

Medidas de Desempenho

- ❑ distinção entre ...
 - ❑ tempo total para a execução de um programa e
 - ❑ tempo gasto pelo processador (tempo de CPU ou tempo de processador) em benefício deste programa (com tarefas específicas do programa)
 - ❑ não inclui tempo de execução de outros programas, nem o tempo de espera por I/O, ...

- ❑ **O tempo de CPU está dividido em ...**
 - ❑ tempo de CPU do usuário (*user CPU time*) – tempo gasto executando instruções do programa do usuário
 - ❑ tempo de CPU do sistema (*system CPU time*) – tempo gasto com tarefas do S.O. necessárias para a execução do programa do usuário

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

☐ No UNIX, o comando “time” pode retornar:

90.7u 12.9s 2:39 65%

Tempo de CPU do usuário (*user CPU time*) = 90,7 segundos

Tempo de CPU do sistema (*system CPU time*) = 12,9 segundos

Tempo total transcorrido (*elapsed time*) = 2 minutos e 39 segundos (= 159 segundos)

Precentual do tempo total transcorrido que foi gasto pela CPU no programa é:

$$\frac{90.7 + 12.9}{159} = 0.65$$

Definições

período do clock T = tempo de duração de um ciclo de clock

freqüência de operação f = número de ciclos de clock por unidade de tempo

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

Exemplo

Se o período de clock $T = 10 \text{ ns}$ ou 10×10^{-9} segundos,
qual a frequência de operação (ou frequência do clock)?

$$\text{Resp.: } f = 1 / T = \frac{1}{10 \text{ ns}} = \frac{1}{10 \times 10^{-9}} = \frac{1}{1 \times 10^{-8}}$$

$$= 1 \times 10^8 \text{ Hz} = \left\{ \begin{array}{l} \underline{\hspace{2cm}} \text{ KHz} \\ \underline{\hspace{2cm}} \text{ MHz} \\ \underline{\hspace{2cm}} \text{ GHz} \end{array} \right.$$

Exercícios

Se o período de clock $T = 100 \text{ ns}$,
qual a frequência do clock em KHz, MHz e GHz?

Se o período de clock $T = 200 \text{ ns}$,
qual a frequência do clock em KHz, MHz e GHz?

Se a frequência do clock 150 MHz , qual o tempo de
clock em ns, us e ms?

MIPS

Significa milhões de instruções por segundo.

Se uma máquina possui 1 MIPS significa que ela executa 1×10^6 instruções em 1 segundo.

$$\text{MIPS} = \frac{\text{n}^\circ \text{ instruções}}{\text{tempo de CPU} \times 10^6}$$

$$\text{MIPS} = \frac{1 \times 10^6}{1 \text{ s}} \div 10^6 = 1$$

Um programa A possui 5 milhões de instruções e é executado em 2 segundos, quantos MIPS a máquina possui?

MIPS

Significa milhões de instruções por segundo.

Se uma máquina possui 1 MIPS significa que ela executa 1×10^6 instruções em 1 segundo.

$$\text{MIPS} = \frac{\text{n}^\circ \text{ instruções}}{\text{tempo de CPU} \times 10^6} \quad \text{MIPS} = \frac{1 \times 10^6}{1 \text{ s}} \div 10^6 = 1$$

Um programa A possui 5 milhões de instruções e é executado em 2 segundos, quantos MIPS a máquina possui?

$$\text{MIPS} = \frac{5 \times 10^6}{2 \text{ s}} \div 10^6 = 2.5$$

Exercícios

Considere a seguinte tabela para 2 programas diferentes e em 2 máquinas diferentes:

	Prog. A	Prog. B
Máquina 1	2 s	1 s
Máquina 2	0.5 s	0.8 s

1) Qual a máq. mais rápida e quanto em cada um dos programas?

2) Se o prog. A possui 100 milhões de instruções na máq. 1 e 20 milhões na máquina 2, qual a velocidade em MIPS em cada uma das máquinas?

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**

Problemas com a medida em MIPS:

- não se pode comparar máquinas com conjuntos de instruções diferentes, pois certamente o nº de instruções será diferente para um mesmo programa
- MIPS varia de um programa para outro na mesma máquina
- MIPS pode variar inversamente ao desempenho

Outras Medidas: MFLOPS

$$\text{MFLOPS} = \frac{\text{nº de operações de ponto flutuante no programa}}{\text{tempo de CPU} \times 10^6}$$

Não se contam instruções de PF e sim operações para evitar comparações injustas entre máquinas com instruções diversas

- somas, subtrações, multiplicações, divisões
- precisão simples, precisão dupla
- instruções mais complexas: seno, raiz quadrada

Problemas na comparação:

- máquinas diferentes têm não apenas conjuntos diferentes de instruções, mas também de operações**
- certas operações (p.ex. soma em precisão simples) são muito mais rápidas do que outras (p.ex. divisão em precisão dupla)**

Proposta ?

