



# Programação de Sistemas Paralelos e Distribuídos

(206610)

## AULA 1

## Conceitos Básicos 1

**Professor: Luiz Augusto Laranjeira**  
[luiz.laranjeira@gmail.com](mailto:luiz.laranjeira@gmail.com)

Material originalmente produzido pelo Prof. Jairo Panetta (ITA) e adaptado para a FGA pelo Prof. Laranjeira.



- Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar: *Introduction to Parallel Computing*, Addison Wesley, 2003
  - Princípios Gerais de Paralelismo; Visão Algoritmica.
- John L. Hennessy, David A. Patterson: *Computer Architecture: A Quantitative Approach*, 4th edition, Morgan Kaufmann, 2007.
  - Bíblia de arquitetura de computadores
- Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, Andy White: *Sourcebook of Parallel Computing*, Morgan Kaufmann, 2003.
  - Um dos melhores livros em aplicações.
- David E. Culler, Jaswinder Pal Singh, Anoop Gupta: *Parallel Computer Architecture, a Hardware/Software Approach*, Morgan Kaufmann, 1999.
  - Ênfase em hardware e seus impactos em software.
- Michael Wolfe: *High Performance Compilers for Parallel Computing*, Addison-Wesley, 1996.
  - Técnicas de otimização e de compilação para paralelismo.
- Livros e padrões de OpenMP, MPI e CUDA (citados nas respectivas aulas)



- Definição de Paralelismo
- Níveis de Paralelismo
- Métricas de Desempenho Paralelo
- Lei de Amdahl
- Necessidade e Utilidade de Paralelismo
- Lei de Moore
- Memory Wall, Power Wall
- Cray no IME



Processamento Paralelo é a realização simultânea de múltiplas partes de uma única computação para reduzir o tempo total de sua execução



- Copiar (“xerox”) documento da ordem de 10.000 páginas utilizando
  - 1 copiadora
  - 2 copiadoras
  - ...
- Cavar uma vala horizontal de 10km de comprimento por 2m de diâmetro utilizando
  - 1 trabalhador
  - 2 trabalhadores
  - ...
- Escrever um arquivo em disco utilizando
  - 1 servidor de arquivos
  - 2 servidores de arquivos
  - ...
- Executar um algoritmo numérico para previsão do tempo utilizando
  - 1 CPU
  - 2 CPUs
  - ...

