenho de aplicações paralelas

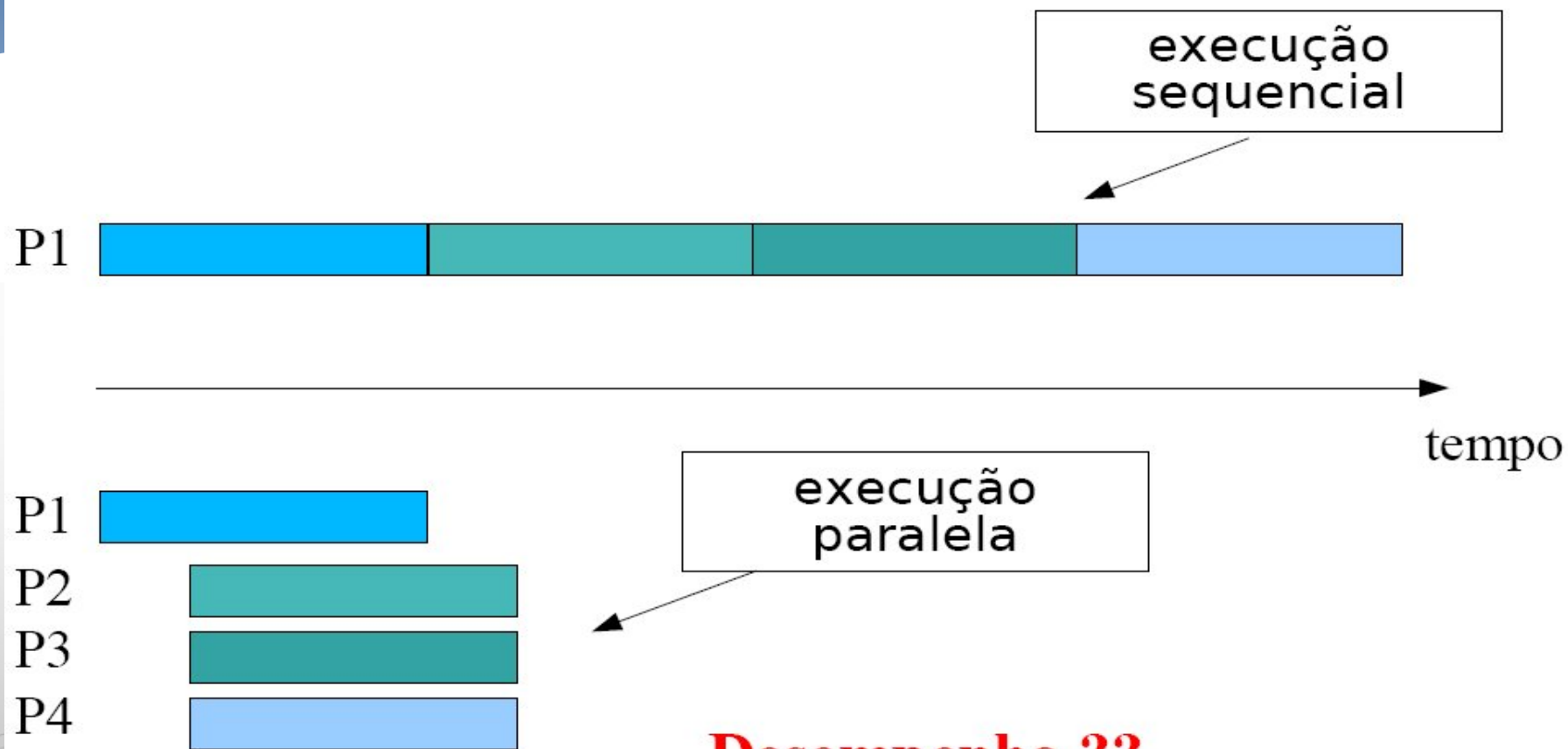


- Algoritmos sequenciais são avaliados em função de seus tempos de execução, normalmente expressos em função do tamanho de sua entrada
- Algoritmos paralelos não dependem exclusivamente do tamanho da entrada sendo também influenciados por suas computações relativas e velocidades de comunicação entre os processos
- **Usando-se duas vezes mais recursos de hardware espera-se que um programa seja executado duas vezes mais rápido!**
- Em programas paralelos isto raramente acontece devido a perdas associadas (overheads) com o paralelismo

