

Programação Paralela

Simone de Lima Martins

Slides baseados no Tutorial Introduction to Parallel Computing
(https://computing.llnl.gov/tutorials/parallel_comp/)



Medida de desempenho

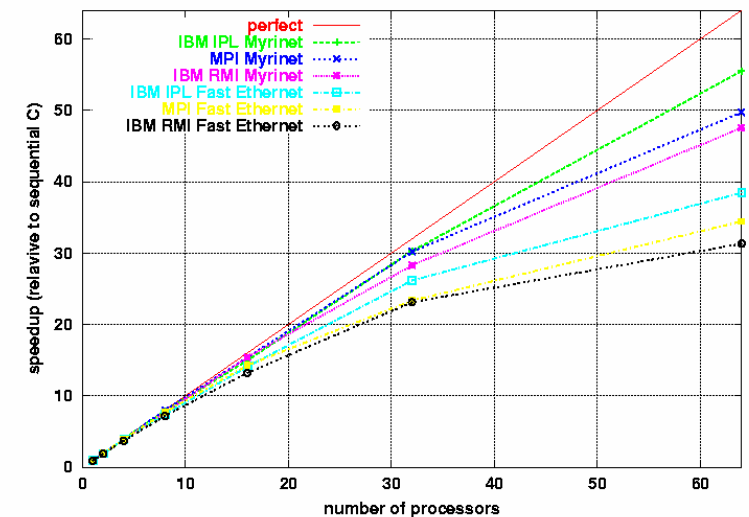
Aceleração (Speedup) = (Tempo em 1 CPU) / (Tempo em p CPUs)

O que devemos comparar?

Programa paralelo em 1 CPU?

Programa equivalente sequencia

Melhor programa sequencial?



Fator de aceleração (speedup)

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

- Para uma máquina paralela com p processadores, speedup ideal seria p (speedup linear)
- $S(p) > p$, chamado de superlinear speedup
 - algoritmo seqüencial sub-ótimo, solução: Simular algoritmo paralelo sequencialmente, executando uma parte paralela após a outra
 - característica particular da arquitetura da máquina paralela
 - memória extra, por exemplo

