

ARQUITETURA DE COMPUTADORES

Avaliação do desempenho de
processadores

Introdução

- Tempo de relógio real necessário para execução de um programa (ou tempo de resposta) inclui:
 - acessos a disco
 - atividades de I/O
 - *overhead* do sistema operacional, da rede, ...
 - Para sistemas multitarefa...
 - *throughput* é uma medida do número de tarefas executada por unidade de tempo
-

Introdução

- ❑ O que otimizar?
 - tempo de resposta de um programa isolado?
 - *throughput* do sistema?
 - ❑ Distinção entre ...
 - tempo de relógio real gasto em um programa, e
 - tempo de CPU efetivamente gasto com tarefas específicas do programa
 - ❑ não inclui tempo de execução de outros programas, tempo de espera por I/O, ...
-

Introdução

- ❑ Tendo em vista os objetivos desta disciplina, desempenho de CPU será medido apenas em função do tempo de CPU do usuário
-

Introdução

Desempenho é consequência de otimizações feitas em 3 dimensões do hardware:

- ❑ Arquitetura
 - processadores RISC
 - instruções otimizadas
 - ❑ Organização
 - paralelismo
 - organização da memória
 - ❑ Tecnologia
 - velocidade de chaveamento das portas lógicas
-

Métricas básicas

- ❑ ciclos de clock = intervalos básicos de tempo nos quais são executadas as operações elementares de uma instrução
 - transferências de valores entre registradores
 - operações aritméticas na ALU
 - ❑ período do clock T = tempo de duração de um ciclo de clock
 - p.ex. 1 ns
 - ❑ frequência de operação f = número de ciclos de clock por unidade de tempo
$$f = 1 / T$$
se $T = 1 \text{ ns}$ então $f = 1 / 1 \cdot 10^{-9} = 10^9 = 1 \text{ GHz}$
-

Métricas básicas

$$\text{tempo de CPU de um programa} = \text{n}^\circ \text{ de ciclos de clock do programa} \times \text{período do clock}$$

por exemplo:

- programa gasta $10 \cdot 10^6$ ciclos
- período do clock é 1 ns



$$\text{tempo de CPU do programa} = 10 \cdot 10^6 \times 1 \cdot 10^{-9} = 10 \cdot 10^{-3} = 10 \text{ ms}$$

Exercício Neander:

```
LDA 80h
ADD 81h
NOT
STA 82h
HLT
```

Métricas básicas

Alternativamente:

$$\text{tempo de CPU de um programa} = \frac{\text{n}^\circ \text{ de ciclos de clock do programa}}{\text{frequência do clock}}$$

Formas de aumento do desempenho:

- diminuir o período do clock ➤ tecnologia, organização
- diminuir n° de ciclos necessários para execução do programa ➤ organização, arquitetura

Objetivos muitas vezes conflitantes.

Métricas básicas (Exemplo)

- programa roda em 10 s na máquina A, que tem clock de 1 GHz
- queremos rodá-lo em 6 s numa máquina B com nova tecnologia e nova organização
- a máquina B pode ter um aumento substancial de frequência de clock
- no entanto, a organização da máquina B exigirá 1,2 vezes mais ciclos de clock para executar instruções do que a máquina A
- qual é a frequência de clock necessária para a máquina B ?

Métricas básicas (Exemplo)

$$\text{tempo de CPU de um programa} = \frac{\text{nº de ciclos de clock do programa}}{\text{frequência do clock}}$$

$$10 \text{ s} = \frac{\text{nº de ciclos de clock em A}}{10^9 \text{ ciclos / s}}$$

$$\text{nº de ciclos de clock em A} = 10 \cdot 10^9 \text{ ciclos}$$

$$6 \text{ s} = \frac{1,2 \times \text{nº de ciclos de clock em A}}{\text{frequência do clock em B}} = \frac{1,2 \times 10 \cdot 10^9}{\text{frequência do clock em B}}$$

$$\text{frequência do clock em B} = 2 \text{ GHz}$$

Métricas Básicas - CPI

$$\text{nº de ciclos de clock do programa} = \text{nº instruções do programa} \times \text{nº médio de ciclos de clock por instrução}$$

CPI = nº médio de ciclos de clock por instrução.
É uma média sobre todas as instruções executadas no programa.

Exercício Neander:

```
LDA 80h
ADD 81h
NOT
STA 82h
HLT
```

Métricas Básicas – CPI (Exemplo)

- duas máquinas implementam o mesmo conjunto de instruções
- um mesmo programa é rodado em ambas as máquinas
- máquina A tem período de clock de 1 ns e CPI = 2,0 para este programa
- máquina B tem período de clock de 2 ns e CPI = 1,2 para o mesmo programa
- qual máquina é mais rápida ?