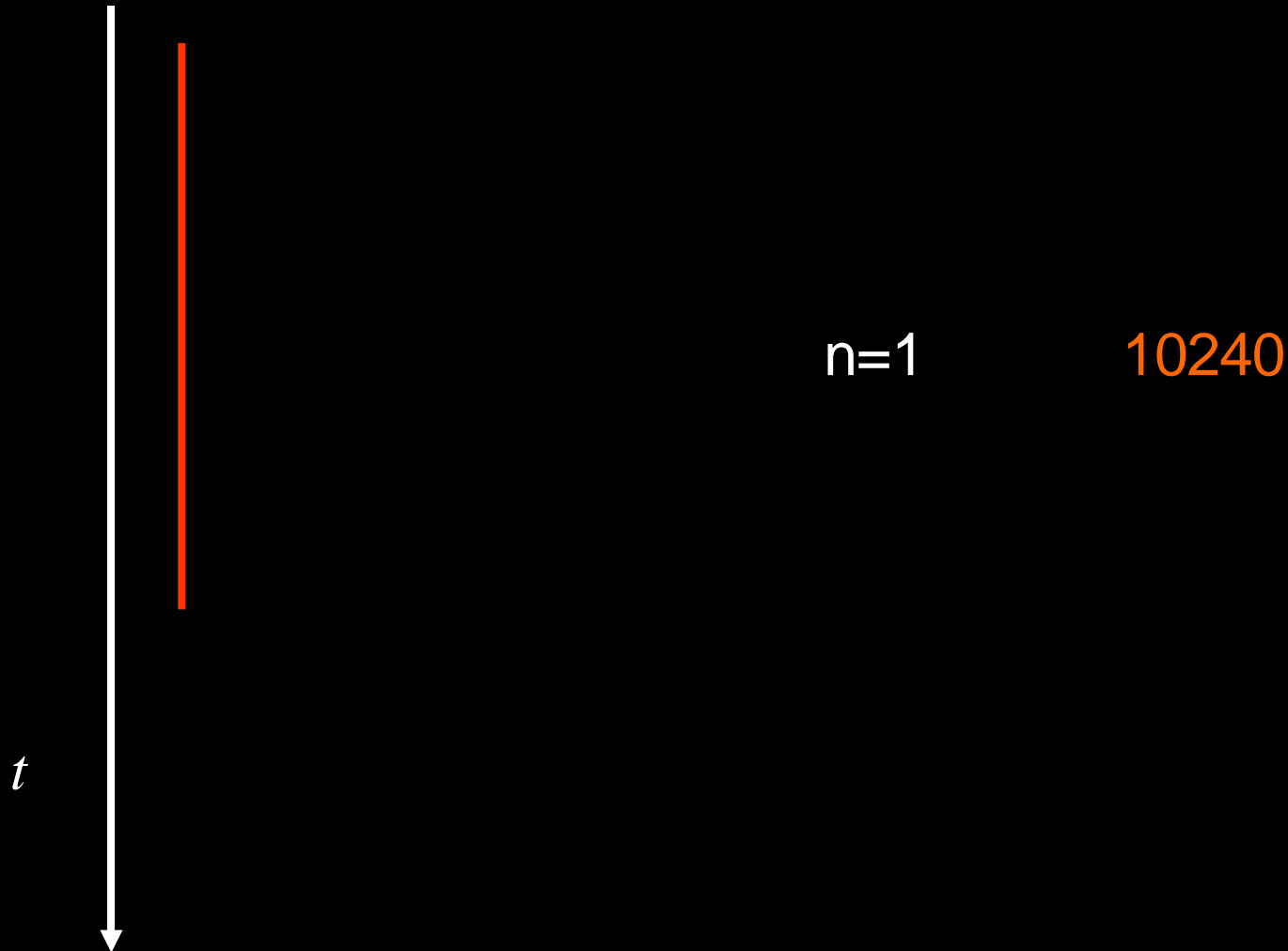


- Requer a replicação (total ou parcial) de entidades trabalhadoras
  - múltiplas copiadoras, múltiplos trabalhadores, múltiplos servidores de arquivos, múltiplas CPUs
- Requer alterar a forma de realizar a tarefa para usufruir da replicação
- Ao replicar  $n$  entidades deseja-se dividir o tempo de execução da tarefa por  $n$ 
  - **máximo teórico !!! (máximo possível???)**



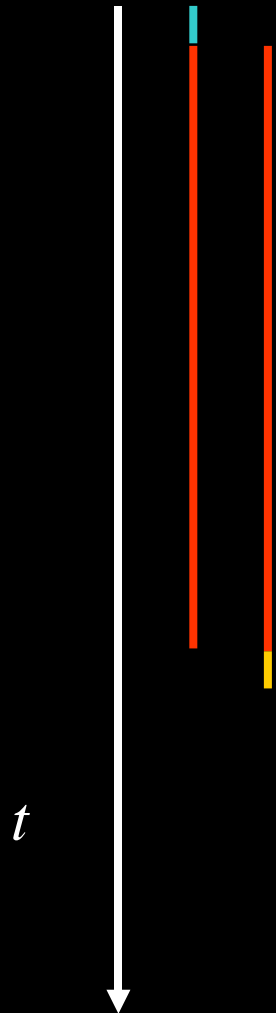
- Suponha:
  - original entregue em ordem crescente das páginas numeradas;
  - cópia deve ser entregue da mesma forma
  - uma única pessoa divide o original em partes iguais a serem copiadas, entrega para os copiadores e recolhe as cópias
  - cada copiadora copia uma única parte
  - 1 segundo para copiar uma página
  - 5 segundos para dividir o original em dois blocos
  - 5 segundos para juntar dois blocos de páginas copiadas
- Estime o tempo para realizar a tarefa utilizando
  - 1, 2, 4, 8, 16, 32 e 64 copiadoras