Overhead de paralelismo

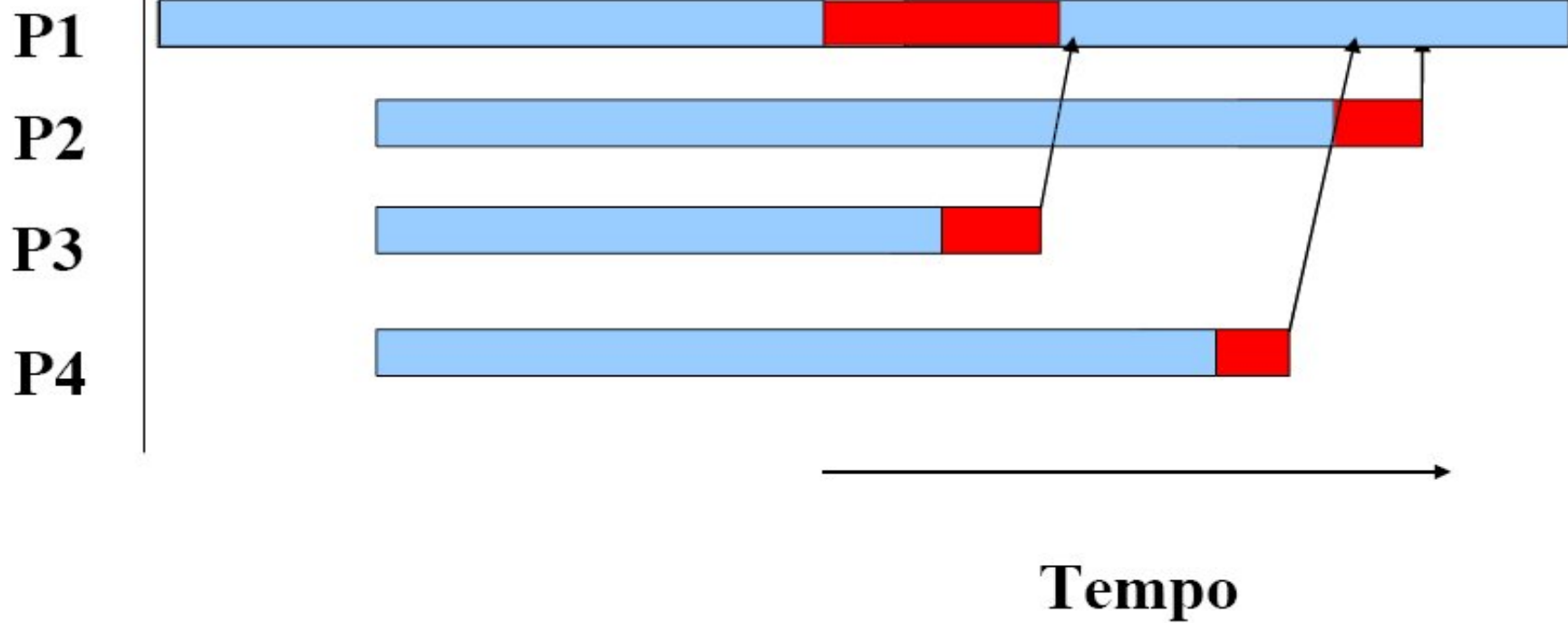
- Tempo necessário para coordenar as tarefas paralelas
- Tempo para iniciar uma tarefa
 - Identificação da tarefa
 - Procura de um processador
 - Carregamento da tarefa
 - Carregamento dos dados
- Tempo para terminar uma tarefa
- Sincronização
- Uma quantificação precisa destas perdas é essencial para compreender a performance de programas paralelos

