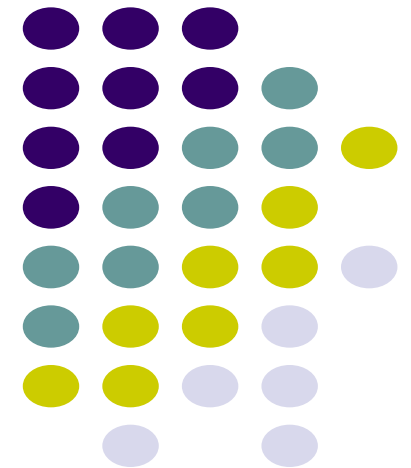
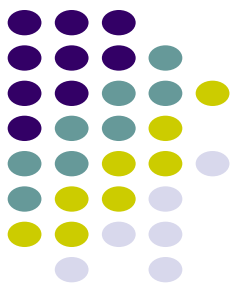


Arquitetura e Organização de Computadores

Avaliando e Compreendendo o Desempenho

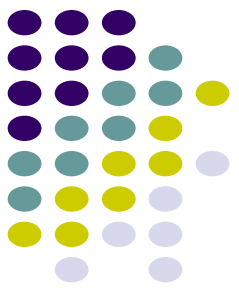
Prof. Sílvio Fernandes





Introdução

- Uma importante razão para examinar o desempenho de um HW é que normalmente é fundamental para a eficiência de um sistema inteiro, incluindo HW e SW
 - Compreender por que um SW funciona de uma determinada maneira
 - Por que um conjunto de instruções pode ser implementado para funcionar melhor que outro
 - Ou como algum recurso de HW afeta o desempenho
- Precisamos entender o que determina o desempenho em um computador

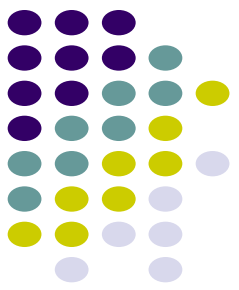


Definindo Desempenho

- Quando falamos que um computador possui melhor desempenho que outro, o que queremos dizer?
 - Exemplo análogo

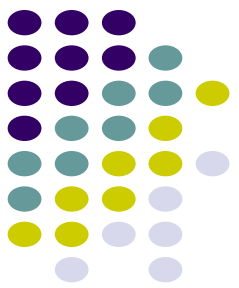
Avião	Capacidade de passageiros	Autonomia de vôo (milhas)	Velocidade de vôo (milhas por hora)	Vazão de passageiros (passageiros × m.p.h.)
Boeing 777	375	4.630	610	228.750
Boeing 747	470	4.150	610	286.700
BAC/Sud Concorde	132	4.000	1350	178.200
Douglas DC-8-50	146	8.720	544	79.424

- Maior velocidade: Concorde
- Maior autonomia: DC-8
- Maior capacidade: 747



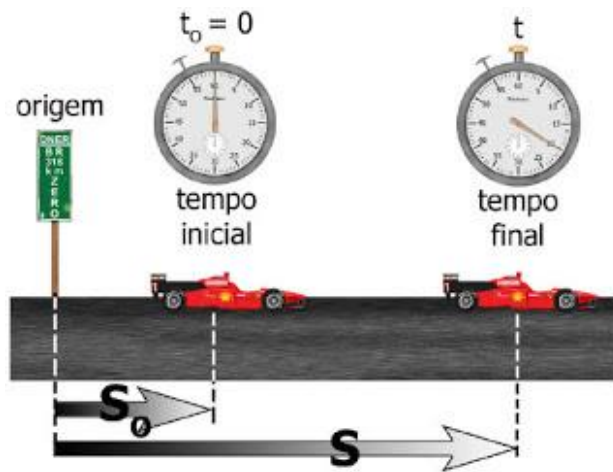
Definindo Desempenho

- Se você estivesse executando um programa em **2 desktops diferentes**, diria que o mais rápido é o computador que termina primeiro.
- Se você estivesse gerenciando um CPD com **vários servidores** realizando tarefas submetidas por muitos usuários, você diria que o computador mais rápido é aquele que completou a maior quantidade de tarefas durante um dia