# Tempo com 1 copiadora



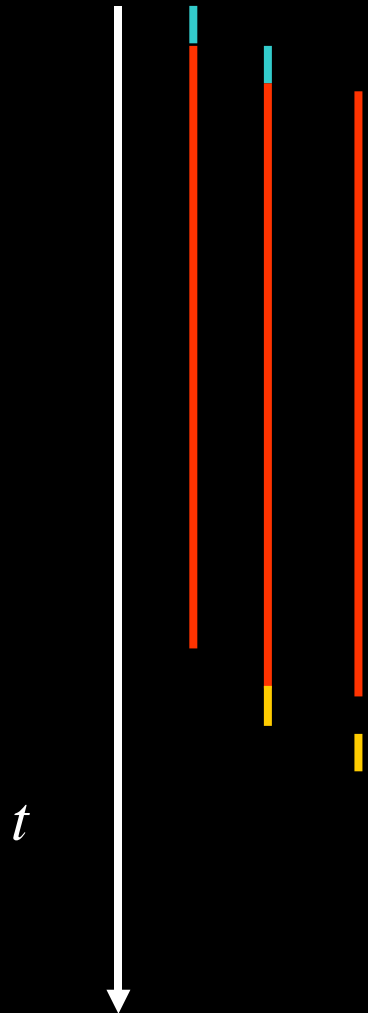


# Tempo com 2 copiadoras



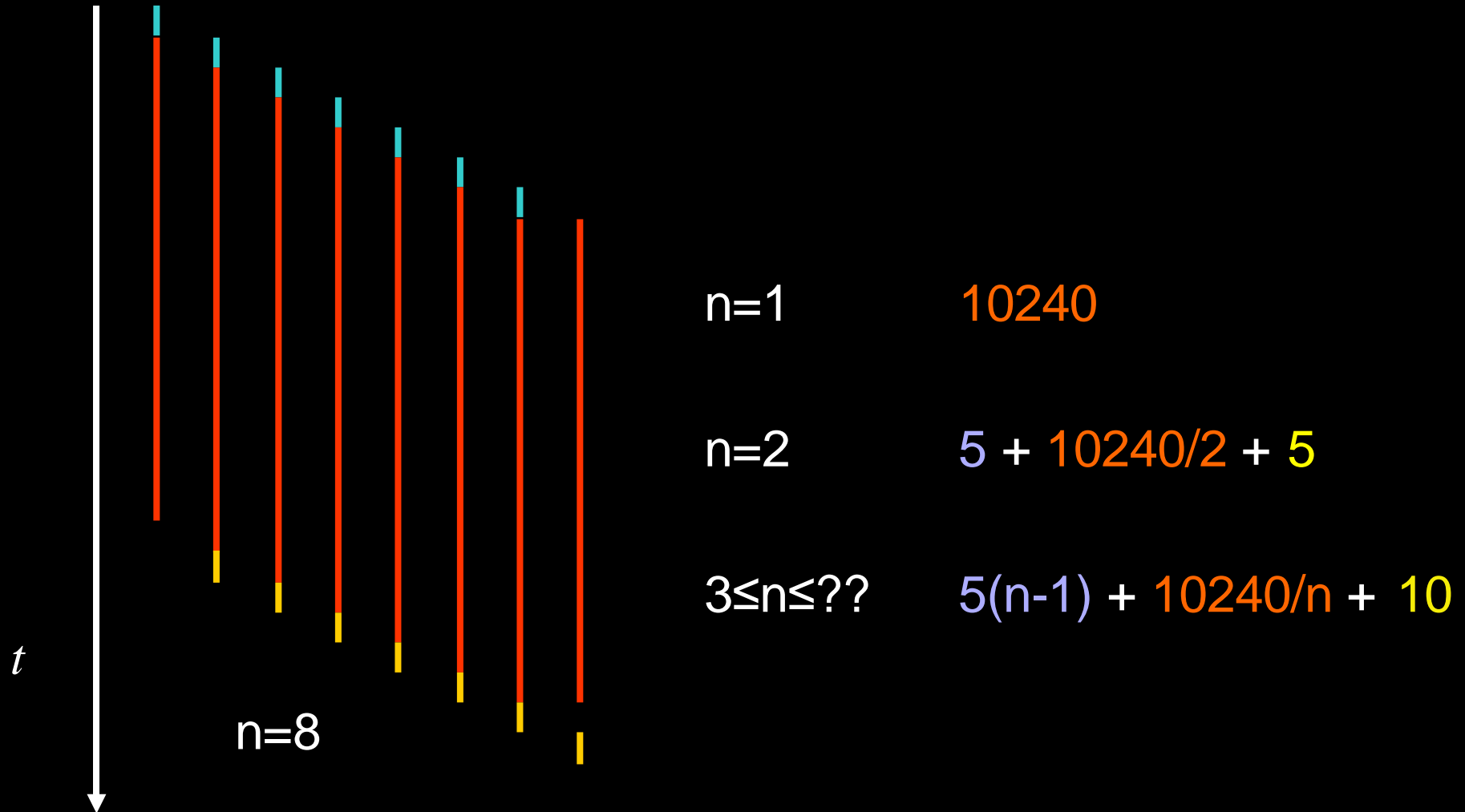
$n=2$

$$5 + 10240/2 + 5$$



$n=3$

$$5 \cdot 2 + 10240/3 + 10$$





Hipótese: tempo total = tempo de entrega sequencial das partes + tempo de copiar uma parte + tempo de recolher duas últimas partes

| Máquinas | Tempo (s) |        |        |       | Ganho |
|----------|-----------|--------|--------|-------|-------|
|          | dividir   | copiar | juntar | total |       |
| 1        | 0         | 10240  | 0      | 10240 | 1,00  |
| 2        | 5         | 5120   | 5      | 5130  | 2,00  |
| 4        | 15        | 2560   | 10     | 2585  | 3,96  |
| 8        | 35        | 1280   | 10     | 1325  | 7,73  |
| 16       | 75        | 640    | 10     | 725   | 14,12 |
| 32       | 155       | 320    | 10     | 485   | 21,11 |
| 64       | 315       | 160    | 10     | 485   | 21,11 |

Para 64 máquinas  $T_d > T_c$  ( $315 > 160$ )  
 Não há ganho com respeito a 32 máquinas



- Como utilizar eficientemente 1024 copiadoras?
  - Alterando os mecanismos de distribuição de dados e de coleta de resultados
- O procedimento (algoritmo) precisaria ser mudado de acordo com o volume de paralelismo para continuar a se ter um ganho crescente
  - Pois muda o gargalo da tarefa
- E se uma copiadora for 1% mais lenta que as outras?
  - Típico de grande número de máquinas iguais
- E se uma copiadora quebrar?
