Análise de desempenho

É necessário que o desempenho paralelo seja melhor que o sequencial, pois o custo e a complexidade deste processo devem valer à pena.



(4) Análise de desempenho

- Medidas de tempo e comparações com a versão sequencial do programa são usadas para verificar se houve ganho de desempenho.
- Caso contrário, o programador deverá verificar pontos de gargalo que estejam atrapalhando o desempenho.

(4) Análise de desempenho: problemas frequentes

- Situações onde haja contenção na sincronização de recursos compartilhados (por exemplo, uso sincronizado de variáveis compartilhadas),
- Desbalanceamento de carga de trabalho entre as threads
- Quantidade excessiva de chamadas à biblioteca de threads

(4) Análise de desempenho: objetivos

Desempenho

Capacidade de reduzir o tempo de resolução do problema à medida que os recursos computacionais aumentam

Escalabilidade

Capacidade de aumentar ou manter o desempenho à medida que os recursos computacionais aumentam



(4) Análise de desempenho: fatores limitantes

Limites arquiteturais

Latência e a largura de banda da camada de interconexão.

Capacidade de memória da máquina utilizada.

Limites algorítmicos

Falta de paralelismo inerente ao algoritmo.

Frequência de comunicação.

Frequência de sincronização.

Sistema de execução

Escalonamento deficiente.

Balanceamento de carga.



(4) Análise de desempenho: medidas

Medida básica: Tempo de Execução

O sistema B é n vezes mais rápido que A quando:

$$Texec(A) / Texec(B) = n$$

Maior desempenho → Menor tempo de execução

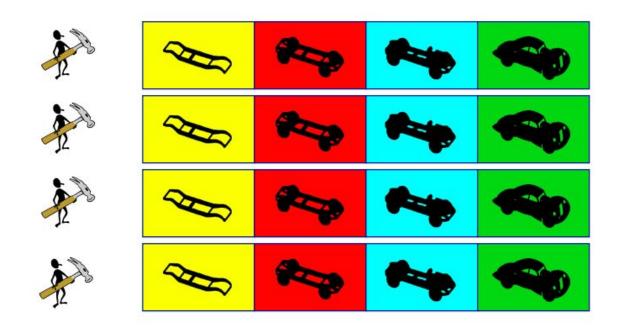
(4) Análise de desempenho: speedup

Speedup: Medida de ganho em tempo

Speedup(P) =
$$T(1 \text{ proc}) / T(P \text{ proc})$$

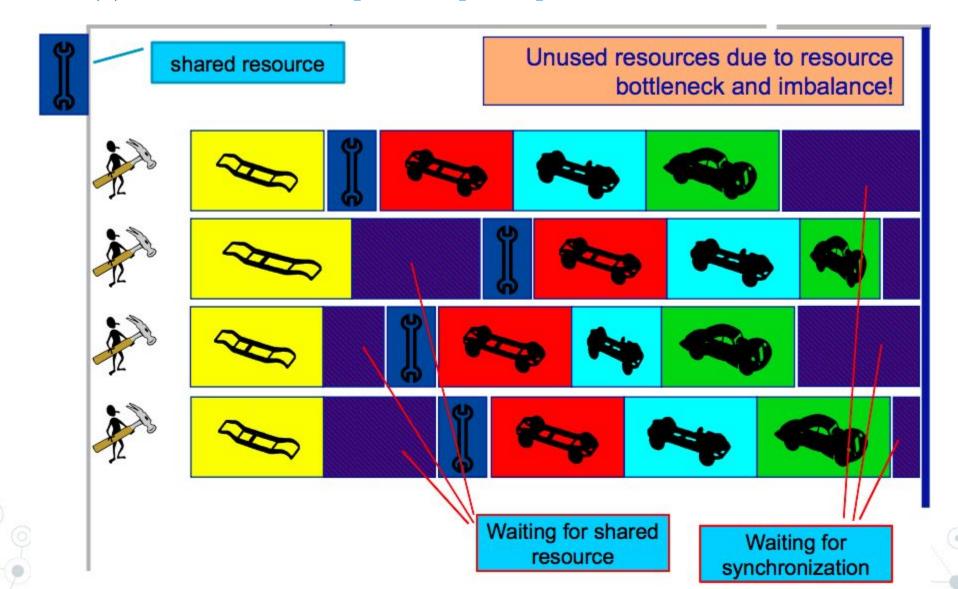
Onde P = número de processadores 1 ≤ Speedup ≤ P