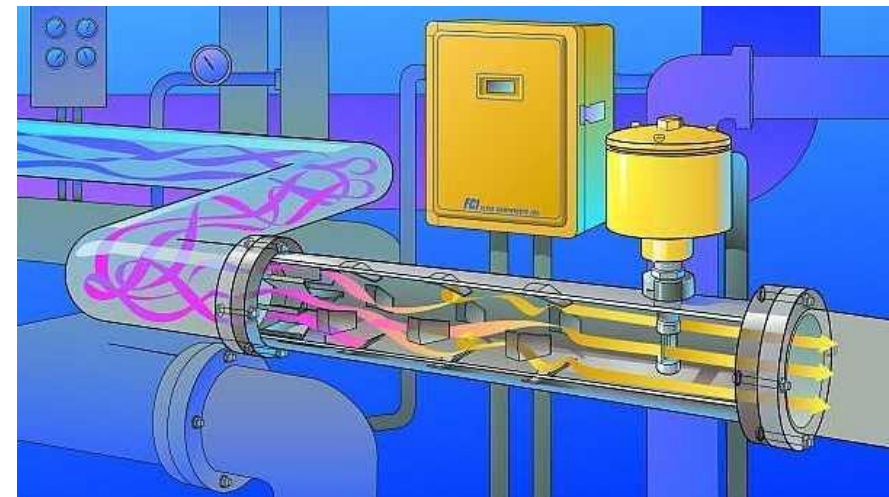


Definindo Desempenho

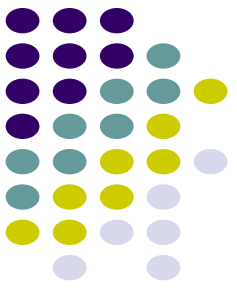
- Usuários de PC estão interessados em **tempo de resposta** ou **latência** ou **tempo de execução**: tempo entre o início e o término de uma tarefa
- Gerentes de CPD estão interessados em **vazão** ou **throughput**: a quantidade total de trabalho feito em um determinado tempo



Tempo de Resposta

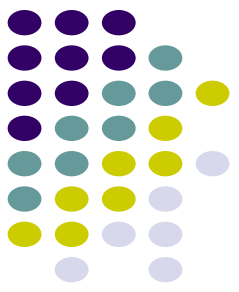


Vazão



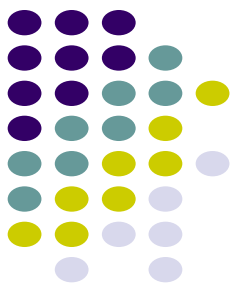
Definindo Desempenho

- Dependendo do caso precisamos de diferentes métricas
- Benchmarks SPEC
 - SPEC CPU para desempenho de CPU
 - SPECweb para servidores Web



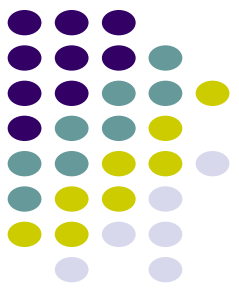
Definindo Desempenho

- Vazão e Tempo de Resposta
 - As seguintes mudanças em um sistema computacional aumentam a vazão, diminuem o tempo de resposta ou as duas coisas?
 - Substituir o processador em um computador por uma versão mais rápida
 - Incluir processadores adicionais em um sistema que usa múltiplos processadores para tarefas distintas, como busca na Web
 - Diminuir o tempo de resposta quase sempre melhora a vazão
 - No caso 1, tanto tempo de resposta quanto vazão são melhorados