  - Tolerância a falhas é desejável; às vezes, imprescindível



- Exemplo atribuído ao Prof. Siang W. Song (IME/USP)
- Similar ao exemplo anterior. Assuma terreno demarcado, número crescente de trabalhadores equipados e um capataz.
  - Quais são os fatores que impedem redução “ótima” do tempo de execução com o aumento do número de trabalhadores?
- E se a vala for vertical?
- Pequenas variações do problema podem causar grandes variações no paralelismo
- Há problemas inerentemente sequenciais
  - alterar 1 bit na memória



- Definição de Paralelismo
- **Níveis de Paralelismo**
- Métricas de Desempenho Paralelo
- Lei de Amdahl
- Necessidade e Utilidade de Paralelismo
- Lei de Moore
- Memory Wall, Power Wall
- Cray no IME



- Paralelismo ao nível de bits
  - BLP: *Bit Level Parallelism*
  - Acelera a execução de uma instrução aumentando a largura do *datapath* (microprocessadores de 4, 8, 16, 32 e 64 bits)
  - Muito explorado até meados da década de 80; esgotado
- Paralelismo ao nível de instruções
  - ILP: *Instruction Level Parallelism*
  - Execução simultânea de múltiplas instruções ou trechos: *RISC*, *VLIW*, *Superscalar*
  - Muito explorado até meados da década de 90; praticamente esgotado
- Paralelismo ao nível de tarefas
  - TLP: *Thread Level Parallelism*
  - Múltiplos fluxos de execução simultâneos: Multi-core chips, Multithreading, etc...
- Curso concentrado em TLP
  - Alguns casos de ILP

