UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

TM-405 - Arquitetura de Computadores I

Aula 2 - Desempenho

Raul Sena Ferreira

Porque medir a performance de um PC?

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

A performance é a responsável pela eficácia de todo o sistema, composto pelo hardware e software

Facilita descobrir quais fatores de hardware influenciam na performance global

Ajuda a descobrir qual computador tem performance melhor do que outro

Fatores de HW que influenciam

- Utilização por parte do compilador das instruções geradas na máquina na geração de um código do programa
- Maneira de o hardware implementar as instruções
- Maneira como a memória e os dispositivos de E/S se comportam durante o processamento de um programa

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Definição de performance

Qual o avião de melhor performance?

- Aquele que possui a maior velocidade?
- Aquele que transporta mais passageiros em um espaço de tempo?

Avião	Capacidade	Autonomia (mi)	Velocidade (mph)
Boeing 777	375	4630	610
Boeing 747	470	4150	610
Concorde (BAC-Sub)	132	4000	1350
Douglas DC-8-50	146	8720	544

Medidas de desempenho

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

- Tempo de execução ou tempo de resposta: é o tempo que um programa leva para ser executada
- Tempo de Resposta (latência): Quanto tempo leva para executar uma instrução
- Throughput: Vazão, quantidade de instruções executadas em um determinado intervalo de tempo

Reduzir o tempo de resposta quase sempre leva a um aumento do Throughput

Calculando a performance

Performance (X) = 1 / Tempo de execução (X)

Ou seja, se Performance(X) > Performance(Y) então:

- 1 / Tempo de execução(X) > 1 / Tempo de execução(Y) ou,
- Tempo de execução(Y) > Tempo de execução(X)

Se X é n vezes mais rápida que Y então:

- Performance(X) / Performance(Y) = n

Performance relativa

Se o computador A executa um programa em 10 segundos e um computador B executa o mesmo programa em 15 segundos, pergunta-se:

Quantas vezes o computador A é mais rápido que B?

O computador A será n vezes mais rápido que o computador B se:

$$P(A) / P(B) = TE(B) / TE(A) = 1.5$$

Logo, **o computador B é 1.5 vezes mais lento que o computador A** ou que

o computador A é 1.5 vezes mais rápido que o computador B

Tempo de CPU

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Tempo de CPU ou Tempo de Processador: Tempo gasto pelo processador em um programa em particular

Tempo de CPU do usuário: Tempo que a CPU gasta na execução das instruções do programa do usuário

Tempo de CPU do sistema: Tempo que a CPU gasta no sistema operacional para executar tarefas em benefício do programa do usuário

Tempo de CPU = Tempo de CPU do usuário + Tempo de CPU do sistema

Tempo de CPU

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Exemplo: Se o tempo de CPU é 90,7s e o de sistema 12,9s e o tempo decorrido é de 2m39s (159s), o percentual do tempo decorrido gasto pelo processo será:

$$(90,7 + 12,9) / 159 = 0,65$$

Ou seja, mais de 65% do tempo total gasto do início ao fim do programa foi gasto com E/S

Relação entre as métricas

Tempo de CPU para um programa = ciclos de clock × Tempo do ciclo de clock

- Ciclos de clock são intervalos de tempo discretos. São também conhecidos como: ticks, clock ticks, períodos de clock, clocks ou ciclos

Tempo de CPU para um programa = ciclos de clock / frequência do clock

- Frequência é o inverso do período: f = 1 / T

Melhora do desempenho:

- Redução do tamanho do ciclo de clock
- Redução do número de ciclos de clock

Ciclo por instruções - CPI

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

CPI = Número de instruções para um programa × Média dos ciclos de clock por instrução

Permite a comparação de diferentes implementações em uma mesma arquitetura do conjunto de instruções

Equação do desempenho:

- Tempo de CPU para um programa = Número de instruções × CPI × Tempo do ciclo de clock
- Tempo de CPU para um programa = (Número de instruções × CPI) / frequência do clock

Ciclo por instruções - CPI

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

CPI = Número de instruções para um programa × Média dos ciclos de clock por instrução

Permite a comparação de diferentes implementações em uma mesma arquitetura do conjunto de instruções

Equação do desempenho:

- Tempo de CPU para um programa = Número de instruções × CPI × Tempo do ciclo de clock
- Tempo de CPU para um programa = (Número de instruções × CPI) / frequência do clock

Exemplo de cálculo

O projetista de um compilador deseja decidir entre duas possíveis sequências de código para a resolução de um problema. Dado os tipos de instrução e o número de ciclos por instrução (CPI) de cada tipo, pergunta-se:

- Qual o código mais rápido?
- Qual a CPI de cada um dos programas?