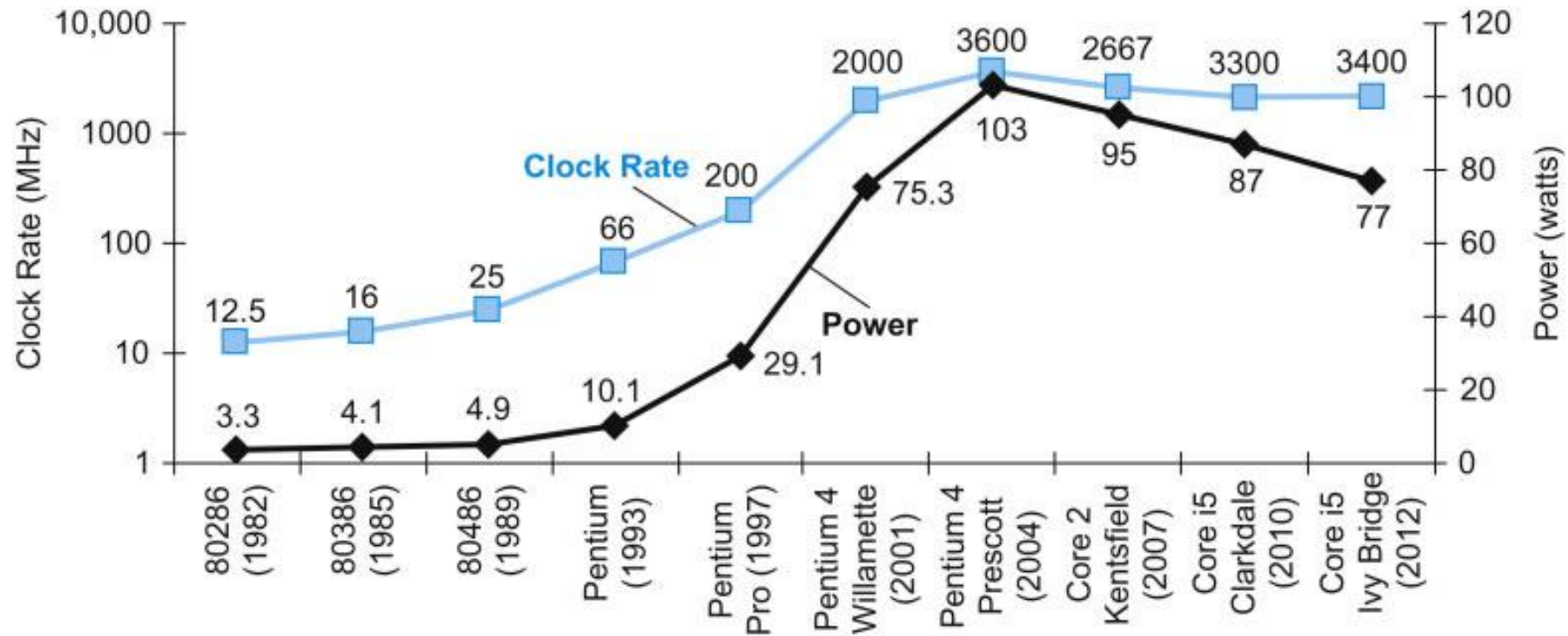
Definindo Desempenho

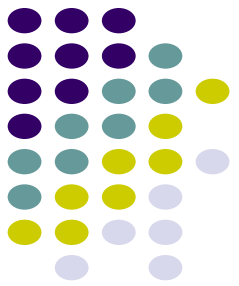
- Vazão e Tempo de Resposta
 - No caso 2, como nenhuma tarefa é realizada primeiro, apenas a vazão é melhorada
 - Mas se a demanda por processamento fosse tão grande quanto a vazão, o sistema poderia forçar as requisições a se enfileirar e, neste caso, aumentar a vazão poderia melhorar o tempo de resposta



Desempenho de processadores

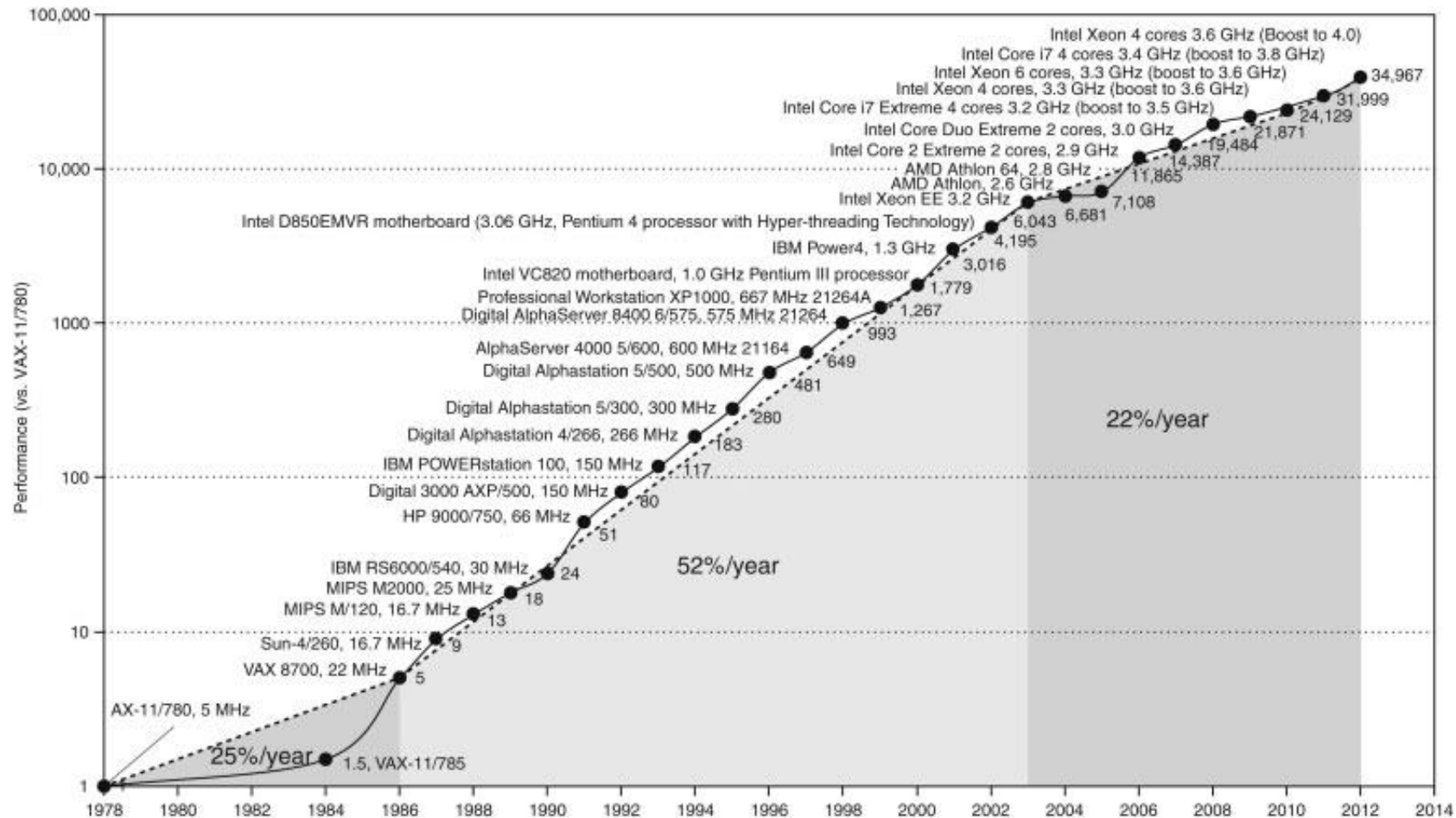
- Taxa de clock e potência dos processadores x86 da Intel em 25 anos

