

## Processamento Paralelo

**multiprocessador** sistema com mais de um processador

**process level parallelism** programas executam independentemente num multiprocessador (servidores Unix)

**processo**  $\triangleq$  programa em execução: estado + espaço de endereçamento

**thread level parallelism** mais de uma linha de execução dentro de um processo

**thread**  $\triangleq$  processo leve, com registradores, PC, pilha privativos, threads compartilham espaço de endereçamento do processo

**data level parallelism** mesma operação aplicada sobre um conjunto de dados (unidades vetoriais)

**instruction level parallelism** paralelismo entre instruções (Pentium)

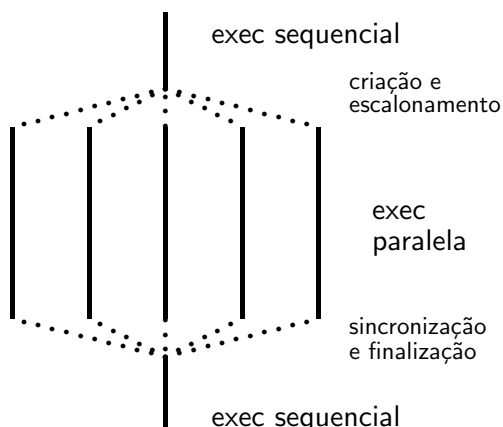
**granularidade** **finas**: ILP, DLP; **grossa**: processo, thread

## Escrever programas paralelos é DIFÍCIL!!

1. Escalonamento: que processador executa qual tarefa quando alocar trabalho, iniciar e terminar execução paralela são operações custosas que **serializam** execução
2. balanceamento de carga: todos processadores **devem** executar mesmo tanto de trabalho proc. com mais carga atrasa a todos
3. sincronização: mal necessário com threads, para garantir correteude sob modelo sequencial de execução **serializa** execução
4. comunicação: cooperação necessita comunicação trablho adicional para “empacotar”, enviar, receber, “desempacotar” mensagens enquanto comunica, não computa

Execução paralela num multiprocessador **deve** prover melhor desempenho e eficiência que execução serial num uniprocessador.

## Execução paralela



## Lei de Amdahl (i)

$$\begin{aligned} \text{Tempo de execução após melhoria} = & \\ & (\text{tempo de execução afetado} / \text{quanto melhorou}) \\ & + \text{tempo de execução não-afetado} \\ & \text{este termo é muito importante} \end{aligned}$$

Exemplo:

programa executa em 100s, multiplicações consomem 80% do tempo total. Quanto devo melhorar o circuito multiplicador se quero tempo total em 20s ?

$$20 = 80/n + 20 \quad \text{erm...}$$

Idem, se quero tempo total em 40s ?

$$40 = 80/n + 20$$

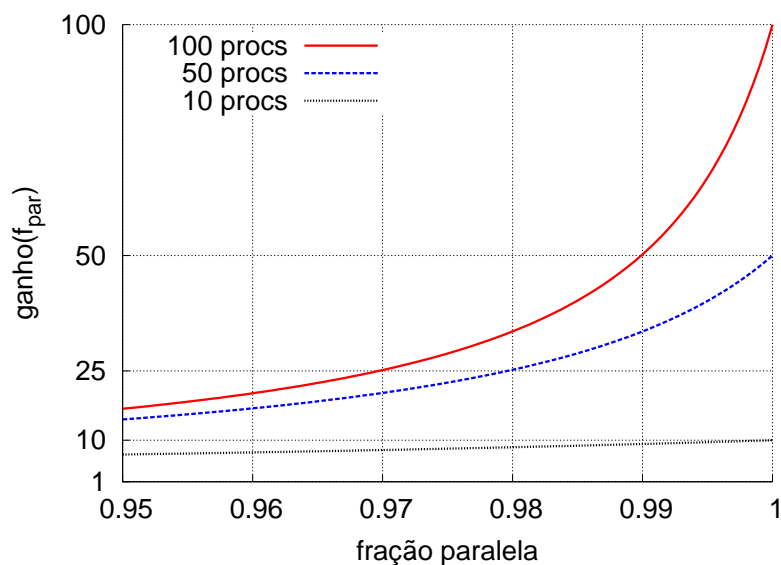
## Lei de Amdahl (ii)

$$\begin{aligned} \text{Ganho}_{\text{total}} &= \frac{\text{Tempo}_{\text{orig}}}{\text{Tempo}_{\text{melhor}}} \\ &= \frac{1}{(1 - \text{Frac}_{\text{melhor}}) + (\text{Frac}_{\text{melhor}} / \text{Ganho}_{\text{melhor}})} \end{aligned}$$

Desejo ganho de 90x ao executar com 100 processadores. Qual a fração da computação que pode ser sequencial?

