

Arquiteturas Paralelas e Distribuídas

# Análise de desempenho e depuração de programas paralelos

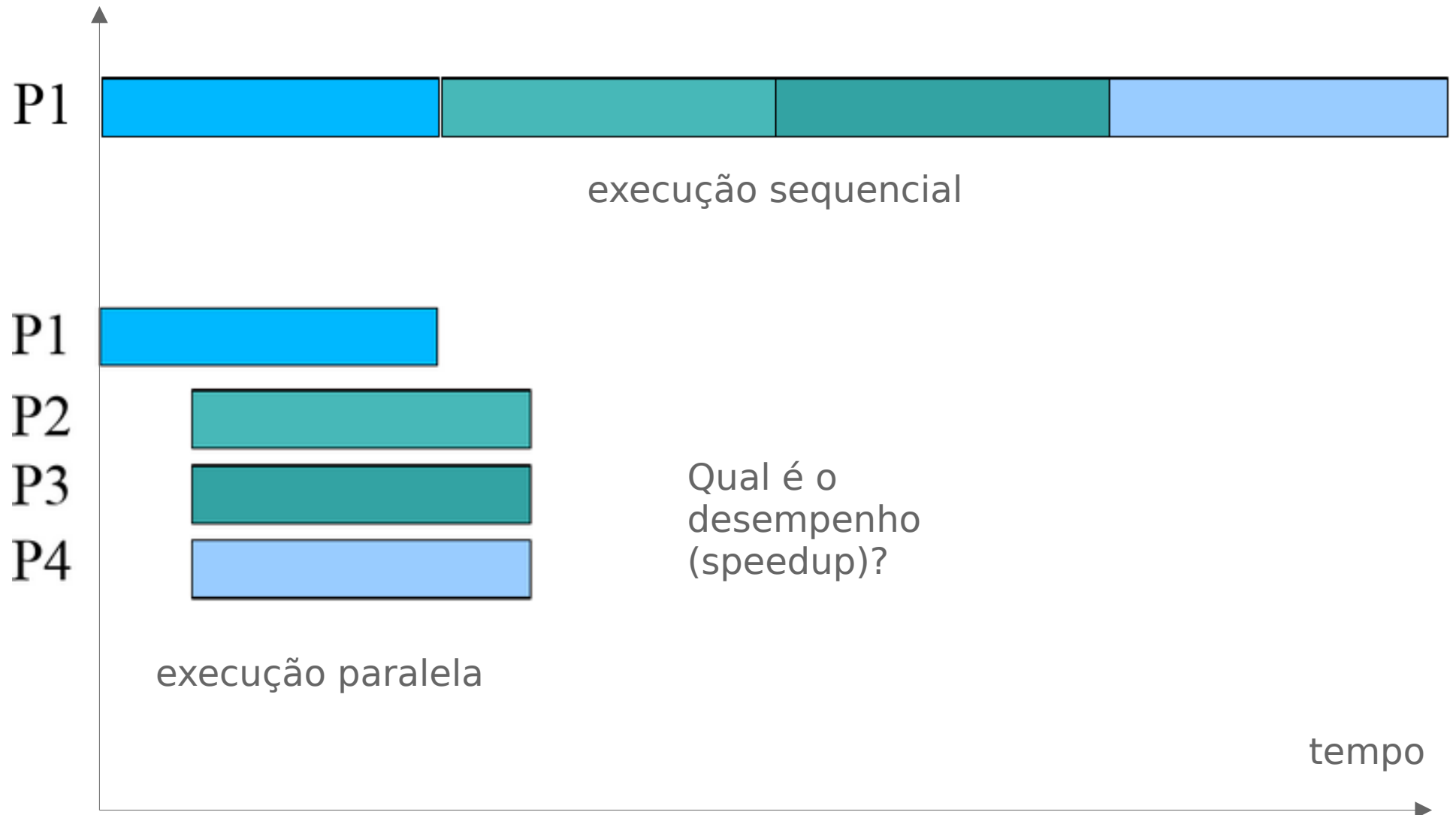
Eduardo Furlan Miranda  
2025-02-01

Baseado em: CHARÃO, A. S. Análise de desempenho de programas paralelos. 2011.

