

# Avaliação de Desempenho

October 29, 2009



# O que é desempenho?

- em primeiro lugar, uma ótima tradução para *performance*... :-)
- tempo de execução (o centro das atenções!)
- outras: projeto, ciclo de vida, manutenção, ...
- mesmo outras medidas de execução podem ser importantes:
  - utilização de memória
  - throughput
  - uso da rede



# Como estudar desempenho?

- lei de Amdahl
  - se programa tem fração  $1/s$  inerentemente sequencial, a maior aceleração que conseguiremos é de  $s$
  - relevante quando se paraleliza programas já existentes
- extrapolação a partir de observações
  - *“implementamos o algoritmo na máquina X e obtivemos uma aceleração de 10.8 em 10 processadores”*



# Como estudar desempenho? (cont)

- análise assintótica
  - análise mostra que o tempo será  $O(n \log n)$
- mas o que está acontecendo nos casos que realmente nos interessam?



# Como estudar desempenho? (cont)

- modelos de desempenho
- experimentos
- simulação

# Modelos de desempenho

- objetivo: explicar dados observados e prever comportamento em circunstâncias futuras
  - necessidade de abstrair detalhes menos importantes
- previsão do tempo de execução:

$$T = f(N, P, U, \dots)$$

- N: tamanho do problema
- P: número de processadores
- U: número de tarefas
- ... outras características



- tempo de execução: tempo decorrido do momento em que o primeiro processador começa a executar uma tarefa da aplicação até o momento em que o último processador para de executar.
- podemos olhar o tempo em cada processador:

$$T = T_{comp}^j + T_{comm}^j + T_{idle}^j$$

- ou o tempo total:

$$T = (T_{comp} + T_{comm} + T_{idle})/P$$

# Reduzindo complexidade

- desenvolver uma expressão matemática para descrever  $T$  é uma tarefa complexa...
- *máquina ideal*
  - sem preocupação com topologia da rede, hierarquia de memória, etc
- análise em escala
  - tentativa de identificar fatores insignificantes
- análise empírica
  - calibragem de modelo com experimentos



# Tempo de computação

- possibilidade de medir partes em programa sequencial
- implementação de *kernels* para medidas
- cuidados com alterações devidas a memória, etc

- diferença intra e inter-processador
- tempo idealizado:

$$T_{msg} = t_{startup} + t_{tword}L$$

- experimentos específicos podem determinar esses tempos

- aceleração (*speedup*) – ganho com P processadores

$$A_{relativa} = \frac{T_1}{T_P}$$

- eficiência (*efficiency*) – utilização de cada processador

$$E_{relativa} = \frac{T_1}{PT_P}$$

- aceleração e eficiência absolutas
  - tempo do melhor algoritmos sequencial

# Anomalias em aceleração

- aceleração anômala ou super linear
- motivos:
  - cache, memória virtual
  - irregularidade das estruturas do problema

- algoritmo pode ser adaptado?
- como reage a crescimento de  $N$ ?
- como reage a crescimento de  $P$ ?
- como reage a alterações em  $t_{startup}$  e  $t_{tword}$ ?

- qual o maior número de processadores que podem ser usados produtivamente?

- qual o maior número de processadores que podem ser usados produtivamente?
- conceito de *isoefficiência*:
  - como a quantidade de computação tem que crescer, quando P cresce, para manter a eficiência constante?

$$E_{relativa} = \frac{T_1}{PT_P} = \frac{T_1}{T_{comp} + T_{comm} + T_{idle}}$$
$$\Rightarrow T_1 = E(T_{comp} + T_{comm} + T_{idle})$$

- abordagem iterativa
  - 1 experimentos para encontrar parâmetros ( $t_{startup}$ , etc)
  - 2 análise teórica
  - 3 implementação
  - 4 experimentos para confirmar previsões de análise

- área com suas próprias questões e literatura
  - twelve ways to fool the masses when giving performance results on parallel computers (David Bailey)
- levantamento do que queremos obter...
- reprodução de experimentos já realizados por outros grupos

# Projeto sistemático de experimentos

- definição precisa de objetivos (fronteiras)
- seleção de métricas
- enumeração de parâmetros que afetam o desempenho
  - todos os parâmetros que podem afetar o desempenho
- seleção de fatores para estudo
- seleção de carga de trabalho
- seleção de métricas
- projetos dos experimentos
- análise e interpretação de resultados



- velocidade
  - tempo de resposta
  - *throughput*
  - recursos consumidos
- confiabilidade
  - tempo entre falhas
- disponibilidade
  - fração do tempo em que sistema está disponível
- outros que não sabemos como medir...
  - usabilidade
  - flexibilidade
  - ...

- históricas X sintéticas
- nível (aplicação, sistema operacional, CPU)
- representatividade (relação da carga sintética com a carga real...)
  - taxa de chegada de pedidos
  - demanda de recursos
  - perfil de utilização

# Erros comuns em análise de desempenho

- objetivos preconceituosos
- abordagem não sistemática
- nível de detalhe inadequado
- parâmetros importantes ignorados
- cargas não representativas
- ...



- algoritmos não determinísticos
- precisão do timer
  - loops de repetição e médias
- custos de inicialização e terminação
- interferência de outros programas
- interferência entre experimentos
- alocação de recursos aleatória

# Avaliação – motivos para surpresas

- desbalanceamentos de carga
- computação replicada
- algoritmo e ferramenta que não combinam
- competição por banda passante



- Foster (DBPP): capítulo 3 (ler)
- David Bailey. Twelve ways to fool the masses when giving performance results on parallel computers. *Supercomputing Review*, Aug. 1991, pg. 54–55.
- R. Jain. *The Art of Computer Systems Performance Analysis*. Wiley, 1991.