Problemas na comparação:

- máquinas diferentes têm não apenas conjuntos diferentes de instruções, mas também de operações**
- certas operações (p.ex. soma em precisão simples) são muito mais rápidas do que outras (p.ex. divisão em precisão dupla)**

Proposta :

Usar não uma métrica única, mas uma que dependa de mais fatores:

- do tempo necessário para cada instrução**
- da quantidade de instruções de um programa**
- de características físicas (de velocidade) da máquina**

-> determinar uma equação de performance da CPU

Tempo de CPU ou CPU time e ciclos de clock

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

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**
- diminuir n° de ciclos necessários para execução do programa**

Estes objetivos são muitas vezes conflitantes.

**Um programa é executado em 10 s em uma máquina a 100 MHz.
Quantos ciclos de clock são necessários para a sua execução?**

**Um programa é executado em 10 s em uma máquina a 100 MHz.
Quantos ciclos de clock são necessários para a sua execução?**

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

Alternativamente:

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

**Um programa é executado em 10 s em uma máquina a 100 MHz.
Quantos ciclos de clock são necessários para a sua execução?**

$$f = 100 \text{ MHz} \Rightarrow \text{tempo de clock} = 1/100\text{MHz} = 1 \times 10^{-8} \text{ s}$$

Como o tempo de CPU = 10 s e considerando que cada ciclo de clock demora 1×10^{-8} s, a quantidade de ciclos será:

$$\text{Num ciclos} = \frac{\text{tempo de todos os ciclos}}{\text{tempo de 1 ciclo}} = \frac{10}{1 \times 10^{-8}} = 10 \times 10^8$$

Ou

$$\text{Num de ciclos} = 100 \text{ MHz} * 10 \text{ s} = 1000 \times 10^6 = 10 \times 10^8$$

Exercício

Um programa roda em 10s em uma máquina A que possui um clock de 400MHz.

Queremos um speedup de 1.5 ao executar esse mesmo programa em uma máquina B.

A máquina B poderá sofrer um overclock substancial sem nenhuma seqüela adicional, entretanto exigirá 1,2 vezes mais ciclos de clock para executar as instruções do que a máquina A.

Qual a frequencia de clock necessária para a máquina B?

Exercício

Um programa roda em 10s em uma máquina A que possui um clock de 400MHz.

Queremos um speedup de 1.5 ao executar esse mesmo programa em uma máquina B.

A máquina B poderá sofrer um overclock substancial sem nenhuma seqüela adicional, entretanto exigirá 1,2 vezes mais ciclos de clock para executar as instruções do que a máquina A.

Qual a frequencia de clock necessária para a máquina B?

- 1) Determinar a quantidade de ciclos necessários na máquina A para a execução do programa**
- 2) Determinar o tempo que o programa deverá gastar na máquina B**
- 3) Determinar a quantidade de ciclos na máquina B**
- 4) Determinar a frequencia da máquina B já conhecendo o tempo e a quantidade de ciclos.**

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

$$1 \quad 10s = \frac{\text{nº de ciclos de clock em A}}{400 \times 10^6 \text{ ciclos / s}}$$

$$\text{nº de ciclos de clock em A} = 4000 \times 10^6 \text{ ciclos}$$

$$2 \quad \text{Speedup} = \frac{\text{tempo s/ melhoria}}{\text{tempo c/ melhoria}} = \frac{\text{tempo na máquina A}}{\text{tempo na máquina B}}$$

$$\text{tempo na máquina B} = 1.5 / 10 = 6.667 \text{ s}$$

$$4 \quad 6.667 \text{ s} = \frac{1,2 \times \text{nº de ciclos de clock em A}}{\text{frequência do clock em B}} = \frac{1,2 \times 4000 \times 10^6}{\text{frequência do clock em B}}$$

$$\text{Frequência de clock de B} = 719.96 \text{ MHz}$$

CPI ou Ciclos por Instrução

Considere um programa com 1000 somas. Suponha que eu gaste 1 ciclo de máquina para cada soma.

Se a máquina opera a 100 MHz, quanto tempo o programa demora?

CPI ou Ciclos por Instrução

Considere um programa com 1000 somas. Suponha que eu gaste 1 ciclo de máquina para cada soma.

Se a máquina opera a 100 MHz, quanto tempo o programa demora?

- 1) Determinar o número de ciclos**
- 2) Determinar o tempo de cada ciclo**
- 3) Calcular o tempo da CPU**

CPI ou Ciclos por Instrução

Considere um programa com 1000 somas. Suponha que eu gaste 1 ciclo de máquina para cada soma.

Se a máquina opera a 100 MHz, quanto tempo o programa demora?

