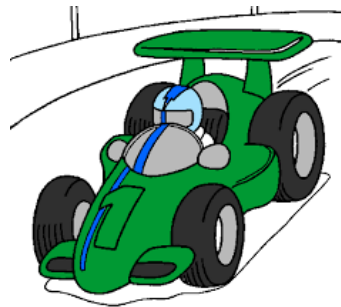


# Performance

GEX 612 - Organização de Computadores

Prof. Luciano L. Caimi  
lcaimi@uffs.edu.br

- Roteiro
  - Latência / tempo de resposta / tempo de execução vs. Throughput
  - Como fazer comparações de performance de forma razoável?
  - Os 3 componentes da Performance da CPU
  - As 2 leis da performance



Porque saber sobre performance:

- **Perspectiva de vendas**

- Dado um conjunto de máquinas, qual possui:
  - Melhor performance?
  - Melhor preço?
  - Melhor custo/benefício

- **Perspectiva de projeto**

- considerando as opções de projeto, qual possui
  - Melhor aumento de performance?
  - Menor custo?
  - Melhor performance/custo?

- **As duas perspectivas requerem**

- Base de comparação
- Métricas de avaliação



# Performance

- Muitas definições são possíveis
- Cada fabricante vai adequar a medida as suas melhores características



# Performance

- Qual tem melhor performance?

<b>Modelo</b>	<b>Passageiros</b>	<b>Autonomia (km)</b>	<b>Velocidade (km/h)</b>	<b>Preço (\$M)</b>
Airbus A310	250	8 300	895	120
Boeing 747	470	6 700	980	200
Boeing 767	250	12 300	885	120
Boeing 777	375	7 450	980	180
Concorde	130	6 400	2 200	350
DC-8-50	145	14 000	875	80



# Performance

- Qual tem melhor performance?

## Do ponto de vista do passageiro?

- tempo de viagem (velocidade e autonomia)
- preço da passagem (passageiros e autonomia)

Modelo	Passageiros	Autonomia (km)	Velocidade (km/h)	Preço (\$M)
Airbus A310	250	8 300	895	120
Boeing 747	470	6 700	980	200
Boeing 767	250	12 300	885	120
Boeing 777	375	7 450	980	180
Concorde	130	6 400	2 200	350
DC-8-50	145	14 000	875	80

## Do ponto de vista da empresa aérea?

- Throughput - medida em: passageiro \* velocidade

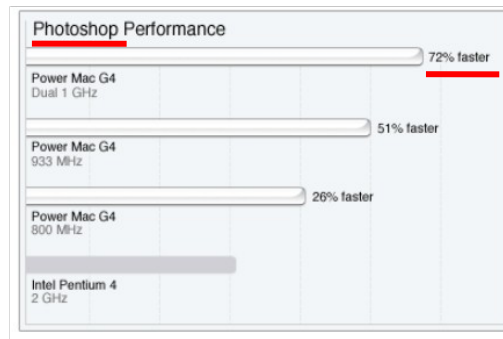
## Do ponto de vista da ANAC/FAA?

- Segurança



# Performance

- Muitas definições são possíveis
- Cada fabricante vai adequar a medida às suas melhores características



**AMD**

**Q:** *Why do end users need a new performance metric?*

**A:** End users who rely only on megahertz as an indicator for performance do not have a complete picture of PC processor performance and may pay the price of missed expectations.

- Desempenho dos computadores: TIME, TIME...
  - Tempo de Resposta (latency)
    - tempo para uma minha tarefa rodar
    - Demora para executar uma tarefa
    - tempo de uma consulta a base de dados
  - Throughput
    - Quantas tarefas a máquina pode rodar cada vez?
    - Qual a taxa média de execução?
    - Quantos trabalhos são concluídos em um período de tempo?