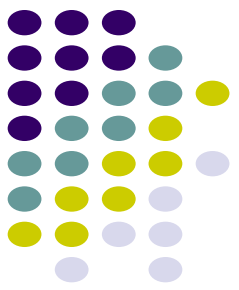




Desempenho de processadores

- Taxa de crescimento do desempenho dos processadores





Definindo Desempenho

- Podemos relacionar o desempenho com o tempo de execução para um computador X:

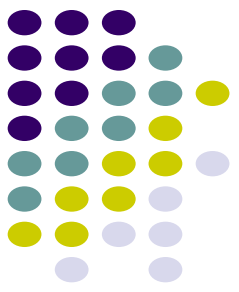
$$Desempenho_x = \frac{1}{Tempo\ de\ execu\c{c}\tilde{o}ao_x}$$

- Se o desempenho de X é maior do que o de Y:

$$Desempenho_x > Desempenho_y$$

$$\frac{1}{Tempo\ de\ execu\c{c}\tilde{o}ao_x} > \frac{1}{Tempo\ de\ execu\c{c}\tilde{o}ao_y}$$

$$Tempo\ de\ execu\c{c}\tilde{o}ao_y > Tempo\ de\ execu\c{c}\tilde{o}ao_x$$



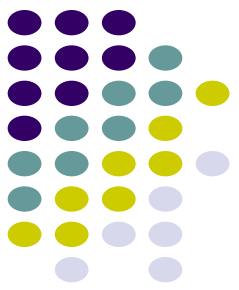
Definindo Desempenho

- Usaremos a frase “X é n vezes mais rápido do que Y” ou “X é n vezes tão rápido quanto Y” para representar

$$\frac{\text{Desempenho}_x}{\text{Desempenho}_y} = n$$

- Se X é n vezes mais rápido do que Y então

$$\frac{\text{Desempenho}_x}{\text{Desempenho}_y} = \frac{\text{Tempo de Execução}_y}{\text{Tempo de Execução}_x} = n$$



Definindo Desempenho

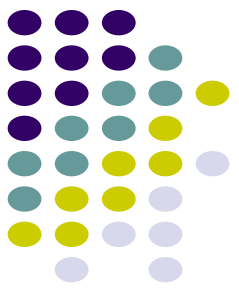
- Se um computador A executa um programa em 10s e o B executa o mesmo programa em 15s, quanto A é mais rápido do que B?

$$\frac{\text{Desempenho}_A}{\text{Desemepnho}_B} = \frac{\text{Tempo de Execução}_B}{\text{Tempo de Execução}_A} = n$$