(4) Análise de desempenho: speedup



Após 4 passos: 4 carros "speedup perfeito" = speedup linear

(4) Análise de desempenho: speedup



(4) Análise de desempenho: eficiência

Eficiência: Medida de uso dos processadores

0 < Eficiência ≤ 1



(4) Análise de desempenho: Lei de Amdhal

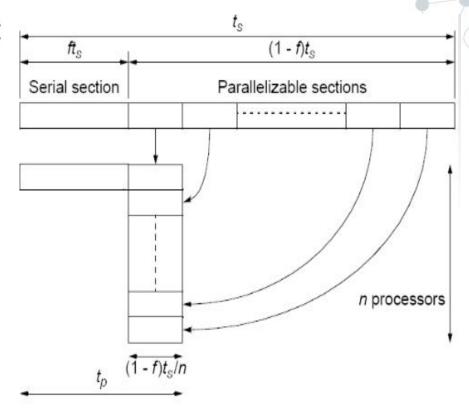
 Limitação teórica para os ganhos de desempenho

Programa serial:
$$Tserial = T0 = (s+q)T0$$

- Onde:
 - s+q=1
 - s corresponde a fração serial do código (impossível de ser paralelizada)
 - q corresponde a fração paralelizavel do código

(4) Análise de desempenho: Lei de Amdhal

- Supondo paralelização ideal:
 - Tpar = sT0 + (q/p)T0
- Speedup = Tserial/Tpar = (s+q)/(s+q/P) =1/(s+q/P)
- Speedup = 1 / [s + (1-s)/P]



(4) Análise de desempenho: Lei de Amdhal

- Lei de Amdhal: considera apenas a escalabilidade dos recursos computacionais, sendo o tamanho do problema fixo.
 - Neste caso, temos limitação do speedup.
- Lei de Gustafson-Barsis: considera o aumento do tamanho da aplicação ou do domínio a medida que se aumentam o número de recursos computacionais.

https://www.youtube.com/watch?v=NaUqvKFj4Oo

(4) Análise de desempenho: tipos de escalabilidade

Escalabilidade forte

Mantém-se o tamanho do problema e escala-se o número de processadores.

É a capacidade de executar aplicações n vezes mais rápidas, onde n é a quantidade de processadores utilizados (speedup).

Escalabilidade fraca

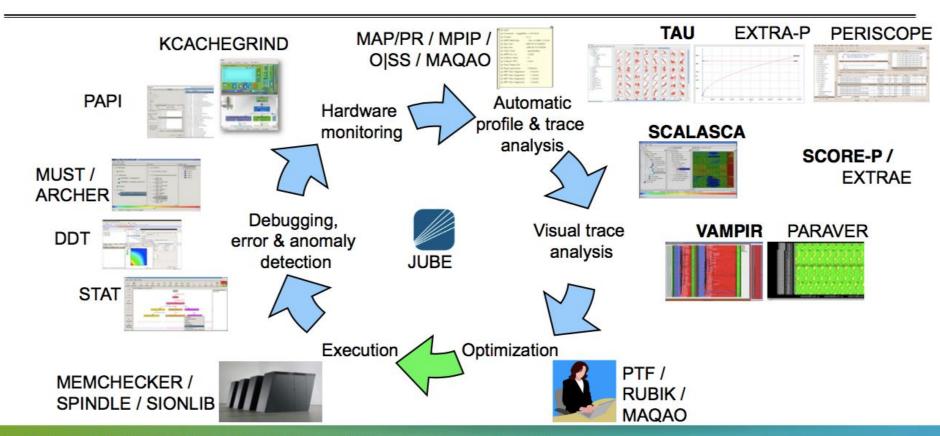
Escala-se o tamanho do problema juntamente com o número de processadores.

É a capacidade de aumentar a carga de trabalho e a quantidade de processadores por um fator de n e manter o tempo de computação.



(4) Análise de desempenho: ferramentas e ciclo de desenvolvimento

Technologies and their integration





Bugs and Speed in HPC Applications: Past, Present, and Future Jeffrey Hollingsworth - ISC 2018