- Objetivo
  - Introduzir conceitos usados na análise de desempenho de programas paralelos
- Conceitos
  - Aceleração
  - Eficiência
  - Escalabilidade

# Execução sequencial X paralela

3/15

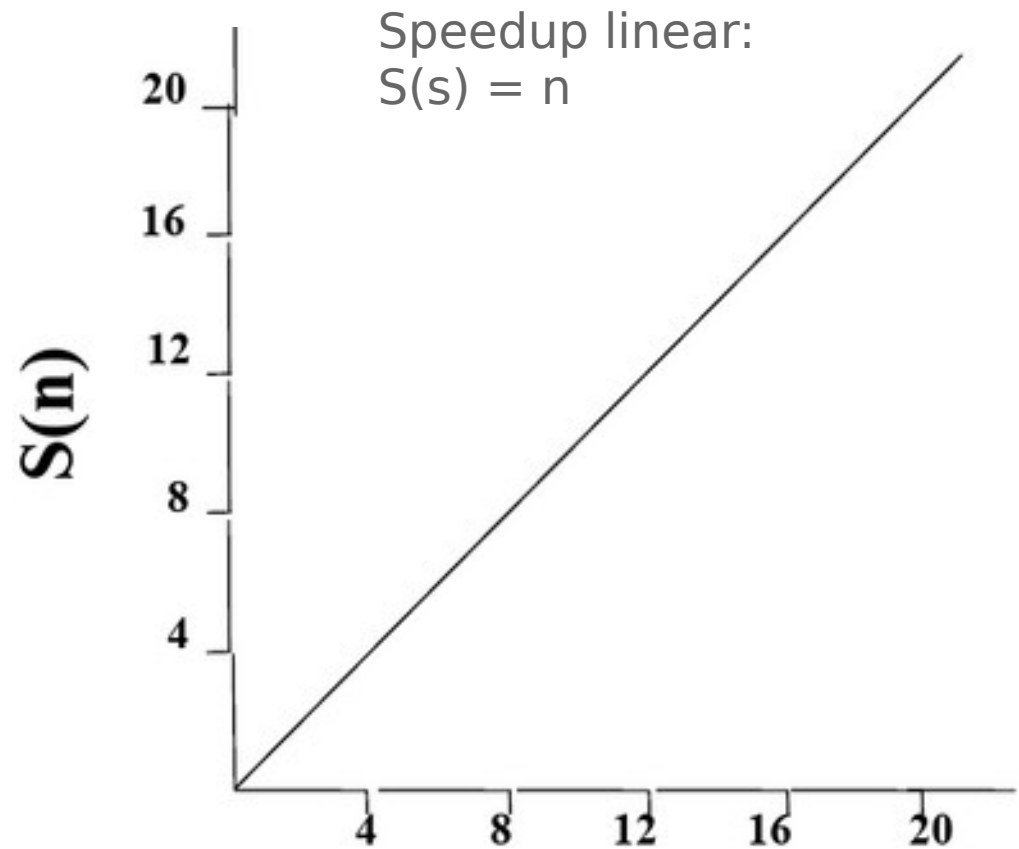


# Fator de aceleração (speedup)

4/15

$$S(n) = \frac{\text{tempo de execução serial}}{\text{tempo de execução paralela}} = \frac{t_s}{t_p}$$

ideal para máquina  
paralela com n  
processadores/núcleos



$$E = \frac{\text{tempo de execução serial}}{t_{\text{exec. paralela}} \times n^{\circ} \text{ de processadores}} = \frac{t_s}{t_p \times n}$$

$$E = \frac{S(n)}{n} \times 100\%$$

A eficiência fornece a fração de tempo que os processadores estão sendo utilizados