- Logo o fator de desempenho é

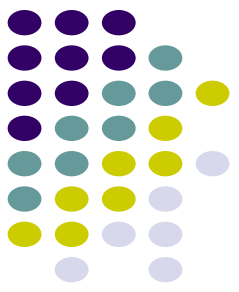
$$\frac{15}{10} = 1,5$$

- Portanto, A é 1,5 vezes mais rápido do que B



Medindo Desempenho

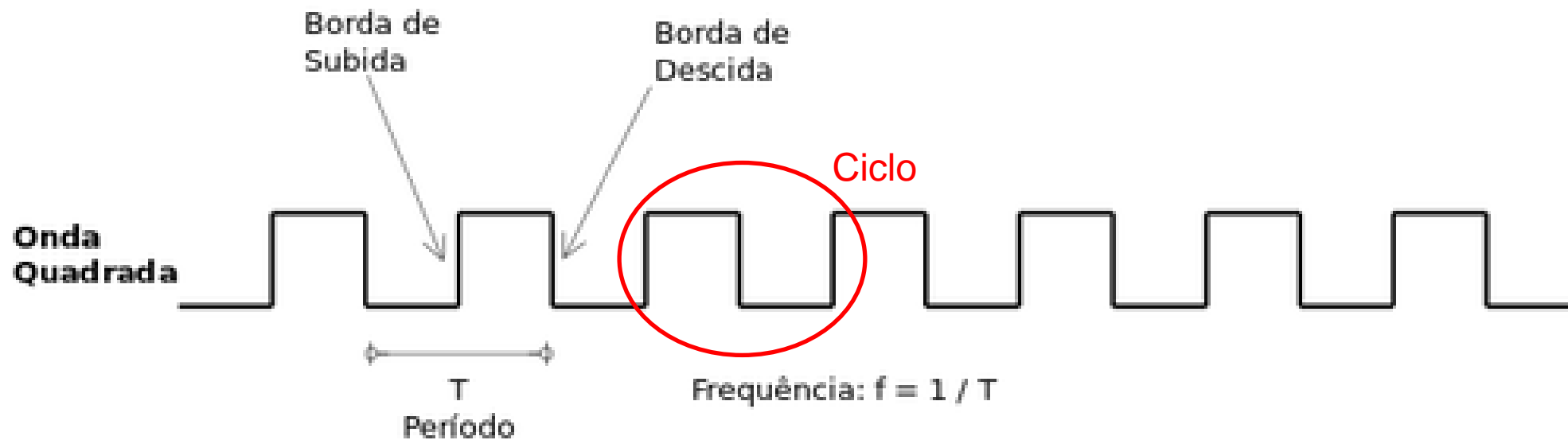
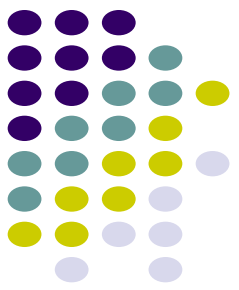
- O tempo é a medida de desempenho dos computadores
 - O computador que realiza a mesma quantidade de trabalho no menor tempo é o mais rápido
- O tempo de execução é medido em **segundos/programa**
- Tempo de execução na CPU ou tempo de CPU
 - Tempo real que a CPU gasta computando para uma tarefa específica (não inclui E/S ou execução de outros programas)
- Tempo de CPU do usuário
 - O tempo de CPU gasto efetivamente em um programa
- Tempo de CPU do sistema
 - O tempo de CPU gasto no SO realizando tarefas a pedido do programa

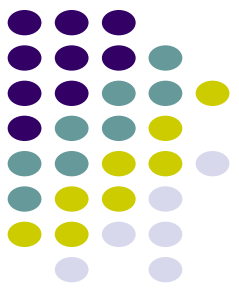


Medindo Desempenho

- Quase todos os computadores são construído usando-se um *clock* que determina quando os eventos ocorrem no HW
 - Intervalos de tempo discretos são **ciclos de clock**
 - Extensão de um **período de clock** é o tempo para um ciclo completo (ex: 250 picossegundos ou 250 ps)
 - **Taxa de clock** é o inverso do período do clock (ex: 4 gigahertz ou 4 GHz)

Medindo Desempenho





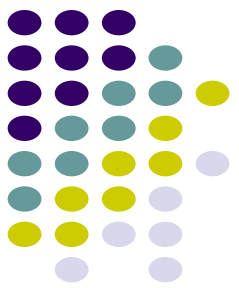
Desempenho da CPU e seus fatores

- Nesse momento, o desempenho final é o tempo de execução da CPU

$$\text{Tempo de execução da CPU para um programa} = \text{Ciclos de clock da CPU para um programa} \times \text{Tempo do ciclo de clock}$$

- Outra alternativa é que, a velocidade de clock e o tempo do ciclo de clock são inversos

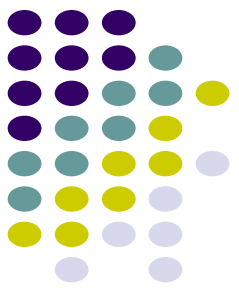
$$\text{Tempo de execução da CPU para um programa} = \frac{\text{Ciclos de clock da CPU para um programa}}{\text{Velocidade de clock}}$$



Desempenho da CPU e seus fatores

- **Exemplo**

- Nosso programa favorito é executado em **10 s** no computador **A**, que possui um clock de **2 GHz**. Estamos tentando ajudar um projetista a construir um computador **B** que execute esse programa em **6 segundos**. O projetista determinou que um aumento substancial na velocidade de clock é possível, mas esse aumento afetará o restante do projeto da CPU, fazendo com que o computador **B** exija **1,2 vezes mais ciclos de clock do que o A** para esse programa. Que velocidade de clock devemos dizer para o projetista buscar?



Desempenho da CPU e seus fatores

- **Exemplo**

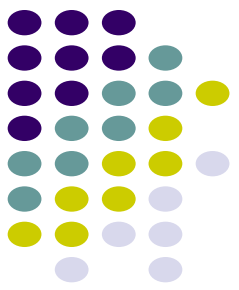
$$\text{Tempo de CPU}_A = \frac{\text{Ciclos de clock da CPU}_A}{\text{Velocidade de clock}_A}$$

$$10 \text{ segundos} = \frac{\text{Ciclos de clock da CPU}_A}{2 \times 10^9 \text{ ciclos/segundo}}$$

$$\text{Ciclos de clock da CPU}_A = 10 \text{ segundos} \times 2 \times 10^9 \text{ ciclos/segundo} = 20 \times 10^9 \text{ ciclos}$$

- O tempo de CPU de B pode ser encontrado usando

$$\text{Tempo de CPU}_B = \frac{1,2 \times \text{Ciclos de clock da CPU}_A}{\text{Velocidade de clock}_B}$$