Execução sequencial x paralela

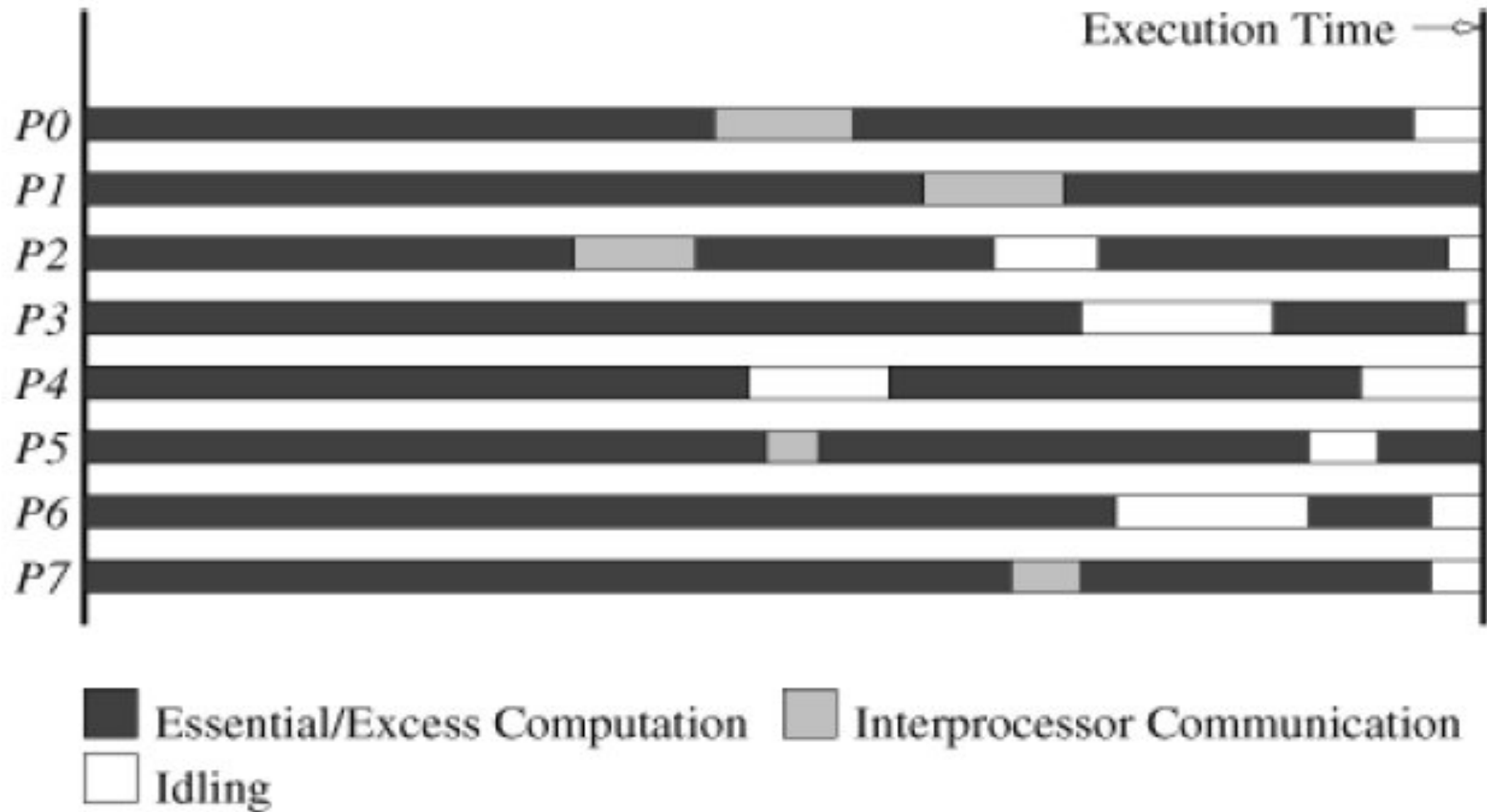


Desempenho ??