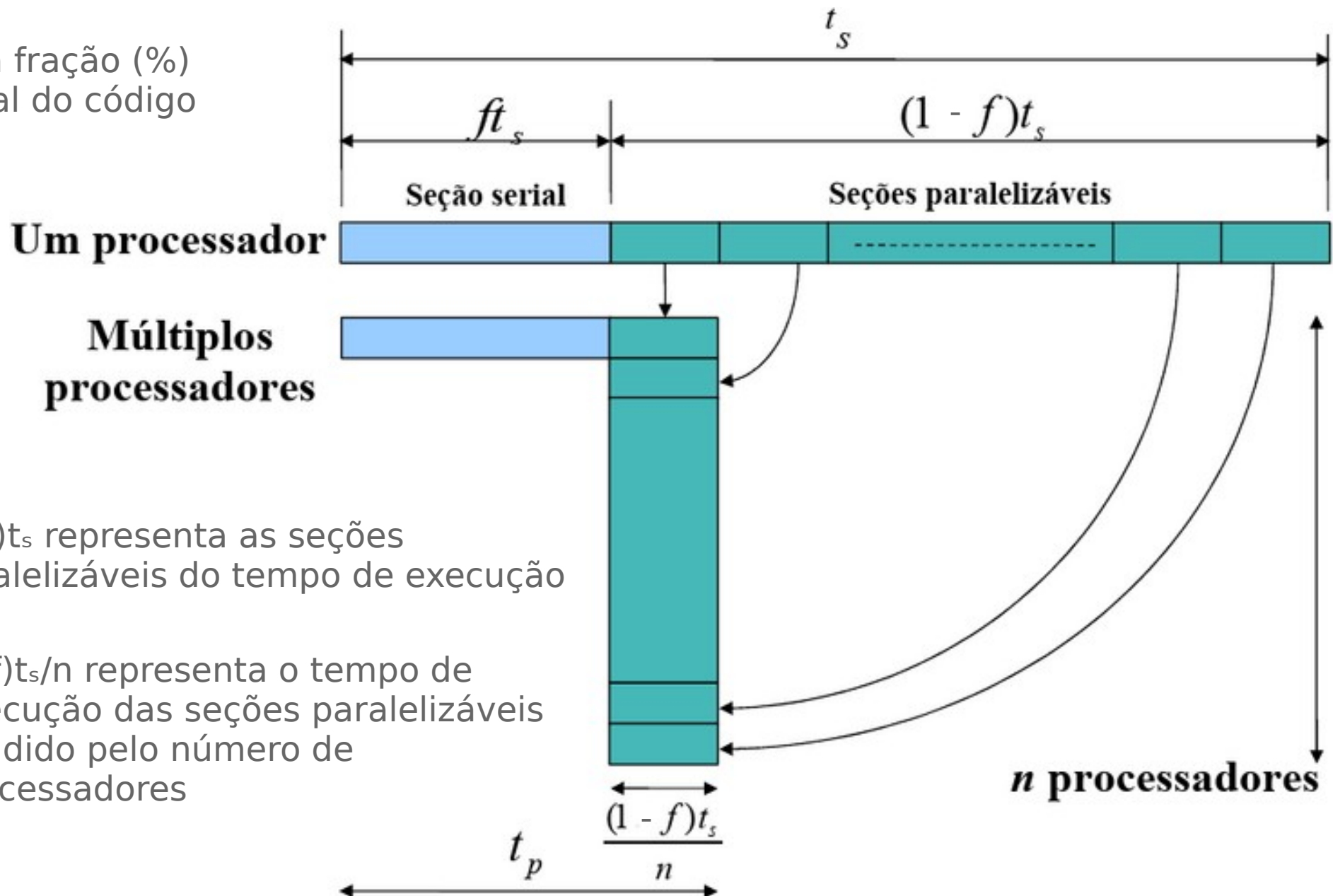
# Speedup

- $S(n) > n$  (superlinear)
  - O desempenho do programa paralelo é mais do que  $n$  vezes mais rápido do que o programa sequencial
  - O algoritmo sequencial original não é a maneira mais eficiente de resolver o problema
  - A arquitetura da máquina paralela possui características específicas, como memória extra ou um sistema de cache mais eficiente
    - Ex.: se cada processador em um sistema paralelo tiver sua própria memória, o conjunto total disponível pode ser maior do que em um sistema sequencial
      - Isso pode reduzir a necessidade de troca de dados entre a memória principal e a memória cache, resultando em um speedup  $>$  que o esperado
- $S(n) < n$  (sub-ótimo)
  - Lei de Ahmdal
  - Sobrecarga do paralelismo (ex.: comunicação)