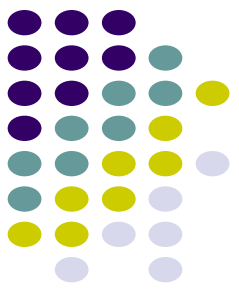
Desempenho da CPU e seus fatores

- **Exemplo**

$$6 \text{ segundos} = \frac{1,2 \times 20 \times 10^9 \text{ ciclos}}{\text{Velocidade de clock}_B}$$

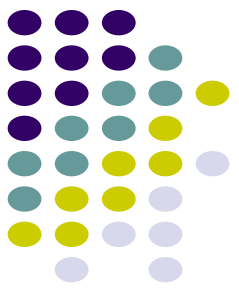
$$\text{Velocidade de clock}_B = \frac{1,2 \times 20 \times 10^9 \text{ ciclos}}{6 \text{ segundos}} = 4 \text{ GHz}$$

- Portanto, a velocidade de clock do computador B deve ser o dobro da velocidade de clock de A para poder executar o programa em 6 segundos



Desempenho da CPU e seus fatores

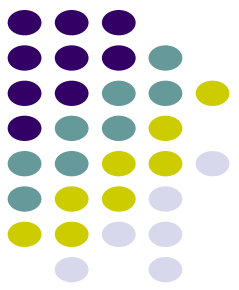
- As equações anteriores não incluem qualquer referência ao número de instruções necessárias para o programa.
- Entretanto, como o compilador claramente gerou instruções para serem executadas, o tempo de execução tem de depender do número de instruções em um programa
- O **tempo de execução** é igual ao **número de instruções** executadas **multiplicado pelo tempo médio gasto por cada instrução**



Desempenho da CPU e seus fatores

$$\text{Ciclos de clock da CPU} = \text{Instruções para um programa} \times \text{Ciclos de clock médios por instrução}$$

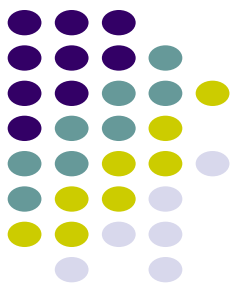
- O termo **ciclos por instrução**, que é o número médio de ciclos de clock que cada instrução gasta para ser executada, normalmente é abreviado como CPI
- O CPI fornece uma maneira de comparar duas implementações diferentes do mesmo conjunto de instruções, já que a contagem de instruções necessária para um programa será, evidentemente, a mesma



Desempenho da CPU e seus fatores

- **Exemplo**

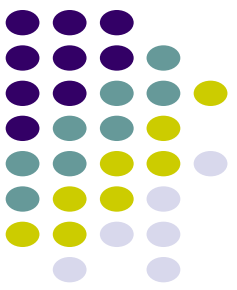
- Vamos supor que temos 2 implementações do mesmo conjunto de instruções. O computador **A** tem um tempo de ciclo de clock de **250ps** e um CPI de **2,0** para um determinado programa, e o computador **B** tem um tempo de ciclo de clock de **500ps** e um CPI de **1,2** para o mesmo programa. **Que computador é mais rápido** para esse programa e o **quanto mais rápido?**
- Sabemos que cada computador executa o mesmo número de instruções para o programa; vamos chamar esse número de I



Desempenho da CPU e seus fatores

● Exemplo

- Primeiro, encontre o número de ciclos de clock do processador para cada computador
 - Ciclos de clock da CPU_A = $1 \times 2,0$
 - Ciclos de clock da CPU_B = $1 \times 1,2$
- Agora podemos calcular o tempo de CPU para cada computador
 - Tempo de CPU_A = Ciclos de clock_A x tempo de clock_A
 - Tempo de CPU_A = $1 \times 2,0 \times 250 \text{ ps} = 500 \times 1 \text{ ps}$
- Da mesma forma, para B:
 - Tempo de CPU_B = Ciclos de clock_B x tempo de clock_B
 - Tempo de CPU_B = $1 \times 1,2 \times 500 \text{ ps} = 600 \times 1 \text{ ps}$



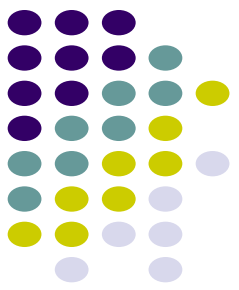
Desempenho da CPU e seus fatores

- **Exemplo**

- Claramente, o computador A é mais rápido
- O quanto mais rápido é calculado pela razão:

$$\frac{\text{Desempenho da CPUA}}{\text{Desemepnho da CPUB}} = \frac{\text{Tempo de ExecuçãoB}}{\text{Tempo de ExecuçãoA}} = \frac{600 \times 1 \text{ ps}}{500 \times 1 \text{ ps}} = 1,2$$

- Podemos concluir que A é 1,2 vezes mais rápido que B para esse programa



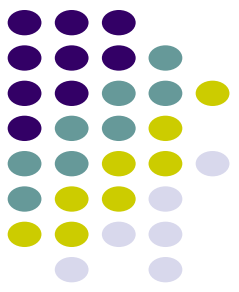
Desempenho da CPU e seus fatores

- Agora podemos escrever essa equação de desempenho básica em termos de contagem de instruções, de CPI e de tempo do ciclo de clock

$$\text{Tempo de CPU} = \text{Contagem de instruções} \times \text{CPI} \times \text{Tempo do ciclo de clock}$$