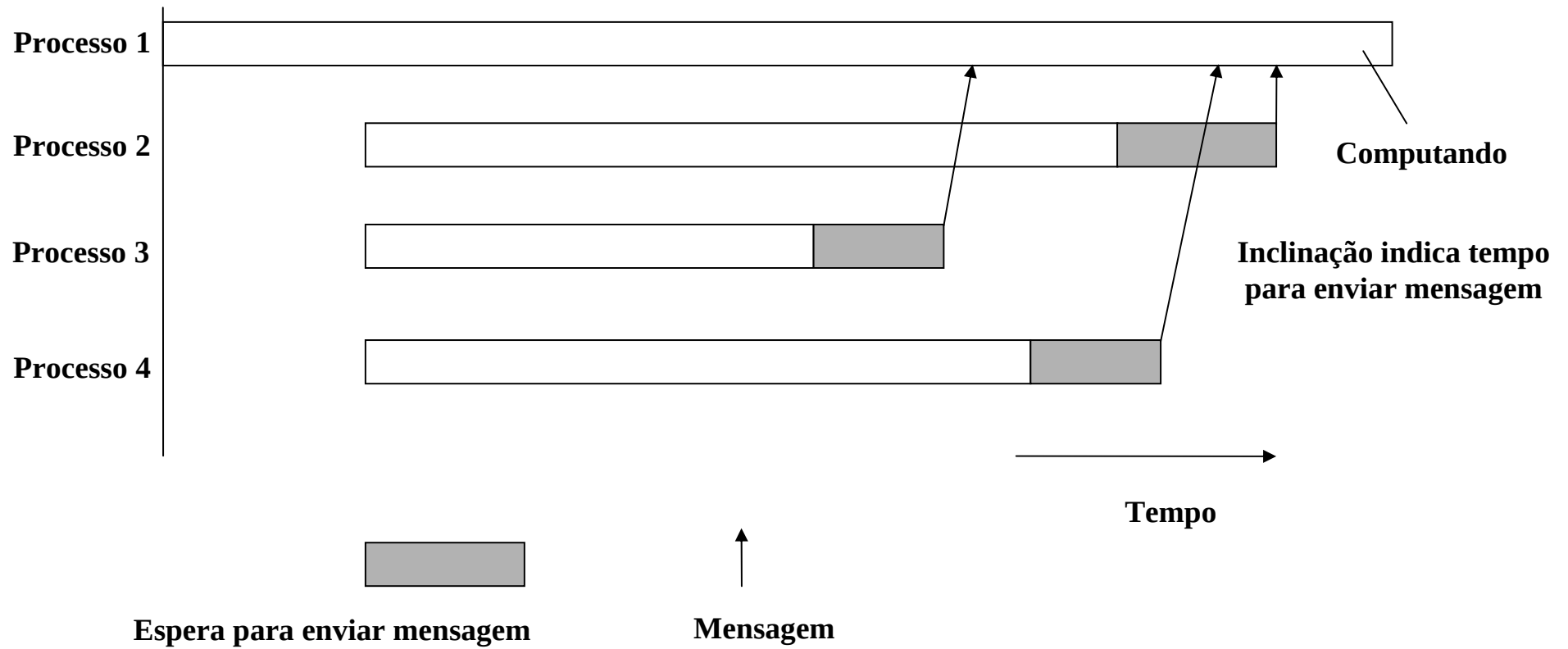
Exemplo de possível melhoria

- Algoritmo de busca de solução em que o espaço de busca pode ser dividido entre os processadores e a busca é feita de forma independente.
- No algoritmo sequencial, os espaços são investigados um depois do outro
- No algoritmo paralelo todos os sub-espaços são investigados em Δt
- Melhor caso: Solução encontrada no último sub-espaço:
- $S(p) = ((p-1)/p * t_s + \Delta t) / \Delta t$
- Pior caso: Solução encontrada no primeiro sub-espaço
- $S(p) = \Delta t / \Delta t = 1$

