

Aula 2

Organização de Computadores

Profa. Débora Matos

Desempenho em computadores

Como medir o desempenho?

- Desempenho poderia ser associado ao computador que termina primeiro a execução de um programa;
- Ou pode ser associado a um servidor que executa mais tarefas por um determinado tempo (*throughput*).

Desempenho em computadores

Como medir o desempenho?

Para maximizar o desempenho é preciso minimizar o tempo de execução para uma determinada tarefa. Assim, o desempenho para o computador A :

$$Desempenho_A = \frac{1}{Tempo\ de\ Execução_A}$$

Desempenho em computadores

Como medir o desempenho?

Mas quantas vezes o computador A é mais rápido do que o computador B?

$$\frac{\text{desempenho}_A}{\text{desempenho}_B} = n$$

Dizemos que A é n vezes mais rápido do que B.

Desempenho em computadores

Exemplo 1: Se o computador A executa um programa em 7 segundos e o computador B executa o mesmo programa em 10 segundos, tem-se que:

$$\frac{\textit{desempenho}_A}{\textit{desempenho}_B} = \frac{\textit{tempo de execu\c{c}\~ao}_B}{\textit{tempo de execu\c{c}\~ao}_A} = \frac{10}{7} = 1,43$$

Dizemos que A é 1,43 vez mais rápido do que B.

Desempenho em computadores

O tempo é a medida de desempenho do computador: o computador que realiza a mesma quantidade de tarefa no menor tempo é **mais rápido**.

O tempo de resposta, ou tempo de relógio ou ainda tempo decorrido corresponde ao tempo total para completar uma tarefa.

No entanto, esse tempo refere-se exatamente ao que??

Nesse valor de tempo estão incluídos o tempo de acesso ao disco, acessos à memória, atividades de E/S, overhead do sistema operacional, etc.

Desempenho em computadores

- Além disso, o computador normalmente executa **diversos programas simultaneamente**, sendo assim, este tempo não reflete o tempo de execução de apenas 1 programa.
- Nesses casos, vale mais analisar o throughput em vez do tempo decorrido para a execução de apenas um programa.
- É preciso distinguir entre o tempo decorrido e o tempo em que o processador está trabalhando na execução de um programa específico.
- **Tempo de CPU** é o tempo gasto apenas na execução do programa.

Desempenho de CPU

Calculando o tempo de execução da CPU de um programa:

tempo de CPU = ciclos de clock da CPU x Tempo de ciclo do clock

$$\text{tempo de CPU} = \frac{\text{ciclos de clock da CPU para um programa}}{\text{frequência de operação}}$$

Sendo assim, é possível melhorar o desempenho de um programa reduzindo o n° de ciclos de clock ou a frequência de operação da CPU.

Como podemos reduzir o número de ciclos de clock de um programa?



ORCJogo

Cite pelo menos 3 possibilidades
para reduzir o tempo de
execução de um determinado
programa:

Desempenho de CPU

Exemplo 2: Um determinado programa é executado em 8 segundos no computador *X*, que tem uma taxa de clock de 1.5GHz. Um projeto de um computador *Y* está sendo desenvolvido para executar o mesmo programa em apenas 5 segundos. No entanto, o projetista verificou que para este computador, o programa vai precisar de 1,4 vez a quantidade de ciclos de clock do computador *X*. Qual é taxa de clock mínima aceitável para esse projeto?

$$\text{tempo de CPU} = \frac{\text{ciclos de clock da CPU para um programa}}{\text{frequência de operação}}$$

Desempenho da Instrução

- Outra forma de medir o desempenho é defini-lo a partir do número de instruções de um programa.
- O número total de instruções de um programa pode ser obtido após a compilação do programa.

$$\text{ciclos da CPU} = \text{instruções} \times \text{ciclos médios por instrução}$$

Ciclos de clock por instrução = CPI

Refere-se ao número médio de ciclos de clock que cada instrução leva para executar.

Desempenho da Instrução

- Equação do desempenho em função do CPI, n° de instruções e tempo de ciclo de clock:

$$\text{ciclos de clock da CPU} = \text{instruções} \times \text{CPI}$$

$$\text{tempo de CPU} = \frac{\text{ciclos de clock da CPU}}{\text{frequência de operação}}$$

$$\text{tempo de CPU} = \frac{\text{instruções} \times \text{CPI}}{\text{frequência de operação}}$$

Desempenho da Instrução

Exemplo 3: Qual computador é mais rápido e por quanto para um determinado programa sabendo que no computador A o tempo de ciclo de clock é de 230ps e CPI igual a 3 e no computador B o tempo de ciclo de clock é de 350ps e um CPI de 1,25.

$$\text{ciclos da CPU} = n^{\circ} \text{ instruções} \times \text{CPI}$$

$$\text{tempo de CPU} = \frac{\text{ciclos da CPU}}{\text{frequência de operação}}$$

Desempenho da Instrução

Exemplo 4: Um projetista de compilador está tentando decidir entre duas sequências de código para um computador. Os projetistas de HW forneceram as seguintes informações:

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

Para determinada instrução na linguagem de alto nível, o escritor do compilador está considerando 2 sequências de código que exigem as seguintes contagens de instruções:

	Contagens de Instrução para cada classe de instrução		
Sequência de código	A	B	C
1	2	1	2
2	4	1	1

Qual a sequência de código executa mais instruções? Qual será mais rápida? Qual CPI médio para cada sequência?