Fontes de perdas



Fontes de perdas



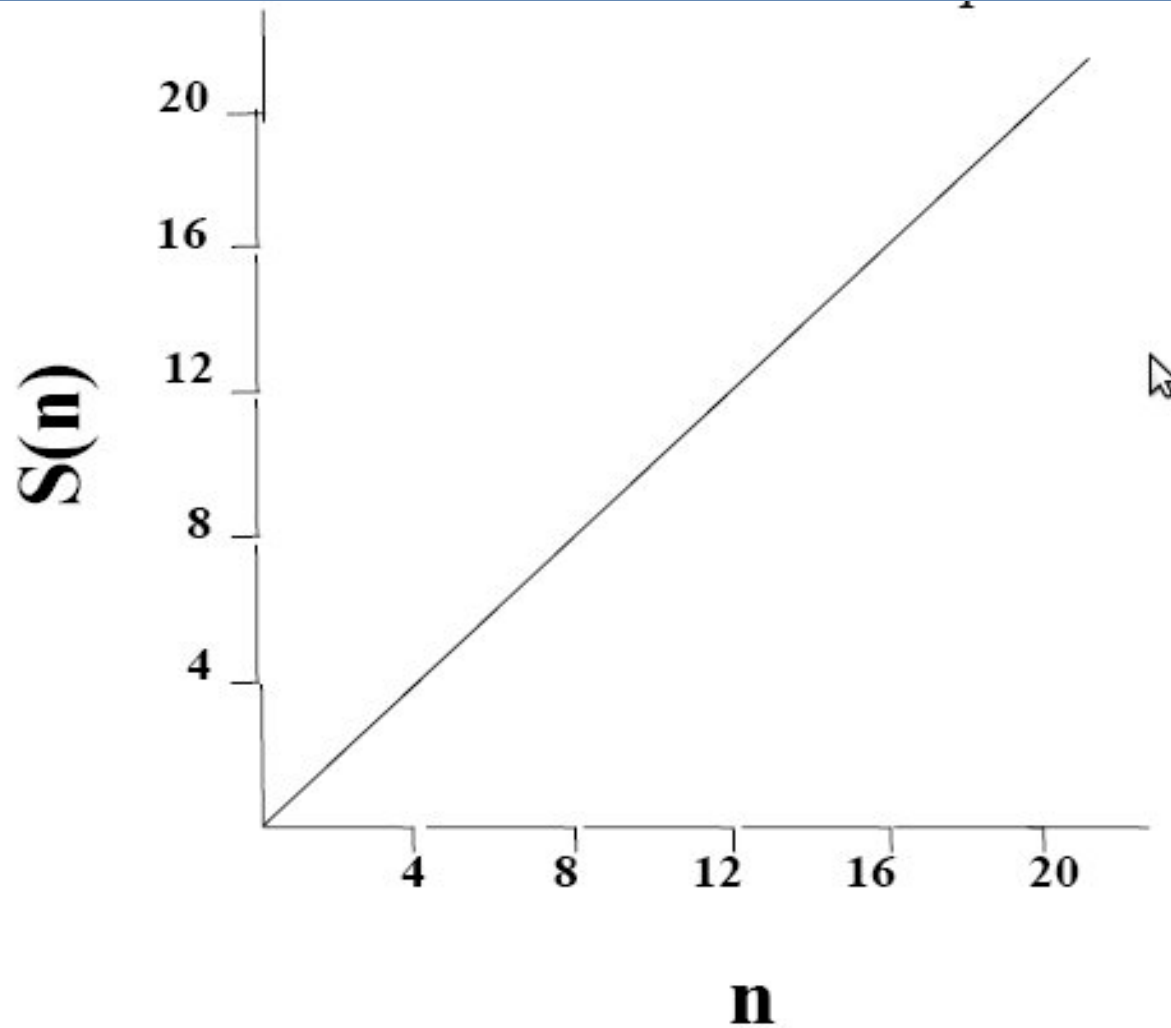
Fontes de perdas

- Interação entre processos
 - Qualquer sistema paralelo não trivial necessita que suas tarefas interajam (comunicação)
 - Geralmente a fonte mais significativa de perdas em processamento paralelo é o tempo gasto em comunicações de dados
- Ociosidade de processadores
 - Desbalanceamento de carga
 - Sincronização
 - Presença de componentes seriais em um programa
- Em muitas aplicações paralelas é impossível prever o tamanho das subtarefas