Métricas Básicas – CPI (Exemplo)

$$\text{nº de ciclos de clock do programa} = N \text{ (nº instruções do programa) } \times \text{CPI}$$

$$\text{nº de ciclos de clock de A} = 2,0 \times N$$

$$\text{nº de ciclos de clock de B} = 1,2 \times N$$

$$\text{tempo de CPU de um programa} = \text{nº de ciclos de clock do programa} \times \text{período do clock}$$

$$\text{tempo de CPU de A} = 2,0 \times N \times 1 \text{ ns} = 2,0 \times N \text{ ns}$$

$$\text{tempo de CPU de B} = 1,2 \times N \times 2 \text{ ns} = 2,4 \times N \text{ ns}$$

portanto: máquina A é 1,2 vezes mais rápida do que B na execução deste programa

Métricas Básicas – medições

$$\text{tempo de CPU} = \text{nº de instruções} \times \text{CPI} \times \text{período do clock}$$

Como obter estes valores?

Métricas Básicas – medições

- ❑ Tempo de CPU é medido executando-se o programa
 - ❑ Período (ou frequência) do clock é divulgado pelo fabricante
 - ❑ nº de instruções e CPI são mais difíceis de obter
 - nº de instruções depende da arquitetura (e do compilador)
 - CPI depende da organização (e do programa)
 - ❑ conhecendo-se nº de instruções ou CPI, pode-se determinar o outro a partir da fórmula
-

Benchmarks

- Benchmarks são **conjuntos de programas** representativos da carga de trabalho de uma máquina
- utilizados para avaliação de desempenho, segundo métricas já discutidas
 - geralmente de domínio público
 - procuram explorar repertório de recursos da arquitetura

Problemas com benchmarks

- melhor avaliação seria feita com a carga de trabalho efetivamente utilizada em cada máquina
 - escolha dos benchmarks e aplicação rigorosa da metodologia de avaliação
 - fabricantes podem tentar otimizar arquitetura e/ou organização e/ou compilador para executar de forma mais eficiente apenas os benchmarks
-

Tipos de benchmarks

- ☐ Aplicações reais ou modificadas ?
 - Quais ? ... Dados de entrada ?
 - ☐ exemplos:
 - programas do SPEC (empresa de padronização de avaliação)
 - iCOMP (benchmarks da Intel, aplicações sob Windows, PC-workload)
 - transações de banco de dados (TPC-A, TPC-B, etc.)
 - simuladores de equações diferenciais (NAS: benchmarks NASA)
 - Outros...
 - ☐ Linpack: rotinas de resolução de sistema de equações lineares
 - ☐ Puzzle, Quicksort, Sieve, ...
 - ☐ Dhrystone, Whetstone, ...
-

Resumo

- ☐ Desempenho = Tempo de CPU do usuário
 - ☐ O que afeta o desempenho?
 - ☐ Métricas básicas
 - ☐ Benchmarks
-

Literatura

□ *Patterson & Hennessy – cap 4*

□ <http://www.anandtech.com/bench/default.aspx?p=87&p2=118&c=1>

Os professores de **Thomas Edison** mandaram ele para casa, dizendo que ele era confuso e atrapalhado.

A ele são atribuídas mais de 1000 patentes destacando-se a lâmpada elétrica incandescente, o gramofone, o cinescópio e o microfone de grânulos de carvão para o telefone.

A idéia de geração e distribuição de eletricidade é atribuída a ele.

Questões adicionais ...

- ❑ O que diz a **Lei de Amdahl**?
Exemplo.

Outras métricas: MIPS

$$\begin{aligned} \text{MIPS} &= \frac{\text{n}^\circ \text{ instruções}}{\text{tempo de CPU} \times 10^6} = \frac{\text{n}^\circ \text{ instruções}}{\text{ciclos} \times \text{período} \times 10^6} \\ &= \frac{\text{n}^\circ \text{ instruções} \times \text{frequência}}{\text{n}^\circ \text{ instruções} \times \text{CPI} \times 10^6} \end{aligned}$$
$$\text{MIPS} = \frac{\text{frequência}}{\text{CPI} \times 10^6}$$

Problemas com a medida em MIPS:

- não se pode comparar máquinas com conjuntos de instruções diferentes, pois certamente o n° de instruções será diferente para um mesmo programa
- MIPS varia de um programa para outro na mesma máquina
- MIPS pode variar inversamente ao desempenho