Culler, Cap. 1





- Definição de Paralelismo
- Níveis de Paralelismo
- Métricas de Desempenho Paralelo
- Lei de Amdahl
- Necessidade e Utilidade de Paralelismo
- Lei de Moore
- Memory Wall, Power Wall
- Cray no IME



- Mede o ganho (no tempo de execução) em utilizar  $p$  processadores

$$S(p) = \frac{T(1)}{T(p)}$$

- Teoricamente,  $S(p) \leq p$ , isto é,  $S(p)$  é, no máximo, igual a  $p$ .
- Na prática, características do processador eliminam este limite (speed-up superlinear)
  - Há casos em que  $S(p) \geq p$



- Mede a eficiência em utilizar  $p$  processadores

$$E(p) = \frac{T(1)}{p \times T(p)}$$

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



- Definição de Paralelismo
- Níveis de Paralelismo
- Métricas de Desempenho Paralelo
- Lei de Amdahl
- Necessidade e Utilidade de Paralelismo
- Lei de Moore
- Memory Wall, Power Wall
- Cray no IME



Seja  $f$  a fração do tempo de execução sequencial de um programa ( $0 < f \leq 1$ ) que é mantida sequencial após a paralelização.

Então, o ganho (*speed-up*) máximo é

$$S(p) = \frac{1}{f}$$

para qualquer número de processadores.

*Gene M. Amdahl, "Validity of the single processor approach to achieving large scale computing capabilities", AFIPS spring joint computer conference, 1967*



| $f$ (%) | $f$   | $S(p) = 1/f$                           |
|---------|-------|----------------------------------------|
| 0       | 0     | $\infty$ (qdo $p \rightarrow \infty$ ) |
| 0.1     | 0.001 | 1000                                   |
| 1       | 0.01  | 100                                    |
| 10      | 0.1   | 10                                     |
| 20      | 0.2   | 5                                      |
| 50      | 0.5   | 2                                      |
| 80      | 0.8   | 1.25                                   |

$f$  é a fração da computação que não pode ser paralelizada



Sejam:

- $T(1)$  o tempo de execução com 1 processador
- $T(p)$  o tempo de execução com  $n$  processadores
- $f$  a fração de  $T(1)$  que é mantida sequencial em  $T(p)$

$$T_{seq} = fT(1)$$

- $S(p)$  o speed-up ao se usar  $p$  processadores

queremos demonstrar que

$$S(p) \leq \frac{1}{f}$$

(para qualquer número de processadores)



$$T(1) = T_{iseq} + T_{ipar} = fT(1) + (1 - f)T(1)$$

$$T(p) = T_{seq} + T_{par} \quad (\text{teórico}), \quad T_{seq} = fT(1), \quad T_{par} = \frac{(1 - f)T(1)}{S(p)}, \quad \min T_{par} = \frac{(1 - f)T(1)}{p}$$

$S(p)_{max} = p$

Logo,

$$T(p) \geq fT(1) + \frac{(1-f)T(1)}{p}$$

pois  $T_{par}$  não pode ser reduzido por fator superior a  $p$ . Assim,

$$\frac{T(p)}{T(1)} \geq f + \frac{(1-f)}{p} \quad \Rightarrow \quad S(p) \leq \frac{1}{f + \frac{(1-f)}{p}}$$