Tempo de execução

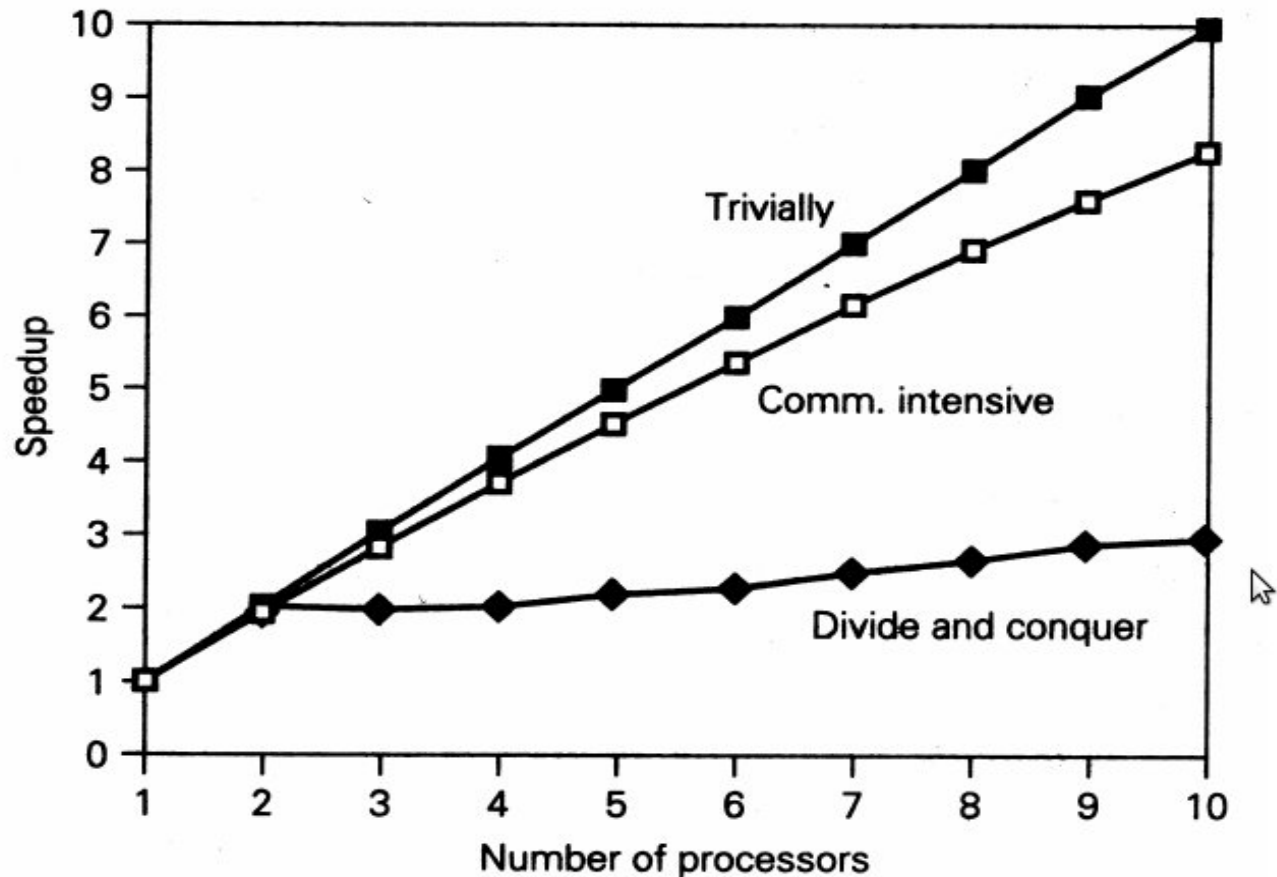
- Tempo de execução serial (TS)
 - É o tempo decorrido entre o início e o final de sua execução em um computador sequencial
- Tempo de execução paralelo (TP)
 - É o tempo transcorrido entre o início de uma computação paralela até o término do último elemento de processamento
- **Aceleração (*speedup*)**
- $S(n) = TS/TP$