# Performance



- Muitas definições são possíveis
- Cada fabricante vai adequar a medida às suas melhores características (hardware, SO, compilador, etc)
- Necessidade de **Benchmarks**
- Alguns benchmarks usuais:
  - [Adobe Photoshop](#) para processamento de imagem
  - [BAPCo SYSmark](#) para aplicações de escritório
  - [SPEC2000](#) para performance de CPU

# Conceitos de performance e speedup



$$\text{Performance} = \frac{1}{\text{tempo de execução}}$$

- Quanto menor o tempo de execução de um programa maior a sua performance
- Quanto maior a performance de um programa menor é seu tempo de execução

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

- máquina X é n vezes mais rápida que a máquina y para executar o programa  
exemplo: programa executado em 20 segundos na máquina X e em 25 segundos na máquina Y

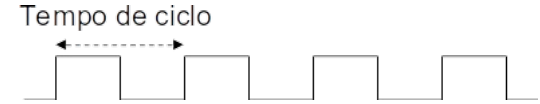


# Conceitos de performance e speedup



$$\text{tempo de execução} = \text{número de ciclos de clock} \cdot \text{tempo de ciclo}$$

$$\text{tempo de ciclo} = \frac{1}{\text{frequência}}$$



O tempo de ciclo (período) é o inverso da frequência do clock (MHz, GHz)

10 nseg tempo de ciclo =>	100 MHz de frequência
5 nseg tempo de ciclo =>	200 MHz de frequência
2 nseg tempo de ciclo =>	500 MHz de frequência
1 nseg tempo de ciclo =>	1 GHz de frequência
500 pseg tempo de ciclo =>	2 GHz de frequência
250 pseg tempo de ciclo =>	4 GHz de frequência
200 pseg tempo de ciclo =>	5 GHz de frequência

Clock: 1 GHz =  $10^9$  ciclos / seg (período = tempo de ciclo =  $10^{-9}$  seg = 1 ns)

200 MHz =  $200 \cdot 10^6$  ciclos / seg (período = tempo de ciclo = 5 ns)

Exemplo: Considere um programa que executa em  $2 \cdot 10^9$  ciclos de clock.

Calcule o tempo de execução do mesmo nas frequências de clock acima

$$\text{tempo de execução} = 2 \cdot 10^9 \cdot 1 \cdot 10^{-9} = 2 \text{ seg}$$

$$\text{tempo de execução} = 2 \cdot 10^9 \cdot 5 \cdot 10^{-9} = 10 \text{ seg}$$

# Conceitos de performance e speedup



$$\textit{tempo de execução} = \textit{número de ciclos de clock} \cdot \textit{tempo de ciclo}$$

Quantos ciclos são requeridos num programa?

Reescrevendo o número de ciclos de clock em função da quantidade de instruções executadas pelo programa temos:

$$\textit{tempo de execução} = \textit{número de instruções} \cdot \frac{\textit{número de ciclos}}{\textit{instrução}} \cdot \textit{tempo de ciclo}$$



# Conceitos de performance e speedup

Quantas instruções são executadas neste trecho de programa?

Cada “for” consiste de duas instruções: incrementar índice, verificar condição de saída

**12,422,450 instruções**

250 instruções

**for i = 1, 100 do**

20 instruções

**for j = 1, 100 do**

40 instruções

**for k = 1, 100 do**

10 instruções

**endfor**

**endfor**

**endfor**

2 + 20 + 124,200 instruções

100 interações

12,422,200 instruções no total

2 + 40 + 1200 instruções

100 interações

124,200 instruções no total

2 + 10 instruções

100 interações

1200 instruções no total

**Contagem estática = 326**



# Conceitos de performance e speedup



$$\textit{tempo de execução} = \textit{número de ciclos de clock} \cdot \textit{tempo de ciclo}$$

Quantos ciclos são requeridos num programa?

Reescrevendo o número de ciclos de clock em função da quantidade de instruções executadas pelo programa temos:

$$\textit{tempo de execução} = \textit{número de instruções} \cdot \frac{\textit{número de ciclos}}{\textit{instrução}} \cdot \textit{tempo de ciclo}$$

Poderíamos assumir que o número de ciclos = número de instruções???