- Ou ainda

$$\text{Tempo de CPU} = \frac{\text{Contagem de instruções} \times \text{CPI}}{\text{Velocidade de clock}}$$

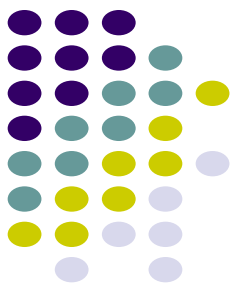


Desempenho da CPU e seus fatores

- Algumas vezes, é possível calcular os ciclos da CPU olhando os diferentes tipos de instruções e usando suas contagens de ciclos de clock individuais

$$\text{Ciclos de clock da CPU} = \sum_{i=1}^n (CPI_i \times C_i)$$

- Onde C_i é a contagem do número de instruções da classe i executadas, CPI_i é a média dos ciclos por instrução para essa classe e n é o número de classes de instrução



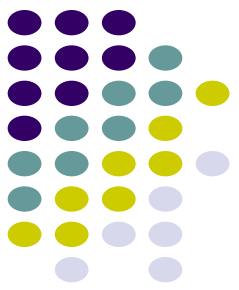
Desempenho da CPU e seus fatores

- **Exemplo:** Um projetista de compilador está tentando decidir entre 2 sequências de código para um determinado computador. Os projetista de HW forneceram os seguintes fatos:

	CPI para esta classe de instrução		
	A	B	C
CPI	1	2	3

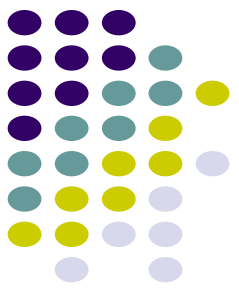
- Para uma determinada instrução em linguagem de alto nível

Sequência de código	Contagem de instrução para classe de instrução		
	A	B	C
1	2	1	2
2	4	1	1



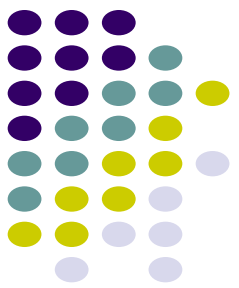
Desempenho da CPU e seus fatores

- **Exemplo:** Qual sequência de código executa mais instruções? Qual será mais rápida? Qual é o CPI para cada sequência?
 - A seq. 1 executa $2+1+2 = 5$ instruções
 - A seq. 2 executa $4+1+1 = 6$ instruções
 - Logo, a seq. 2 executa mais instruções
 - Para encontrar o número total de ciclos de clock para cada sequência:
 - $\text{Ciclos de clock da CPU} = \sum_{i=1}^n (\text{CPI}_i \times C_i)$
 - Ciclos de clock da CPU₁ = $(2 \times 1) + (1 \times 2) + (2 \times 3) = 10$ ciclos
 - Ciclos de clock da CPU₂ = $(4 \times 1) + (1 \times 2) + (1 \times 3) = 9$ ciclos



Desempenho da CPU e seus fatores

- **Exemplo:** Qual sequência de código executa mais instruções? Qual será mais rápida? Qual é o CPI para cada sequência?
 - Logo, a seq. 2 é a mais rápida, mesmo que ela executa uma instrução extra
 - Os valores de CPI podem ser calculados:
 - $CPI = \frac{\text{Ciclos de clock da CPU}}{\text{Contagem de instruções}}$
 - $CPI_1 = 10 / 5 = 2$
 - $CPI_2 = 9 / 6 = 1,5$
 - Quando estiver comparando 2 computadores, você precisa considerar TODOS os 3 componentes, que se combinam para formar o tempo de execução

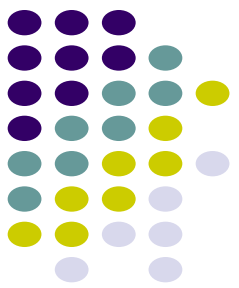


Desempenho da CPU e seus fatores

- Podemos ver como esses fatores são combinados para fornecer o tempo de execução medido em segundos por programa

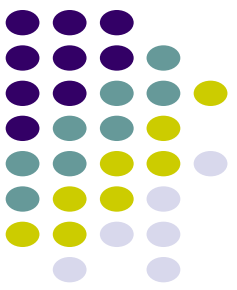
$$\text{Tempo} = \frac{\text{segundos}}{\text{programa}} = \frac{\text{instruções}}{\text{programa}} \times \frac{\text{ciclos de clock}}{\text{instrução}} \times \frac{\text{segundos}}{\text{ciclo de clock}}$$

Componentes do desempenho	Unidade de medida
Tempo de execução da CPU para um programa	Segundos para o programa
Contagem de instruções	Instruções executadas para o programa
Ciclos de clock por instrução (CPI)	Número médio de ciclos de clock por instrução
Tempo de ciclo de clock	Segundos por ciclo de clock