Speedup: ideal



Speedup: limites

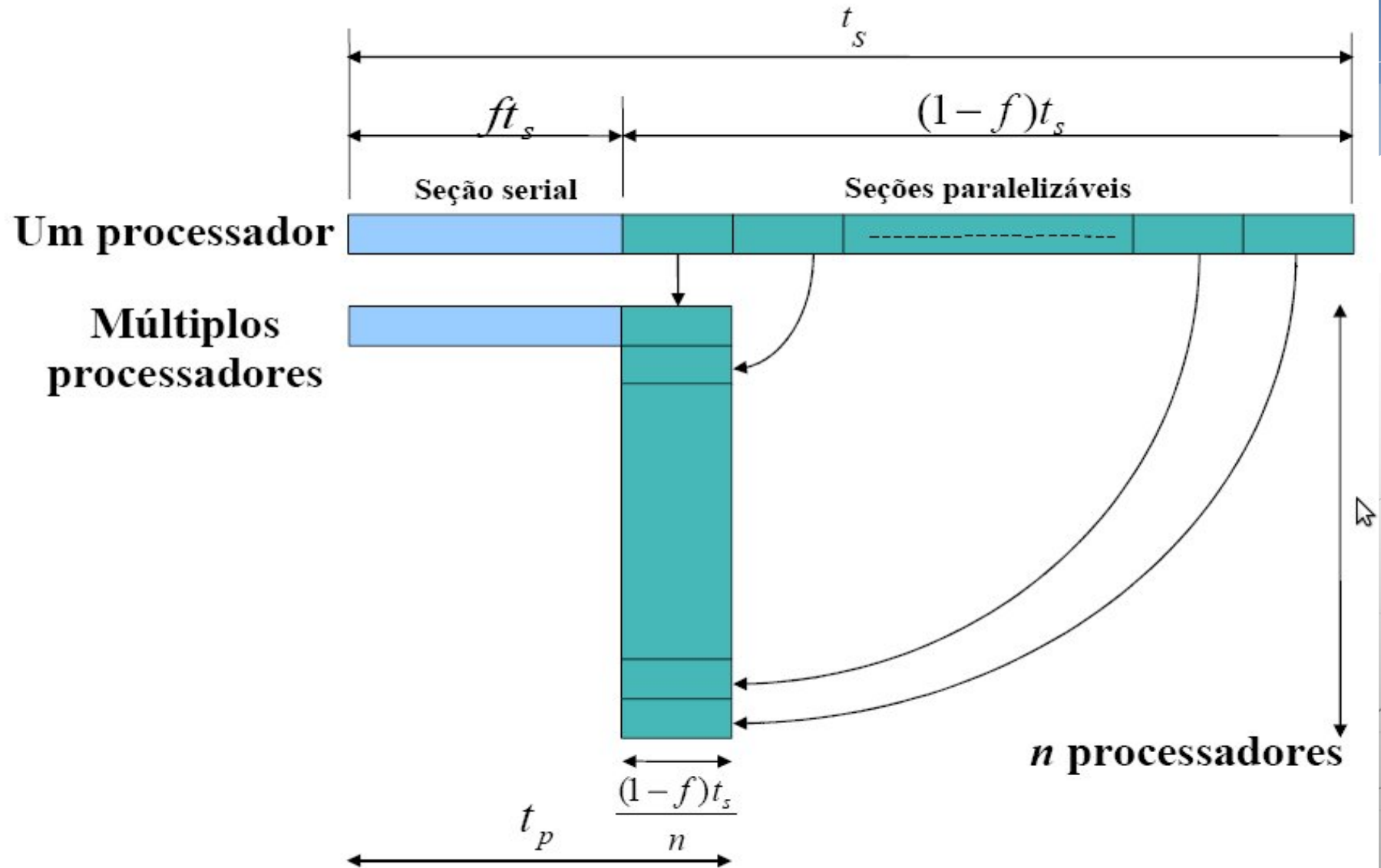
Limites para o paralelismo



Eficiência

- É a medida da fração de tempo para o qual um elemento de processamento é empregado de forma proveitosa.
- Razão do speedup e do número de elementos de processamento
- $S(n)$ = speedup
- n = número de processadores
- **$E = (S(n) / n) * 100$**