Desempenho da Instrução

Para pensar:

- Mudar o conjunto de instruções de forma a **reduzir a contagem de instruções não é sinônimo de melhoria de desempenho**. Por que?
- Como as instruções podem apresentar CPI diferentes, a redução do tempo de CPU vai depender das instruções utilizadas em cada programa. Assim, **menos instruções em um programa não significa menor tempo de CPU**.

Desempenho da Instrução

Para pensar:

- A contagem de instruções depende da arquitetura e não da organização (não importa como as instruções foram de fato implementadas).
- Já o número de ciclos de cada instrução depende do projeto de hardware, ou seja, da organização.

Desempenho da Instrução

Para pensar:

- Além das análises anteriores, de que outros fatores dependem o desempenho de um programa?
 - Algoritmo
 - Linguagem de programação
 - Compilador
 - Arquitetura
 - Organização (HW)

Desempenho da Instrução

Para pensar:

- Qual seria o CPI mínimo na organização de um computador?
- É possível que um CPI seja menor do que 1?
 - Esse valor pode ser menor do que 1 quando o processador permite executar mais de uma instrução em um mesmo ciclo de clock. Técnicas de paralelismo permitem $CPI < 1$. Nesse caso é mais interessante falar de IPC – número de instruções por ciclo de clock.

Lei de Amdahl

A lei de Amdahl indica como uma máquina se comporta depois de certas melhorias comparada a sua versão anterior.

$$\text{TempoEx}_{\text{novo}} = \text{TempoEx}_{\text{velho}} \times \left[(1 - \text{Fração}_{\text{melhorada}}) + \frac{\text{Fração}_{\text{melhorada}}}{\text{Speedup}_{\text{melhorada}}} \right]$$

$$\text{Speedup}_{\text{global}} = \frac{\text{TempoEx}_{\text{velho}}}{\text{TempoEx}_{\text{novo}}} = \frac{1}{(1 - \text{Fração}_{\text{melhorada}}) + \frac{\text{Fração}_{\text{melhorada}}}{\text{Speedup}_{\text{melhorada}}}}$$

TempoEX = tempo de execução

Observa-se que, se for melhorado apenas um aspecto no computador/programa, então a parte que **não** foi melhorada poderá vir a dominar a performance do sistema. Sendo assim, os ganhos sempre serão maiores quando a melhora ocorrer **no caso mais comum**.

Lei de Amdahl

- A melhoria de desempenho possível com determinado aprimoramento é **limitado** pela quantidade com que o recurso aprimorado é utilizado.
- Tornar o caso comum rápido tenderá a melhorar o desempenho mais do que otimizar o caso raro.
- Normalmente para avaliar o desempenho de computadores são utilizados *benchmarks* (ex. SPEC).

```
strcpy:
    subi $sp, $sp, 4
    sw   $s0, 9($sp)
    add  $s0, $zero, $zero
L1:    add $t1, $a1, $s0 ←
    lb   $t2, 0($t1)
    add  $t3, $a0, $s0
    sb   $t2, 0($t3)
    beq  $t2, $zero, L2
    addi $s0, $s0, 1
    j    L1
L2:    lw   $s0, 0($sp)
    addi $sp, $sp, 4
    jr   $ra
```

Medida em MIPS

- MIPS – milhões de instruções por segundo.

$$MIPS = \frac{\text{contagem de instruções}}{\text{tempo de execução} \times 10^6}$$

Qual o problema com a medida de desempenho em MIPS?

- Não pode ser utilizado para comparar computadores com diferentes ISAs (conjunto de instruções);
- O MIPS varia entre os programas no mesmo computador;
- Exemplo: um programa executa mais instruções em uma determinada CPU, o MIPS pode aumentar, mas isso não reflete um aumento de desempenho, pois depende do CPI.

Medida em MIPS

Exemplo 5: Considere as seguintes medidas obtidas para um programa:

Medida	Computador A	Computador B
Número de instruções	10 bilhões	8 bilhões
Taxa de clock	4 GHz	4 GHz
CPI	1,0	1,1

- a) Qual computador tem maior avaliação de MIPS?
- b) Qual o computador é mais rápido?