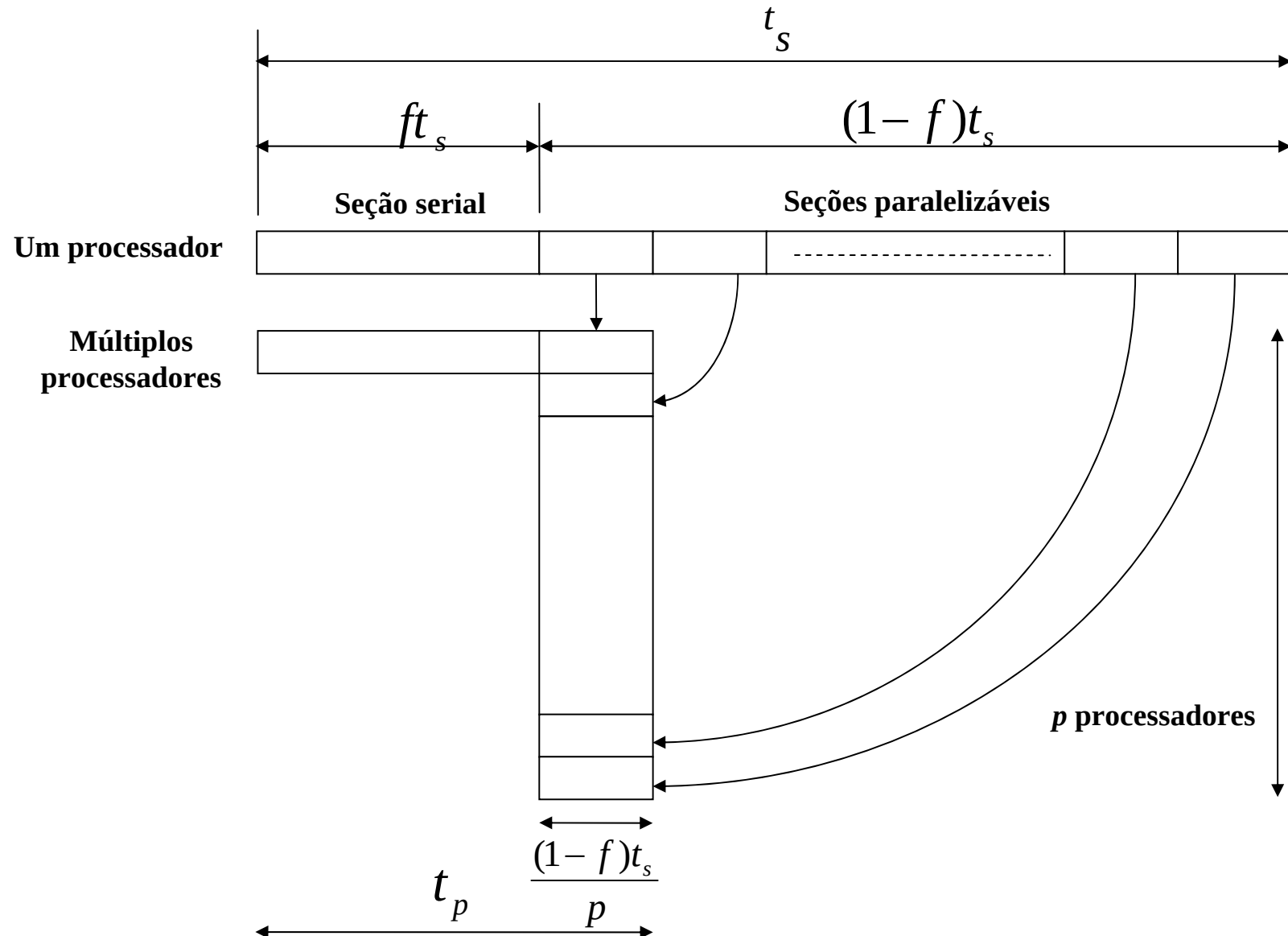
Aceleração máxima

- Limites na aceleração
 - Períodos que alguns processadores estão sem fazer trabalho
 - Computações extras que não são necessárias na versão sequencial: por exemplo, recalcular constantes localmente
 - Tempo de comunicação entre processadores

Speedup



Speedup máximo



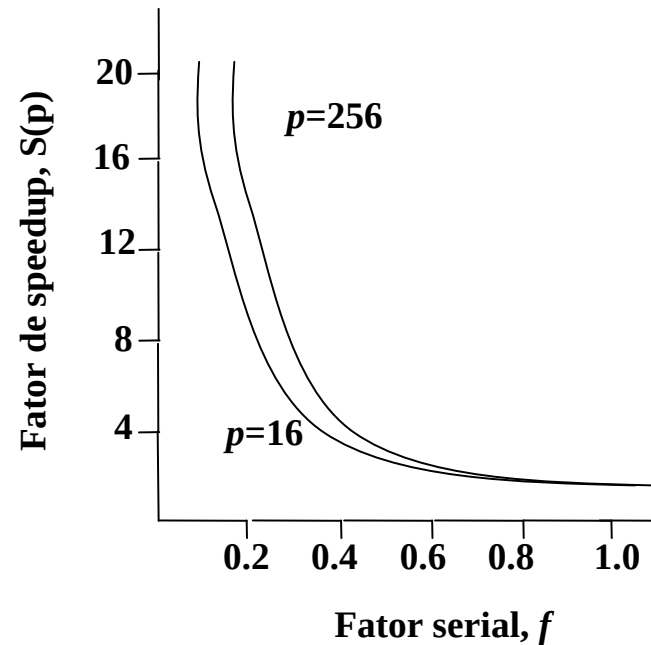
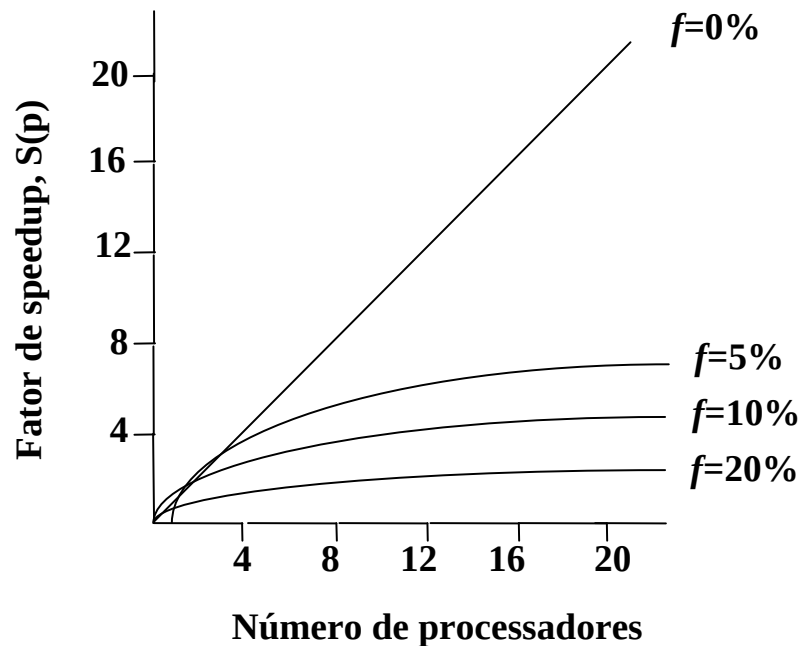
Speedup