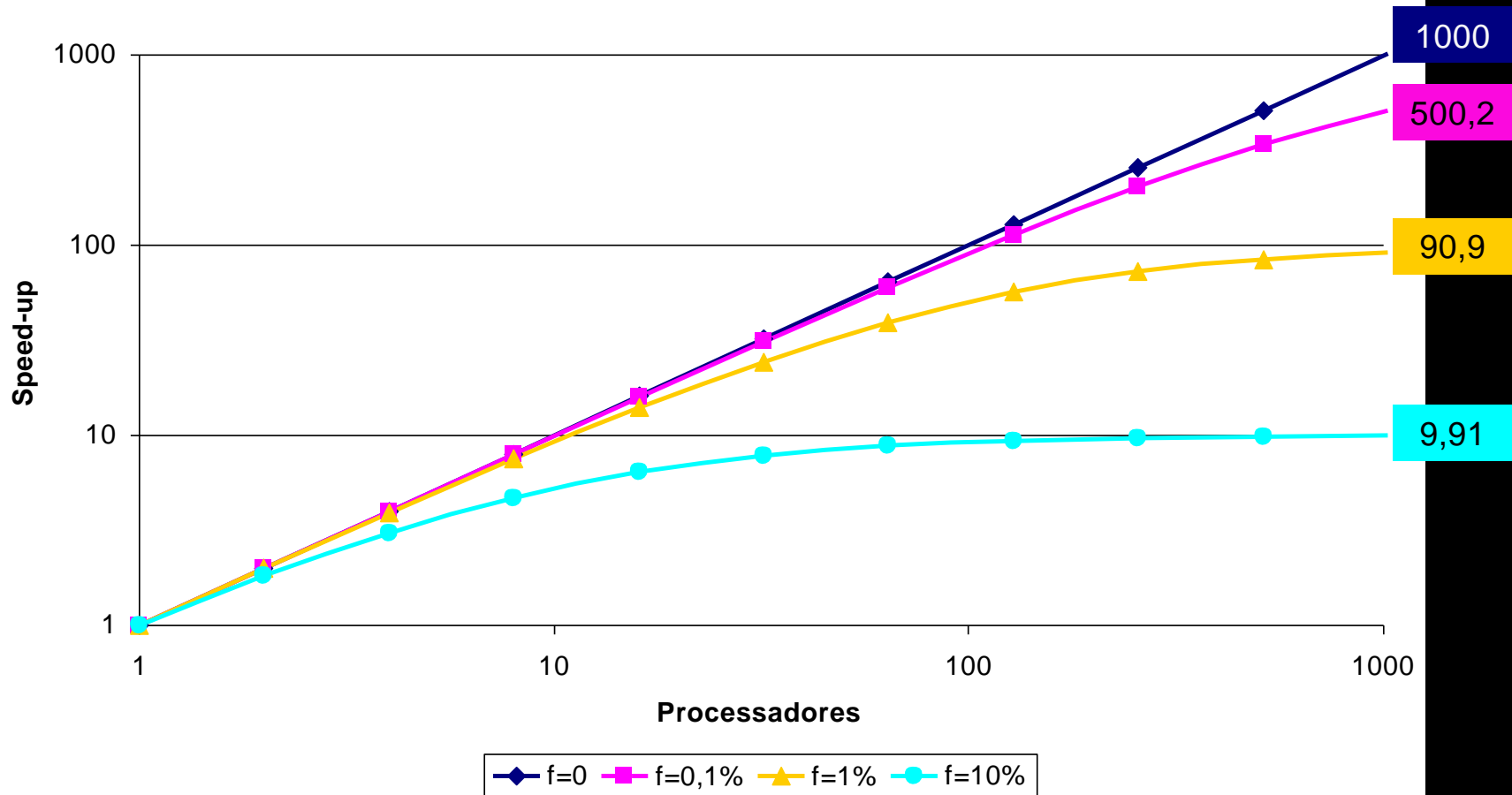
Como o lado direito é uniformemente crescente em  $p$ , o limite superior de  $S(p)$  ocorre quando  $p$  tende a infinito, ou seja,

$$S(p) \leq \frac{1}{f}$$





Speed-up máximo em função da fração do tempo de execução sequencial mantida sequencial na execução paralela





- Tipicamente, a fração do tempo sequencial mantida sequencial diminui ao aumentar o tamanho do problema
  - Técnica comum quando o speed-up não é bom: aumentar artificialmente o tamanho do problema<sup>(1)</sup>
  - Mas há casos em que o aumento é necessário para observar resultados físicos invisíveis em problemas menores <sup>(2)</sup>
- **Strong speed-up**
  - Mantém o tamanho do problema e escala o número de processadores
- **Weak speed-up**
  - Escala o tamanho do problema com o mesmo número de processadores