Classe de instrução	CPI
A	1
B	2
C	3

	Número de instruções		
Código	Classe A	Classe B	Classe C
1	2	1	2
2	4	1	1

2017.1

Resolução

- Total de instruções do programa 1: **2 + 1 + 2 = 5 instruções**
- Total de instruções do programa 2: **4 + 1 + 1 = 6 instruções**
- O número de ciclos de clock para o código 1: **(2 x 1) + (1 x 2) + (2 x 3) = 10**
- O número de ciclos de clock para o código 2: **(4 x 1)+(1 x 2)+(1 x 3)=9 ciclos**
- O código mais rápido vem de:
  - **CPI médio do código 1 = 10/5 = 2.0**
  - **CPI médio código 2 = 9/6 = 1.5**
- O código mais rápido é o 2

O código 2 tem o CPI mais baixo, ainda que execute uma instrução a mais

MIPS e MFLOPS

MIPS:

- Million instruction per second
- No. instruções / (tempo de execução ×10^6)

MFLOPS:

- Millions of floating-point operations per second
- No. operações de PF / (tempo de execução ×10^6)

Desvantagens:

- Contexto limitado. Distorções. Resultados enganadores

Lei de Amdahl

O aumento de desempenho possível com uma determinada melhoria é limitado pela quantidade de uso do recurso melhorado

Tempo de execução após melhoria:

- Tempo de execução não afetado + (Tempo de execução afetado / Quantidade de melhoria)

Lei de Amdahl - Exemplo

UFRRJ
UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO

Suponha que um programa seja executado em 100s e a multiplicação seja responsável por 80s desse tempo. O quanto precisamos melhorar a velocidade da multiplicação se quisermos que o programa seja executado 4x mais rápido?

- Tempo de execução após melhoria = 25
- Tempo de execução não afetado = 100 - 80
- Tempo de execução afetado = 80
- Quantidade de melhoria = ???

Lei de Amdahl - Solução

- 25 = (100 - 80) + (80 / n)
- (25 - 20) × n = 80
- n = 16

n = 16 → Quantidade de melhoria a ser aplicada sobre a parte “melhorável”

Aceleração ou SpeedUp

Medida de como a máquina se comporta após a implementação de uma melhoria em relação ao seu comportamento anterior

Aceleração = Desempenho após a melhoria / desempenho antes da melhoria

- Aceleração = 1: Não houve melhoria efetiva
- Aceleração > 1: Houve melhoria
- Aceleração < 1: A melhoria deixou o programa mais lento (não houve melhoria)

Programas para avaliação de desempenho

Conjunto de programas executado pelo usuário: Carga de Trabalho (**workload**)

Avaliação entre diferentes sistemas: Quatro níveis de programa (em ordem decrescente de precisão de previsão)

- Programas reais
- Núcleos ou kernels: Pedaços de programas reais
- Toy Benchmarks: 10 a 100 linhas de código que produzem um resultado
- conhecido a priori
- Benchmarks sintéticos: frequência média de operações de um grande conjunto de programas

Programas para avaliação de desempenho

Benchmarks:

- Aplicações que representam cargas de trabalho (workloads), cujo objetivo é estimar o desempenho das cargas de trabalho reais
- Podem conter aplicações típicas de processamento científico, compiladores, processadores de texto
- Exemplos:
  - NAS
  - SPLASH
  - SPEC (System Performance Evaluation Cooperative): SPECint, SPECfp, SPECWeb
  - Whetstone (ponto flutuante)
  - Dhrystone (inteiro e string)

"I do not fear computers. I fear the lack of them."
Isaac Asimov