Como eu gasto 1 ciclo para a execução de cada instrução (ou cada soma), irei Gastar um total de 1000 ciclos:

$$\text{Num. Ciclos} = 1000 * 1 = 1000$$

CPU time = número de ciclos * tempo de cada ciclo

$$\text{CPU time} = 1000 * 1 / (100 \times 10^6) = \frac{1000}{100 \times 10^6} = 1000 \times 10^{-8} \text{ s} = 10 \times 10^{-6} \text{ s} = 10 \text{ us}$$

E se eu gastasse 3 ciclos para cada uma das instruções ?

IC (Instruction Count) ou qtd. de instruções

IC representa a quantidade de instruções que um determinado programa possui.

Finalmente:

$$\text{CPU time} = \text{IC} * \text{CPI} * \text{tempo de clock}$$

Ou

$$\text{CPU time} = \text{IC} * \text{CPI} * \frac{1}{\text{frequência de clock}}$$

$$\text{CPU time} = \text{IC} * \text{CPI} * \text{tempo de clock}$$

De acordo com a fórmula da performance da CPU, temos então 3 fatores:

IC - função do algoritmo utilizado, do compilador e da arquitetura do conjunto de instruções da máquina;

CPI - função da organização da máquina e da arquitetura do conjunto de instruções

Tempo de clock - função do Hardware/Organização da máquina

Considere uma máquina e 2 programas diferentes que serão executados. O primeiro possui 2000 instruções e o segundo possui 3000 instruções.

Qual o tempo de execução de cada um considerando um CPI de 5 e a frequência da máquina de 100 MHz.

Qual o speedup?

CPI médio

O que acontece é que normalmente um programa possui diferentes tipos de instruções. Dessa forma podemos calcular um CPI médio.

Considere o seguinte problema:

Em uma máquina operando a 100MHz, irei executar um programa que possui 2 tipos de instruções, instruções aritméticas e instruções de desvio.

O programa possui 2000 instruções, onde 75% são as instruções aritméticas o restante corresponde às instruções de desvio.

Se o CPI para as instruções aritméticas for 3 e para as outras instruções for 5, qual o tempo de CPU necessário para a execução do programa?

Podemos calcular o tempo para a execução de cada um dos diferentes tipos de instrução ou seja:

$$\text{CPU time}_{\text{aritméticas}} = 3 \cdot 1500 \cdot 10\text{ns} = 45 \text{ us}$$

$$\text{CPU time}_{\text{desvios}} = 5 \cdot 500 \cdot 10\text{ns} = 25 \text{ us}$$

$$\text{CPU time} = 45\text{us} + 25\text{us} = 70 \text{ us}$$

Ou

$$\text{CPU time} = 3 \cdot 1500 \cdot 10\text{ns} + 5 \cdot 500 \cdot 10\text{ns} = (3 \cdot 1500 + 5 \cdot 500) \cdot 10\text{ns} = 70 \text{ us}$$

Que é o mesmo que : **CPU time = CPU_{num ciclos} * tempo de clock**

$$\text{CPU}_{\text{num ciclos}} = \sum_{i=1}^n \text{CPI}_i * \text{IC}_i$$

CPI_i = Ciclos por instrução i

IC_i = quantidade de instruções i

Um projetista de compilador deseja decidir entre duas possíveis seqüências de código para a resolução de um problema dados os tipos de instruções e o n° de ciclos por instrução de cada tipo, qual seqüência é mais rápida?

tipo de instrução	CPI
A	1
B	2
C	3

código	n° de instruções		
	tipo A	tipo B	tipo C
1	2	1	2
2	4	1	1

O código 1 executa $2 + 1 + 2 = 5$ instruções

O código 2 executa $4 + 1 + 1 = 6$ instruções

$$\text{nº de ciclos de clock do programa} = \sum_{i=1}^n (CPI_i \times C_i)$$

Nº ciclos de clock para código 1 = $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$ ciclos

Nº ciclos de clock para código 2 = $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$ ciclos

$$CPI = \frac{\text{ciclos de clock}}{\text{nº de instruções}}$$

$$CPI \text{ código 1} = 10 / 5 = 2.0$$

$$CPI \text{ código 2} = 9 / 6 = 1.5$$

Código 2 é mais rápido, mesmo que execute uma instrução a mais, pois necessita de 9 ciclos de clock (portanto, tem CPI mais baixo).

Mas também podemos pensar em termos de um CPI médio para a máquina:

$$\text{CPI}_{\text{médio}} = \frac{\sum_{i=1}^n \text{CPI}_i * \text{IC}_i}{\text{IC}} = \sum_{i=1}^n \text{CPI}_i * \frac{\text{IC}_i}{\text{IC}}$$

No exercício anterior:

CPU time = IC * CPI médio * tempo de clock

CPI médio = (3 * 1500/2000 + 5*500/2000) = 2.25 + 1.25 = 3.5

CPU time = 2000 * 3.5 * 10 ns = 70000 ns = 70 us

Considere o seguinte problema:

Em uma máquina operando a 100MHz, irei executar um programa que possui 2 tipos de instruções, instruções aritméticas e instruções de desvio.