<sup>(1)</sup> D. H. Bailey: *Twelve Ways to Fool the Masses When Giving Performance Results on Parallel Computers*, *Supercomputing Review*, Aug. 1991

<sup>(2)</sup> J. L. Gustafson: *Reevaluating Amdahl's Law*, *CACM* May 1988



- Definição de Paralelismo
- Níveis de Paralelismo
- Métricas de Desempenho Paralelo
- Lei de Amdahl
- **Necessidade e Utilidade de Paralelismo**
- Lei de Moore
- Memory Wall, Power Wall
- Cray no IME



# Porque Paralelismo? Há Motivos Perenes



- Aplicações (usuários) requerem computadores cada vez mais potentes
- Velocidade e poder computacional de CPUs crescem a velocidade assombrosa
  - Base: "Lei" de Moore (veremos em seguida)
- Requisitos das aplicações (usuários) crescem mais rapidamente do que a velocidade das CPUs
- Acoplar múltiplas CPUs é uma forma viável de reduzir a distância entre as necessidades dos usuários e a velocidade de uma única CPU

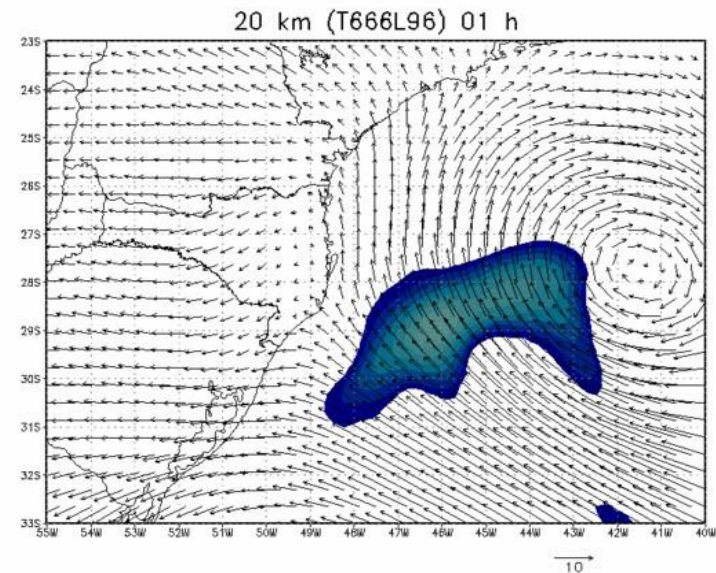
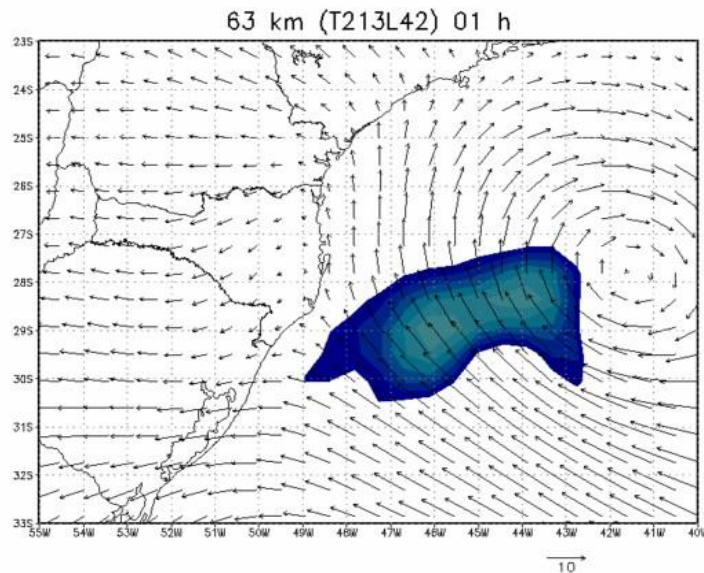
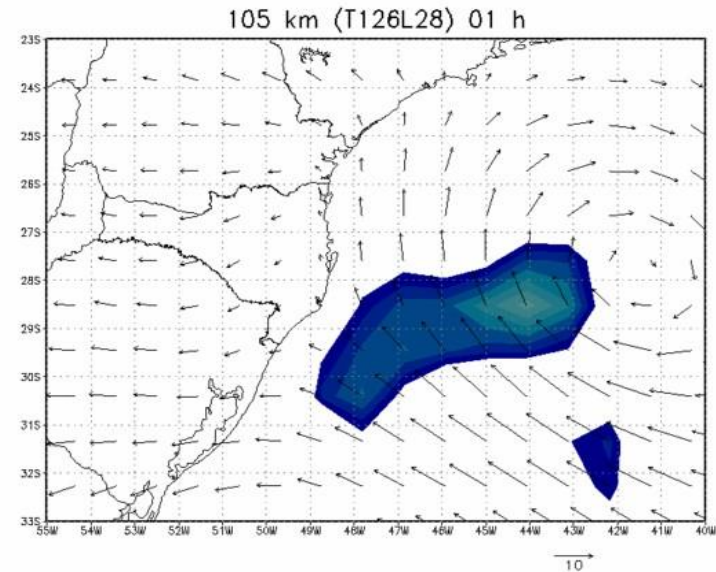
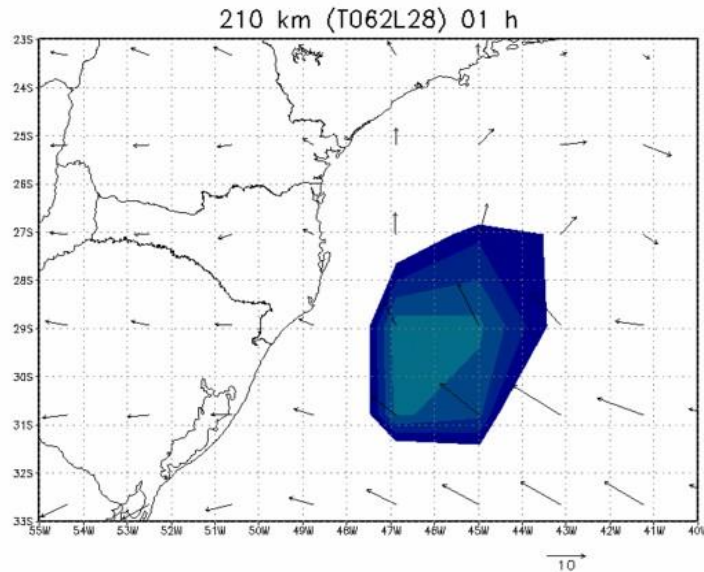