A barreira da Potência

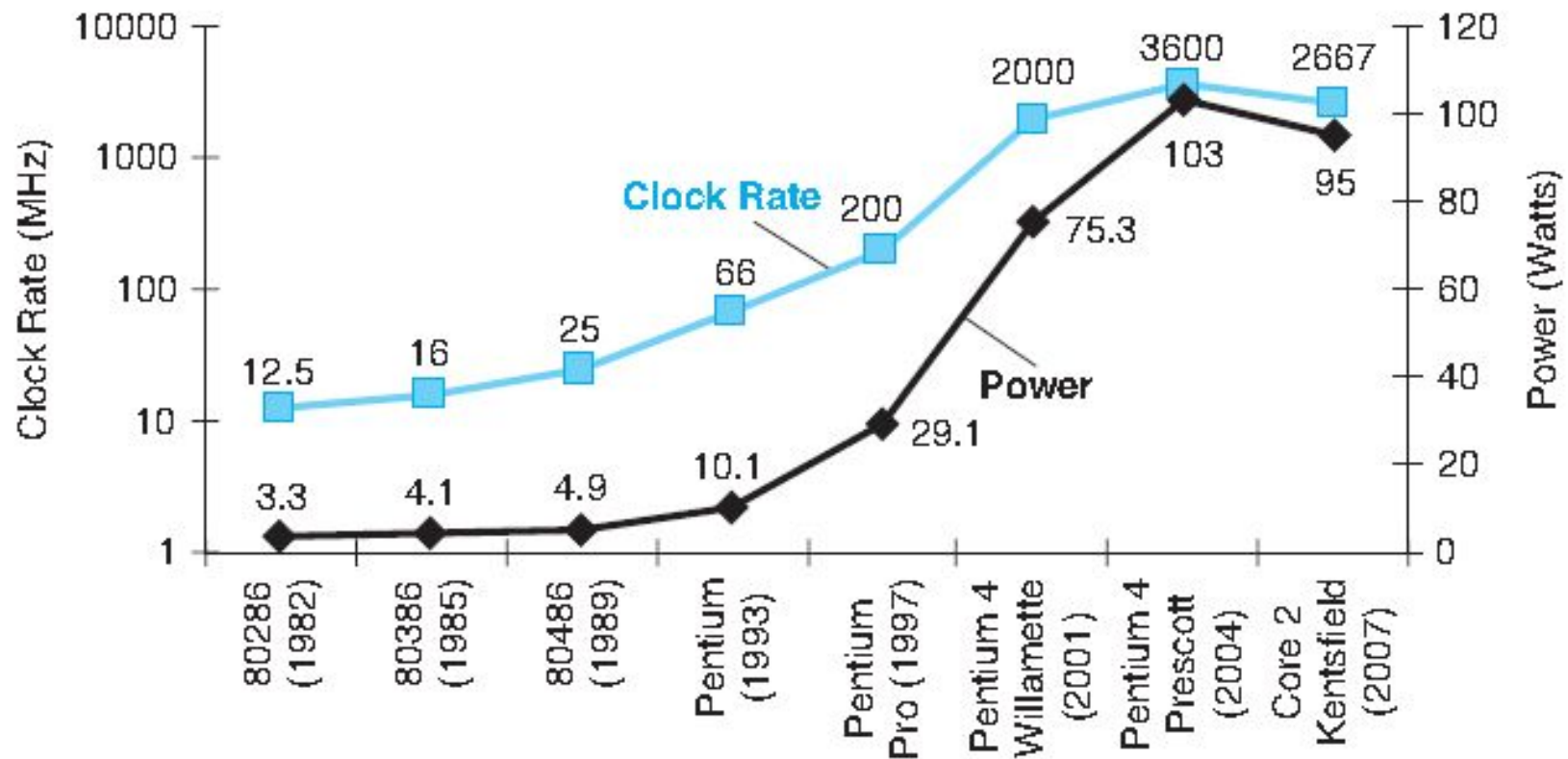


FIGURE 1.15 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip.

A barreira da Potência

- A tecnologia dominante para circuitos integrados é a CMOS (*Complementary Metal Oxide Semiconductor*);
- Em CMOS a principal fonte de dissipação de potência é a potência dinâmica (potência consumida durante o chaveamento).
- A potência dinâmica depende da carga capacitiva de cada transistor, da tensão elétrica e da frequência com que o transistor é comutado.

$$Potência = Carga\ capacitiva \times Tensão^2 \times Frequência$$

Conclusão

Para um conjunto de instruções de uma dada arquitetura, um aumento na performance da CPU pode assim ser obtido:

- Aumentando a frequência do relógio.
- Fazendo melhorias na organização do processador de forma a diminuir o CPI.
- Obtendo otimizações nos compiladores a fim de diminuir o número de instruções ou que geram instruções com CPI médio mais baixo (usando instruções mais simples, por exemplo).

Exercícios

- Verificar os conhecimentos a respeito do desempenho de uma CPU.
- Fazer a lista de exercícios.