Speedup máximo



Speedup

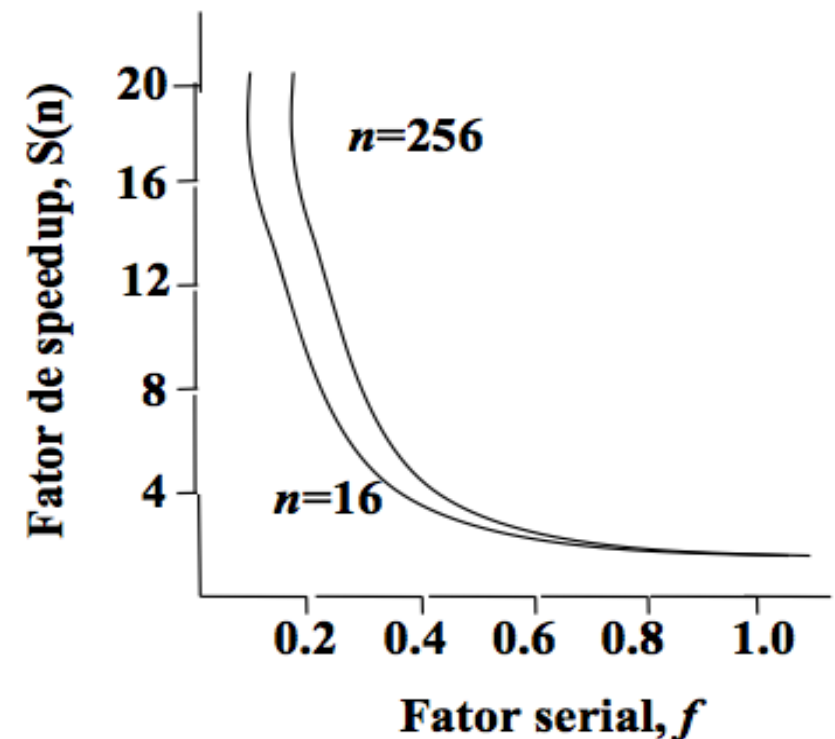
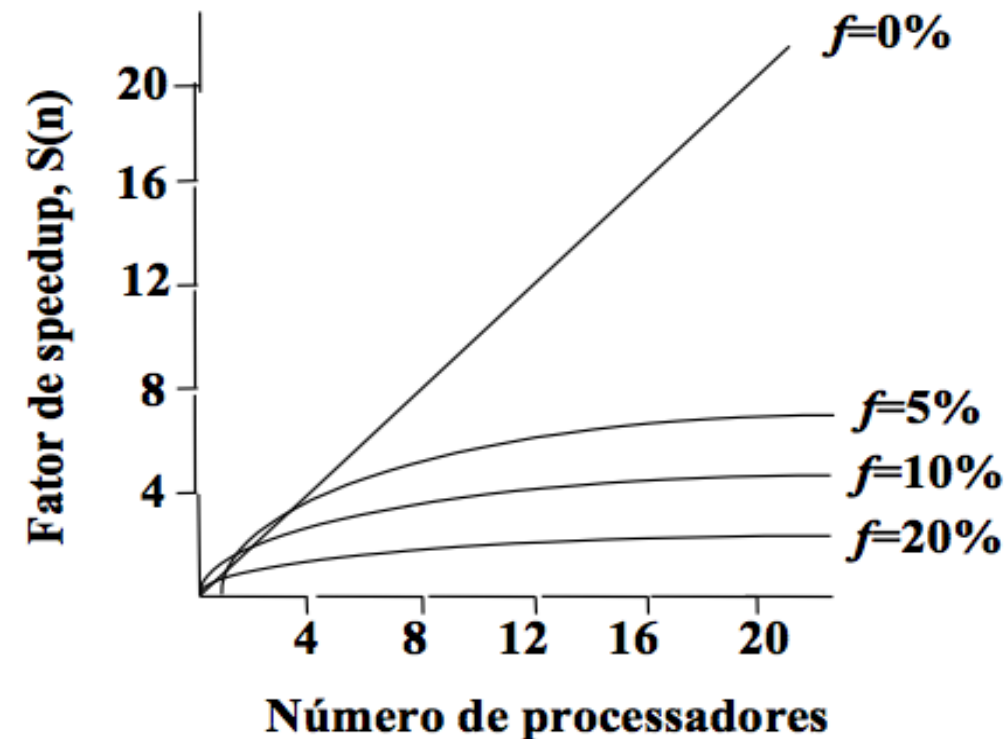
- $S(n) > n$ (superlinear)
 - Algoritmo sequencial sub-ótimo
 - Característica particular da arquitetura da máquina paralela
- $S(n) < n$ (sub-ótimo)
 - Lei de Amdahl
 - Sobrecarga do paralelismo

Amdahl's law

- Considera que o tamanho do problema é fixo
 - Speedup é limitado pela fração serial
-
- $\text{Speedup} = 1 / (s + p/N)$
 - s = fração sequencial
 - p = fração paralela
 - N = número de processadores
 - $s + p = 1$
 - Max speed up = $1/s$

Amdahl's law

- Mesmo com número infinito de processadores a aceleração é limitada a $1/s$



Escalabilidade

- Escalabilidade de hardware ou de arquitetura
 - Aumento do tamanho do sistema impacta no desempenho
 - Facilidade de agregar processadores
- Escalabilidade do algoritmo paralelo
 - Algoritmo pode suportar um aumento do tamanho do problema
 - Exemplo: adição de matrizes: duplica o tamanho da matriz, duplica o número de passos
 - Exemplo: multiplicação de matrizes: duplica o tamanho da matriz, quadruplica o número de passos
 - Exemplo: aumentar a precisão do tempo