# Speedup máximo

7/15

$f$  é a fração (%)  
serial do código



# Lei de Amdahl (1967)

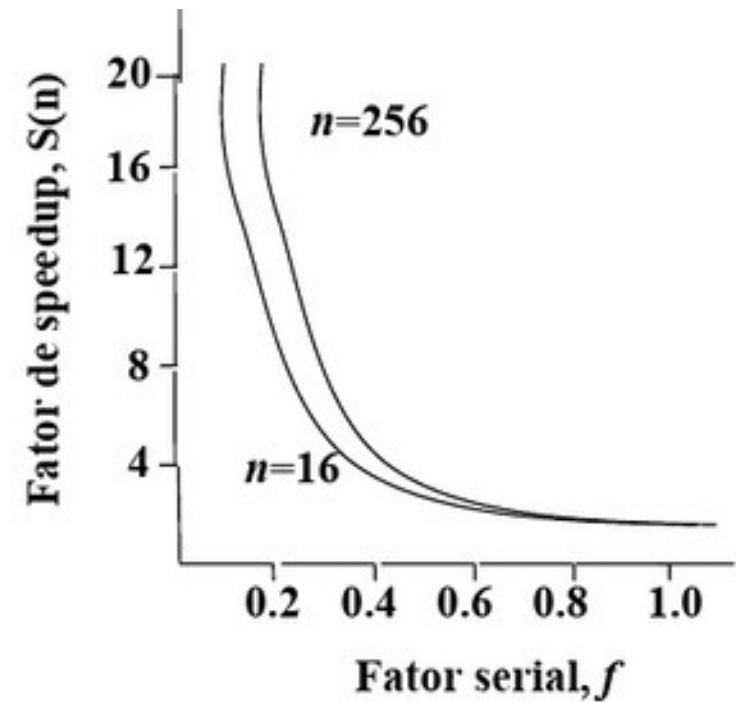
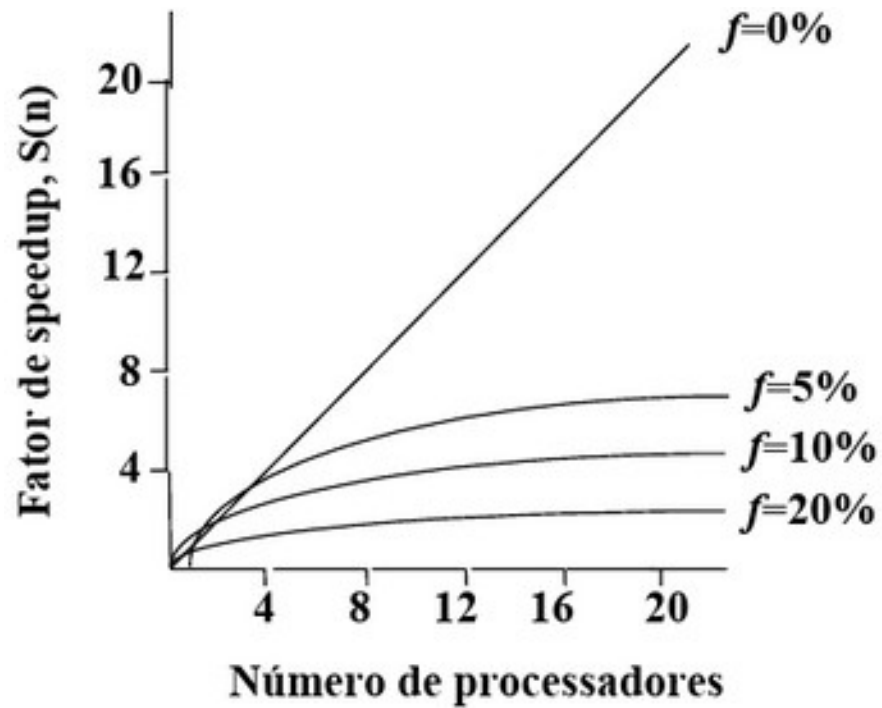
- O fator de speedup é dado por

$$S(n) = \frac{t_s}{ft_s + (1-f)\frac{t_s}{n}} = \frac{n}{1 + (n-1)f}$$

- Speedup limitado pela fração serial
- Considera que o tamanho do problema é fixo
- Speedup máximo:  $S_{\max} = 1 / f$