- O fator de speedup é dado por:

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

- Equação conhecida com Lei de Amdahl

Lei de Amdhal



- Mesmo com número infinito de processadores o speedup é limitado a $1/f$
- 5% de fração serial $\rightarrow$ speedup máximo de 20, independente do número de processadores

Reavaliação da Lei de Amdhal

- Gustafson verificou aceleração muito maior em uma aplicação do que a prevista por Amdhal.
- Amdhal fez a análise considerando tamanho do problema fixo
- Gustafson considerou tempo de processamento paralelo fixo aumentado-se o tamanho do problema e parte serial fixa, não proporcional ao tempo total sequencial
- A idéia é que, em geral, se utilizam mais processadores para resolver problemas maiores

Reavaliação da Lei de Amdhal

- Gustafson denominou esta medida de aceleração escalonada
- Tempo sequencial = $ft_s + (1-f)t_s$
- Tempo paralelo = $ft_s + ((1-f)t_s)/p$
- Considerando tempo paralelo constante e igual a 1
 - $ft_s p + (1-f)t_s = p \Rightarrow t_s = p - ft_s p + ft_s = p + ft_s(1-p)$
- Aceleração escalonada = $p + ft_s(1-p)$
- Considerando ft_s fixo, temos um gráfico linear da aceleração com o número de processadores

Eficiência

$$E = \frac{\text{tempo de execução utilizando 1 proc.}}{\text{tempo de execução utilizando múltiplos procs.} \times \text{número de processadores}} = \frac{t_s}{t_p \times n}$$

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

- Eficiência fornece a fração de tempo que os processadores estão sendo utilizados para processamento
- Caso $E=50\%$, em média os processadores estão sendo usados metade do tempo em computação real
- $E=100\%$ temos speedup igual a p ou seja todos os processadores estão sendo utilizados o tempo todo

Computações por passagem de mensagens

- Tempo de computação
- Tempo de comunicação
- Taxa de Comp./Com.=Tempo de computação/tempo de comunicação

Como medir desempenho

Use wall clock time

```
int start, stop;  
start = get_time();  
do computations ...  
stop = get_time();  
print(stop-start);
```

Exclua inicializações

Meça desempenho de I/O

Evite instruções de depuração

Depuração de desempenho

Tente entender o comportamento

Não assuma que problema é de comunicação

Determine onde o tempo está sendo gasto

Meça número de mensagens por segundo, volume de dados por segundo, tempo sem fazer nada

Depuração de desempenho

- Se possível, execute o programa como um processo único e depure como um programa sequencial
- Execute o programa com poucos (2 ou 4) processos em um único processador para verificar envio e recebimento de mensagens
- Execute o programa com poucos processos em mais de um processador para verificar problemas de comunicação que podem influenciar na sincronização dos processos