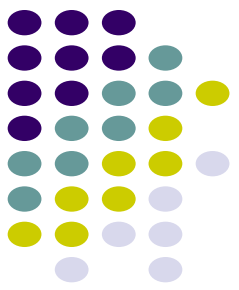
Desempenho da CPU e seus fatores

Componente de HW ou SW	Afeta o quê?
Algoritmo	Contagem de instruções, possivelmente CPI
Linguagem de programação	Contagem de instruções, CPI
Compilador	Contagem de instruções, CPI
Arquitetura do conjunto de instruções	Contagem de instruções, taxa de clock, CPI



Falácias e Armadilhas

- **Armadilha:** *Esperar que a melhoria de um aspecto de um computador aumente o desempenho geral por uma quantidade proporcional ao tamanho da melhoria*
- **Exemplo:**
 - Suponha que um programa execute em **100 s** em um computador, com operações de multiplicação responsáveis por **80 s** desse tempo. Quanto terei de melhorar a velocidade da multiplicação se eu quiser que meu programa execute **5 vezes mais rápido**?



Falácias e Armadilhas

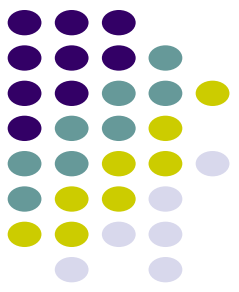
- Podemos usar a Lei de Amdahl
 - A melhoria de desempenho possível com determinado aprimoramento é limitada pela quantidade com que o recurso aprimorado é utilizado.
- Então temos

$$\frac{\text{Tempo de execução após aprimoramento}}{\text{Quantidade de aprimoramento}} + \text{Tempo de execução não afetado}$$

- Para este problema

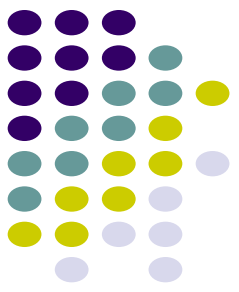
$$\text{Tempo de execução após aprimoramento} = \frac{80}{n} + (100 \text{ s} - 80 \text{ s})$$

$$20 \text{ s} = \frac{80 \text{ s}}{n} + (20 \text{ s}) \quad \rightarrow \quad 0 = \frac{80 \text{ s}}{n}$$



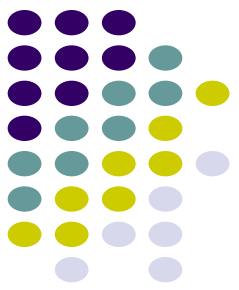
Verifique você mesmo

1. Suponha que queremos saber se uma aplicação que usa o desktop de um cliente e um servidor remoto está limitada pelo desempenho da rede. Para as mudanças a seguir, diga se apenas a vazão melhora, se o tempo de resposta e a vazão ou se nenhum dos dois melhora
 - a) Um canal de rede extra é acrescentado entre o cliente e o servidor, aumentando a vazão total da rede e reduzindo o atraso para obter acesso à mesma (já que agora há dois canais)



Verifique você mesmo

- b) O software de rede é melhorado, reduzindo, assim, o atraso da comunicação via rede, mas não melhorando a vazão
 - c) Mais memória é acrescentada ao computador
2. O desempenho do computador C é 4 vezes melhor do que o desempenho do computador B, que executa uma determinada aplicação em 28 segundos. Quanto tempo o computador C levará para executar essa aplicação?

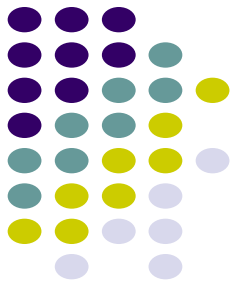


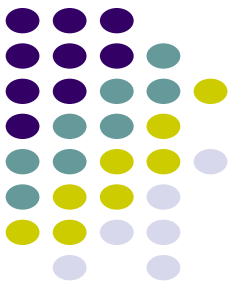
Verifique você mesmo

3. Uma determinada aplicação Java roda 15 s em um processador de desktop. É lançado um novo compilador Java que exige apenas seis décimos da quantidade de instruções do antigo compilador. Infelizmente, ele aumenta o CPI em 1,1. Em que velocidade podemos esperar que a aplicação seja executada usando esse novo compilador?
- a) $(15 \times 0,6) / 1,1 = 8,2 \text{ s}$
 - b) $15 \times 0,6 \times 1,1 = 9,9 \text{ s}$
 - c) $(15 \times 1,1) / 0,6 = 27,5 \text{ s}$

Respostas

1. a: ambos; b: tempo de resposta; c: nenhum
2. 7 segundos
3. b





Referências

- PATTERSON, D. A. ; HENNESSY, J.L. Organização e projeto de computadores – a interface hardware software. 3. ed. Editora Campus, 2005.
- STALLINGS, W. Arquitetura e organização de computadores: projeto para o desempenho. 8. ed. Prentice Hall, 2009.