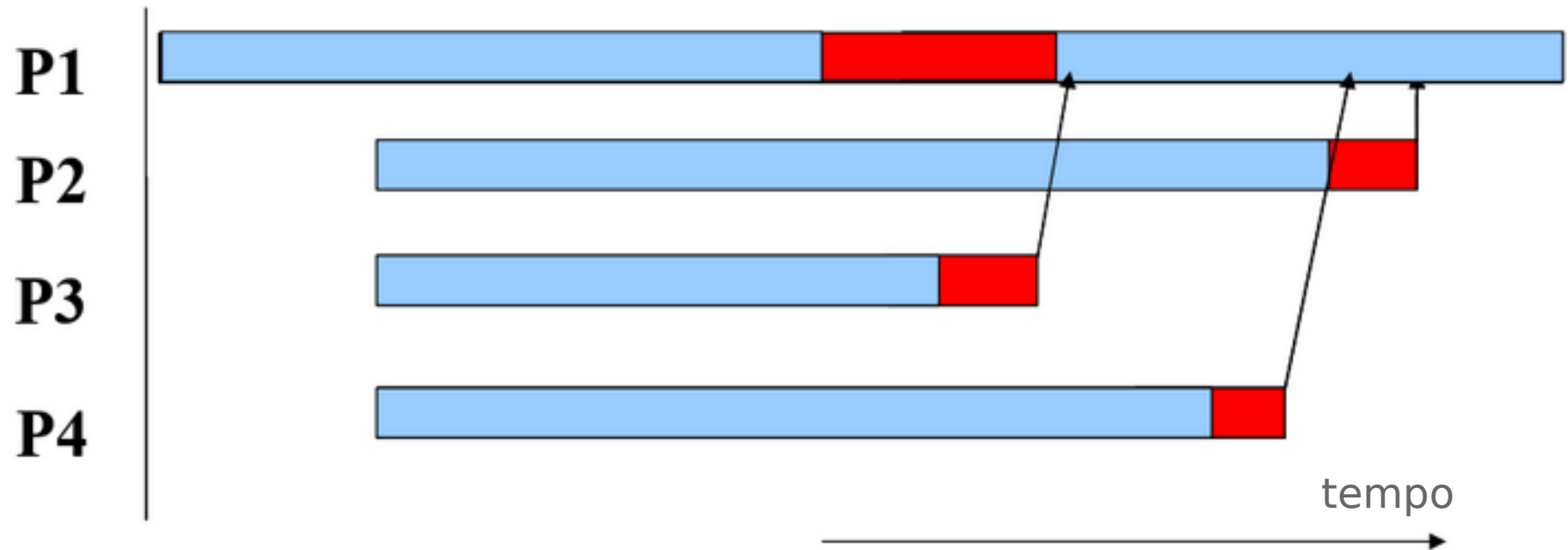
# Lei de Amdahl



- Mesmo com número infinito de processadores o speedup é limitado a  $1/f$
- Ex.: para  $f = 5\%$ , o speedup máximo é  $1/0,05 = 20$ , independente de  $n$

# Sobrecargas na execução paralela

10/15



- Em vermelho, tempos gastos à mais, além do que foi gasto no cálculo sequencial
- Ex.: ações ou esperas para comunicação, sincronização, criação de threads/processos

# Escalabilidade

- Escalabilidade de hardware ou de arquitetura
  - Aumento do tamanho do sistema aumenta o desempenho
  - Facilidade de agregar processadores
- Escalabilidade do algoritmo paralelo
  - Problema de tamanho fixo
  - Problema de tamanho variável

- Escalabilidade do algoritmo
  - Algoritmo pode suportar um aumento do tamanho do problema com um aumento baixo e limitado dos passos de computação
    - Ex. 1: adição de matrizes - duplica o tamanho da matriz, duplica número de passos
    - Ex. 2: multiplicação de matrizes duplica o tamanho da matriz, quadruplica número de passos

# Lei de Gustafson (1988)

$$\text{Speedup} = f + P(1-f)$$

onde

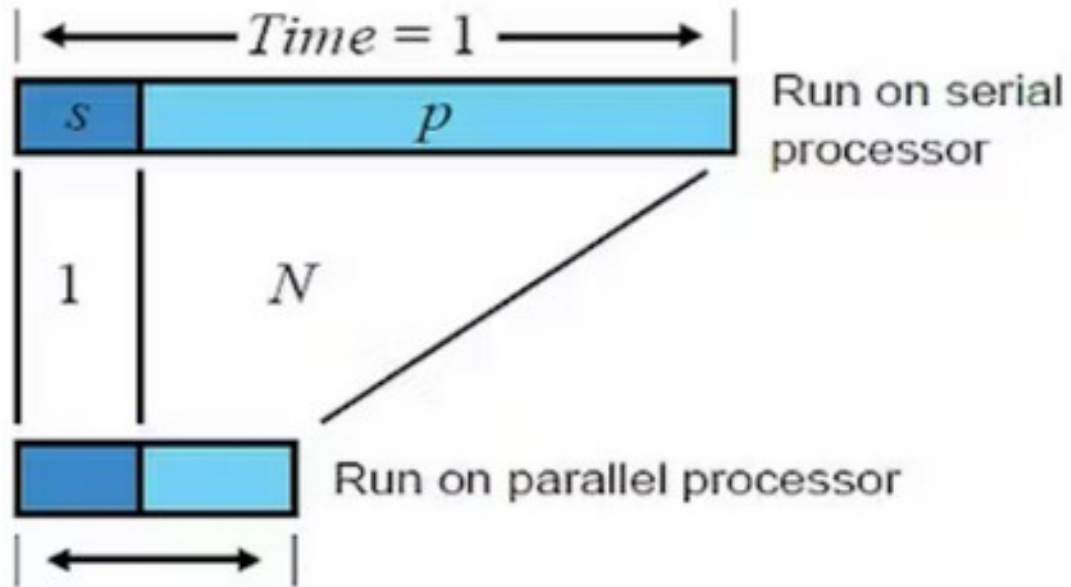
$f$  = fração serial do problema escalonado (a parte que não pode ser paralelizada)