- Usuários estão insatisfeitos com os modelos matemáticos utilizados nas simulações computacionais (não representam aspectos críticos da realidade)
- Modelos matemáticos mais precisos requerem computadores ordens de magnitude mais velozes que os atualmente disponíveis (verdade “desde sempre”)
- Exemplo: Modelo de Circulação Global da Atmosfera



| <b>Resolução Horizontal</b><br>(km) | <b>Flops por Dia de Previsão</b><br>(normalizado) | <b>Dias de Previsão por Dia de CPU</b><br>(1 dia de CPU NEC SX6) |
|-------------------------------------|---------------------------------------------------|------------------------------------------------------------------|
| 210                                 | 1                                                 | 995                                                              |
| 105                                 | 6                                                 | 100                                                              |
| 63                                  | 52                                                | 16                                                               |
| 40                                  | 219                                               | 7                                                                |
| 20                                  | 727                                               | 1                                                                |

Complexidade  $O(1/\text{resolução}^4)$







- Usuários continuarão a demandar maior potência computacional, alcançável por paralelismo
  - Até quando paralelismo resolve?  
*“I know how four oxen can push a wagon,  
but do not how 1024 chicken would do it”*
- Entretanto, há motivos mais recentes para usar paralelismo
  - Motivos alheios à vontade dos usuários e dos fabricantes



# Porque Paralelismo?

## Motivos Recentes:

- “Lei” de Moore
- Memory Wall
- Power Wall