O programa possui 10000 instruções, onde 60% são as instruções aritméticas o restante corresponde às instruções de desvio.

O CPI para as instruções aritméticas é 4 e para as outras instruções for 5.

A) Qual o CPI médio da máquina

B) Considere um novo hardware onde o CPI para as instruções aritméticas passe a ser 3. Qual o speedup sobre a máquina original?

Benchmarks

Workload:

- **É o conjunto de programas que um usuário qualquer costuma rodar**
- **Para comparar o desempenho de duas máquinas A e B bastaria rodar o mesmo workload em ambas**
- **A maioria dos usuários não possui um workload**
- **Cada classe de usuário está interessado em determinados tipos de programas**

Benchmarks

Benchmarks:

- São programas desenvolvidos especialmente para serem utilizados como exemplos para a medida de desempenho
- Tentam reproduzir o comportamento de workloads reais de determinados grupos de usuários
- Geralmente, são de domínio público
- Procuram explorar repertório de recursos da arquitetura
- Benchmarks usuais: SPEC (SPECint e SPECfp), Dhrystone, Whetstone,...

Problemas com benchmarks

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

Intel: tem sua própria medida de desempenho (o índice iComp)

SPEC : System Performance Evaluation Corporation

Consórcio de empresas, criado em 1989, com o objetivo de melhorar a medida e as informações disponíveis sobre o desempenho de processadores e computadores
(<http://www.spec.org>)

Membros do consórcio SPEC

*3DLabs * Acer Inc. * **AMD** * **Apple** * **ATI** * Azul
Systems * BEA Systems * Borland * Bull * **Dell** *
EMC * Exanet * Fabric* Freescale * Fujitsu * **HP** *
Hitachi * **IBM** * **Intel** * ION * JBoss * **Microsoft** *
Mirapoint * **NEC** * Network Appliance * Novell *
NVIDIA * Openwave Systems * **Oracle** * Panasas *
PathScale * Portland Group * S3 Graphics Co., Ltd.
* SAP AG * SGI * **Sun*** Super Micro * Sybase *
Symantec* Unisys * Verisign **

Programas do SPEC 95 (1)

- 8 programas inteiros na linguagem C (medida SPECint95)

go : Inteligência Artificial, joga o Go

M88ksim: Simulador do chip Motorola 88k (executa um programa de teste)

gcc: Compilador C Gnu gerando código para o SPARC

compress: Compressão e descompressão de arquivos na memória

li : Interpretador Lisp

ijpeg: Compressão e descompressão gráficas

perl: Manipula strings e números primos na linguagem Perl

vortex: Programa de banco de dados

Programas do SPEC 95 (2)

- 10 Programas de ponto flutuante na linguagem F77 (medida SPECfp95):

tomcatv : Programa gerador de malhas

swim: Modelo para águas rasas com uma grade 513 x 513

su2cor: Física quântica (simulação do método de Monte Carlo)

hydro2d: Astrofísica (equações hidrodinâmicas de Navier Stokes)

mgrid: Potenciais de campo (solução em 3-D)

applu: Equações diferenciais parciais parabólicas e elípticas

trub3d : Simula uma turbulência isotrópica homogênea em um cubo

apsi : Resolve problemas envolvendo temperatura, vento e poluição

fpppp : Química quântica

wave5: Física do plasma (simulação de partícula eletromagnética)

Vamos montar diferentes benchmarks

Tipo de instruções	B1	B2	B2	B3
Instruções da ALU	40%			
Instruções de desvio	30%			
Instruções de acesso à memória	20%			
Outras	10%			

Exercício:

Considere uma máquina funcionando a 100 MHz e o Benchmark B1.

Considere ainda a tabela abaixo com os CPIs de cada tipo de instrução.

Tipo de instruções	CPI
Instruções da ALU	4
Instruções de desvio	3
Instruções de acesso à memória	5
Outras	6

- a) Qual o CPI médio da máquina ?
- b) Suponha um overclock de 12%. Qual o speedup sobre a máquina original ?
- c) Suponha uma alteração no Hardware e no acesso à memória. Essa alteração reduz em dois ciclos as instruções da ALU ao custo de aumentar em 1 ciclo os acessos à memória. Qual o speedup sobre a máquina original?
- d) Considere um novo compilador que reduza em 50% as instruções da ALU. Qual o speedup sobre a máquina original?
- e) Qual a melhor alteração, b, c ou d ?