$(1-f)$  = fração paralela do problema escalonado

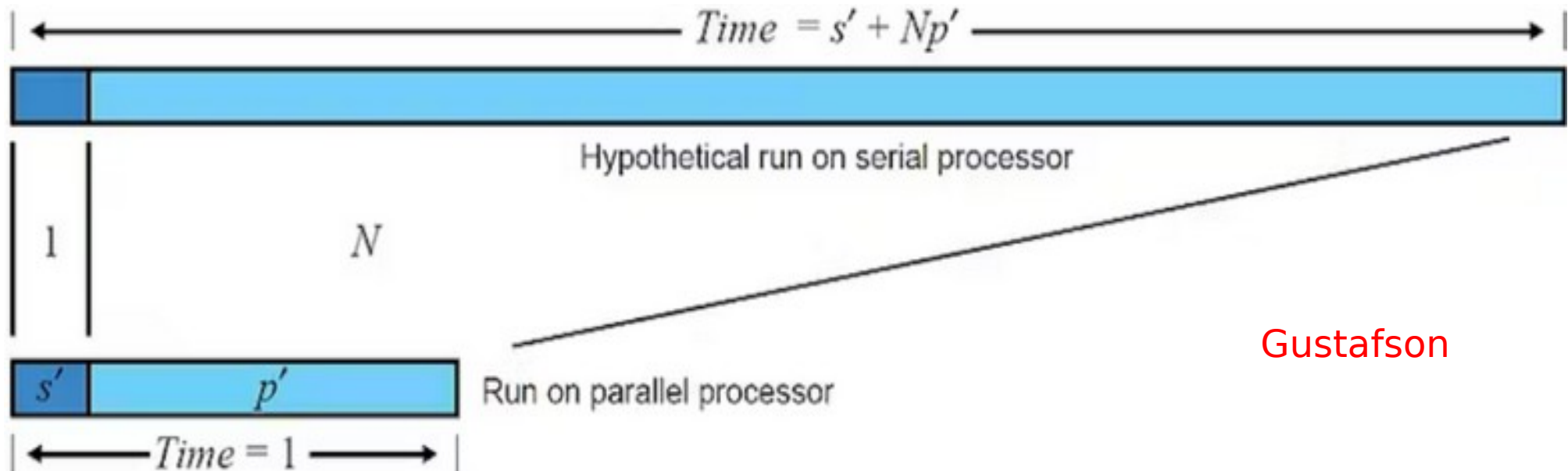
$P$  = quantidade de processadores

- Reavalia a lei de Amdahl sob o ponto de vista da escalabilidade
- O tamanho do problema aumenta com o número de processadores
  - Pode-se resolver problemas maiores no mesmo intervalo de tempo
- Considera que o tempo de execução paralelo é fixo

## Amdahl



- $s'$  = parte serial
- $p'$  = parte paralela
- $s'$  corresponde à fração serial ( $f$ )
- $p'$  corresponde à fração paralela ( $1-f$ )
- $N$  = quantidade de processadores



## Gustafson

- Na análise de desempenho é necessário conhecer os tempos de execução
- Speedup, Eficiência e Escalabilidade são conceitos fundamentais na análise de desempenho
- Lei de Amdahl: o speedup máximo é limitado pela fração serial do código, em um **problema fixo**
- Lei de Gustafson: o speedup cresce com a **escala do problema** junto com o número de processadores
- Sobrecargas e escalabilidade: a execução paralela tem sobrecargas, e a escalabilidade do algoritmo suporta aumento no tamanho do problema