# Conceitos de performance e speedup



$$\textit{tempo de execução} = \textit{número de ciclos de clock} \cdot \textit{tempo de ciclo}$$

Reescrevendo o número de ciclos de clock em função da quantidade de instruções executadas pelo programa temos:

$$\textit{tempo de execução} = \textit{número de instruções} \cdot \frac{\textit{número de ciclos}}{\textit{instrução}} \cdot \textit{tempo de ciclo}$$

$$\textit{tempo de execução} = \textit{número de instruções} \cdot CPI \cdot \textit{tempo de ciclo}$$

# Conceitos de performance e speedup



$$\text{tempo de execução} = \text{número de instruções} \cdot CPI \cdot \text{tempo de ciclo}$$

Exemplo: Considere um programa com 1000 instruções com a seguinte frequência de instruções

Tipo de Operação	Ciclos / instrução	Frequência de uso	Total de instruções	total de ciclos
ALU	4	50%	500	$= 500 * 4 = 2000$
Load	5	20%	200	$= 200 * 5 = 1000$
Store	3	10%	100	$= 100 * 3 = 300$
Branch	3	20%	200	$= 200 * 3 = 600$
				3900

$$CPI \text{ média} = \frac{3900}{1000} = 3.9$$



# Conceitos de performance e speedup



Como melhorar a performance?

$$\textit{tempo de execução} = \textit{número de instruções} \cdot \textit{CPI} \cdot \textit{tempo de ciclo}$$

$$\frac{\textit{segundos}}{\textit{programa}} = \frac{\textit{instruções}}{\textit{programa}} \cdot \frac{\textit{ciclos}}{\textit{instrução}} \cdot \frac{\textit{segundos}}{\textit{ciclo}}$$



# Conceitos de performance e speedup



Como melhorar a performance?

$$\textit{tempo de execução} = \textit{número de instruções} \cdot \textit{CPI} \cdot \textit{tempo de ciclo}$$

	# de instruções	CPI	tempo de ciclo
Algoritmo			
Linguagem de programação			
Compilador			
ISA			
Organização do processador			
Tecnologia			

# Conceitos de performance e speedup



Como melhorar a performance?

$$\text{tempo de execução} = \text{número de instruções} \cdot CPI \cdot \text{tempo de ciclo}$$

	# de instruções	CPI	tempo de ciclo
Algoritmo	X	X	
Linguagem de programação	X	X	
Compilador	X	X	
ISA	X	X	X
Organização do processador		X	X
Tecnologia			X



# Lei de Amdahl



- O quanto a performance pode ser melhorada é limitada pela frequência de uso do recurso

# Conceitos de performance e speedup



Exemplo: Considere um programa com 1000 instruções com a seguinte frequência de

Tipo de Operação	Ciclos / instrução	Frequência de uso	Total de instruções	total de ciclos
ALU	4	50%	500	$= 500 * 4 = 2000$
Load	5	20%	200	$= 200 * 5 = 1000$
Store	3	10%	100	$= 100 * 3 = 300$
Branch	3	20%	200	$= 200 * 3 = 600$
				3900

Melhorando o Load em 60%

Tipo de Operação	Ciclos / instrução	Frequência de uso	Total de instruções	total de ciclos
ALU	4	50%	500	$= 500 * 4 = 2000$
Load	5 → 2	20%	200	$= 200 * 2 = 400$
Store	3	10%	100	$= 100 * 3 = 300$
Branch	3	20%	200	$= 200 * 3 = 600$
				3300

- Como ele é utilizado em 20% dos casos o impacto da melhoria no total de ciclos é de 15,4%
- No caso extremo (impossível) em que o Load fosse executado em 0 ciclos a melhoria no total de ciclos seria de 20%

- O quanto a performance pode ser melhorada é limitada pela frequência de uso do recurso

$$\text{Speedup} = \frac{\textit{tempo de execução antigo}}{\textit{novo tempo de execução}}$$

$$\textit{Speedup} = \frac{1}{(1 - p) + p/N}$$

**Exemplo:** Um processador gasta 30% do tempo executando instruções de adição, 25% do tempo em instruções de multiplicação e 10% em instruções de divisão. Avalie o speedup em cada um dos casos:

- a) Reprojetar a adição tornando-a 2x mais rápida
- b) Reprojetar a multiplicação tornando-a 3x mais rápida
- c) Reprojetar a divisão tornando-a 10x mais rápida

**Solução:**

- d) Adição:  $\text{speedup} = 1 / [0.7 + 0.3 / 2] = 1.18$
- e) Multiplicação:  $\text{speedup} = 1 / [0.75 + 0.25 / 3] = 1.20$
- f) Divisão:  $\text{speedup} = 1 / [0.9 + 0.1 / 10] = 1.10$