Lei de Gustafson (1988)

- Análise da Lei de Amdahl considerando escalabilidade
- Considera que o tempo de execução paralela é fixo, assim como t_f
 - Parte serial é fixa sendo independente da carga
 - **“pode-se resolver problemas maiores no mesmo intervalo de tempo”**
- $\text{Speed up} = s + N_p$

Lei de Gustafson (1988)

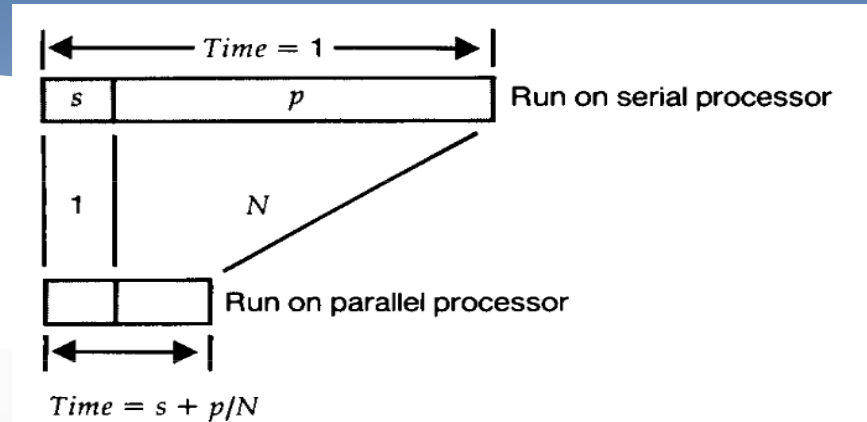


FIGURE 2a. Fixed-Sized Model for $Speedup = 1/(s + p/N)$

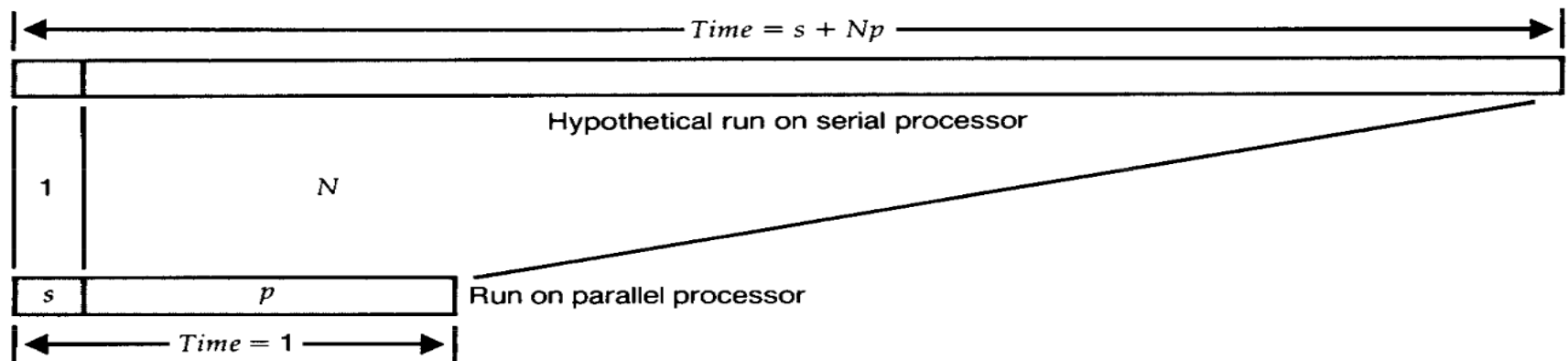


FIGURE 2b. Scaled-Sized Model for $Speedup = s + Np$

Considerações finais

- Teoricamente, o speedup nunca pode exceder o número de elementos de processamento p
- Na prática ocorre o fenômeno conhecido como superlinear speedup
 - O trabalho realizado por um algoritmo sequencial é maior que sua formulação paralela
 - Características de hardware (exemplo: cache)
- Somente um sistema paralelo ideal contendo p elementos de processamento pode fornecer um speedup igual a p
- Na prática não é atingido pois os elementos de processamento não dedicam 100% de tempo para a execução do programa