$$\begin{aligned} 90 &= \frac{1}{(1 - f) + f/100} \\ f &= 0,999 \\ (1 - f) &= 0,1\% \quad \text{fração serial} = 0,1\% \end{aligned}$$

## Lei de Amdahl (iii)



## Escalabilidade (i)

Programa soma 10 escalares (serial) e então duas matrizes 10x10 com  $P_{10}$  e com  $P_{100}$ . Desempenho proporcional ao tempo  $t$  da adição.

$$T(P_1) = 10t + 100t = 110t$$

$$T(P_{10}) = 10t + 100t/10 = 20t$$

$$T(P_{100}) = 10t + 100t/100 = 11t$$

$$G(P_1/P_{10}) = 110t/20t = 5,5 \quad 5,5 \text{ com } P=10$$

$$G(P_1/P_{100}) = 110t/11t = 10 \quad 10 \text{ com } P=100$$

$$E(P_{10}) = G(P_{10})/10 = 5,5/10 = 55\%$$

$$E(P_{100}) = G(P_{100})/100 = 10/100 = 10\%$$

$T$  tempo,  $G$  ganho,  $E$  eficiência =  $G(P_n)/n$

UFPR Centro de Informática

7

## Escalabilidade (ii)

Programa soma 10 escalares (serial) e então duas matrizes 100x100 com  $P_{10}$  e com  $P_{100}$ . Desempenho proporcional ao tempo  $t$  da adição.

$$T'(P_1) = 10t + 10.000t = 10.010t$$

$$T'(P_{10}) = 10t + 10.000t/10 = 1.010t$$

$$T'(P_{100}) = 10t + 10.000t/100 = 110t$$

$$G'(P_1/P_{10}) = 10.010t/1.010t = 9,9 \quad 9,9 \text{ com } P=10$$

$$G'(P_1/P_{100}) = 10.010t/110t = 91 \quad 91 \text{ com } P=100$$

$$E'(P_{10}) = G'(P_{10})/10 = 9,9/10 = 99\%$$

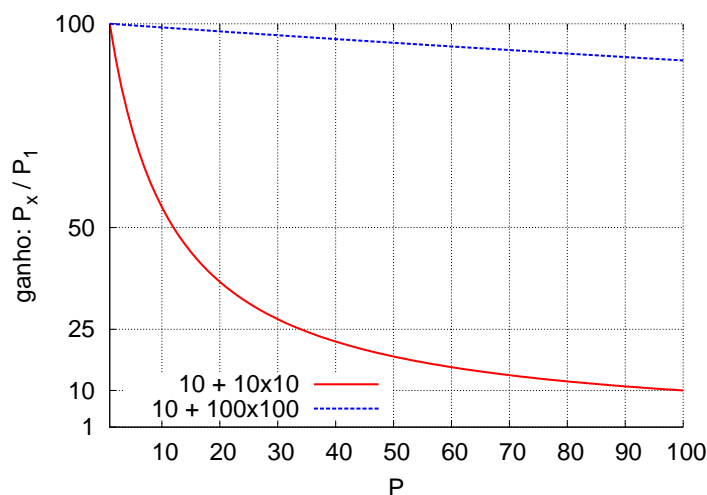
$$E'(P_{100}) = G'(P_{100})/100 = 91/100 = 91\%$$

UFPR Centro de Informática

8

## Escalabilidade (iii)

Programa soma 10 escalares (serial) e duas matrizes NxN com  $P_{1..100}$



UFPR Centro de Informática

9

## Escalabilidade (iv)

- **Strong Scaling:** mede-se ganho **sem** aumentar conjunto de dados **mesmo problema em menor tempo**  
Para conjunto de trabalho  $|M|$ ,  $\text{mem}/\text{proc} = M/P$
- **Weak Scaling:** mede-se ganho aumentando problema **proporcionalmente a P**  
**problema maior no mesmo tempo**  
Para conjunto de trabalho  $|M|$ ,  $\text{mem}/\text{proc} = M$

## Balanceamento de Carga (i)

Programa soma 10 escalares (serial) e então duas matrizes  $100 \times 100$  com  $P_{10}$  e com  $P_{100}$ . Desempenho proporcional ao tempo  $t$  da adição.  $P^0$  executa 2% do trabalho,  $P^{1..99}$  dividem resto. Como afeta ganho?

$$T''(P_{100}) = 10t + \max(9800t/99, 200t/1) = 210t$$

$$G''(P_1/P_{100}) = 10.010t/210t = 48 \ll 91 \approx 1/2$$

$P^0$  executa 5% do trabalho,  $P^{1..99}$  dividem resto. Como afeta ganho?

$$T'''(P_{100}) = 10t + \max(9500t/99, 500t/1) = 510t$$

$$G'''(P_1/P_{100}) = 10.010t/510t = 20 \ll 91 \approx 1/5$$

## Balanceamento de Carga (ii)

Mesmo problema,  $P_0$  executa mais trabalho,  $P_{1..99}$  dividem resto

