Tópico 01:
Revisão de desempenho de computadores

Adapted from "Computer Architecture: A Quantitative Approach", J. Hennessy and D. Patterson, Morgan Kaufmann Publishers. Copyright 1998 UCB.

Medida de desempenho (uma analogia)

Avião	DC a Paris	Velocidade	Passageiros	Throughput (p.km/h)
Boeing 747	6,5 horas	980 km/h	470	460.600
BAC/Sud Concorde	3 horas	2170 km/h	132	286.440

- **Tempo para realizar uma tarefa (Tempo Ex)**
  - Tempo de execução, tempo de resposta, latência
- **Tarefas por hora, segundos, ns … (Desempenho)**
  - Throughput, bandwidth

Medida de desempenho

- **"X é n vezes mais rápido que Y" significa:**

$$n = \frac{\text{TempoEx(Y)}}{\text{TempoEx(X)}} = \frac{\text{Desempenho(X)}}{\text{Desempenho(Y)}}$$

- **Programas para avaliar desempenho**
  - Programas reais (ex.compilador C, TeX, Spice)
  - Kernels (ex.Livermore loops)
  - Bench. sintéticos (ex.Whetstone, Dhrystone)
  - Conjuntos de benchmarks (ex. SPEC)
    - » Especificação de parâmetros p/ HW e SW
    - » Especificação dos flags do compilador
    - » Resultados: caso básico e caso otimizado

SPEC: "System Performance Evaluation Cooperative"

- **SPEC 1992** (Lista: Figura 1.9)
  - SPECint92 (6 programas) e SPECfp92 (14 programas)
- **SPEC1995** (http://www.spec.org/osg/cpu95)
  - SPECint95 (8 programas) e SPECfp95 (10 programas)
  - SPECint_base95, SPECfp_base95 (flags pré-definidos)
- **Comparação de processadores**
  - http://infopad.eecs.berkeley.edu/CIC/summary
- **<u>Tempo de execução é a medida de desempenho de computadores!</u>**
- **A noção de "workload"**
  - Vários programas e seus pesos

Como expressar o desempenho ?

- **A média aritmética (ponderada) captura tempo de execução** **(ex. figura 1.12)**

$$\sum_{i=1}^{n} P_i \times T_i$$

- **Tempo de execução normalizado é útil para expressar proporção**
  - **X vezes mais rápido que uma SPARCstation 10**
  - **Não use média aritmética de tempo de execução normalizado, use a média geométrica (ex. figura 1.13)**

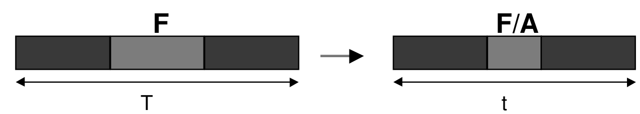
$$\sqrt[n]{\prod_{i=1}^{n} TN_i}\ \text{, onde } TN_i = \frac{T_{ref}}{T_i}$$

A Lei de Amdahl

- **Aceleração devida a uma melhoria M:**

$$A_{total}(M) = \frac{TempoEx_{semM}}{TempoEx_{comM}} = \frac{Desempenho_{comM}}{Desempenho_{semM}}$$

- **Melhoria M acelera uma fração F de uma tarefa por um fator A e o restante da tarefa não é afetado.**



$$t = (1-F)\times T + F \times \frac{T}{A} \Rightarrow \frac{T}{t} = \frac{1}{(1-F)+F/A}$$

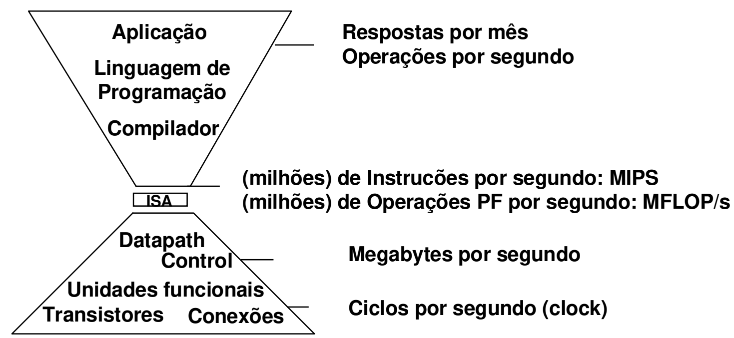
A Lei de Amdahl

$$A_{total} = \frac{T}{t} = \frac{1}{(1-F)+F/A}$$

- **Observação:**
  - Aceleração incremental diminui conforme mais melhorias são introduzidas
- **Corolário:**
  - Se uma melhoria é utilizável apenas por uma fração de uma tarefa, há um limite para A_{total}.
- **Exemplo:**
  - Instruções em PF melhoradas p/ rodar 2X mais rápido;
  - Mas apenas 10% das instruções são PF

$$A_{total} = \frac{T}{t} = \frac{1}{(1-0{,}1)+0{,}1/2} = \frac{1}{0{,}95} = 1{,}053$$

Métrica de desempenho



Desempenho da CPU

$$\text{tempo}_{\text{CPU}} = \frac{\text{segundos}}{\text{programa}} = \frac{\text{instruções}}{\text{programa}} \times \frac{\text{ciclos}}{\text{instrução}} \times \frac{\text{segundos}}{\text{ciclo}}$$

$$\boxed{\text{tempo}_{\text{CPU}} = \text{I} \times \text{CPI} \times \text{T}_{\text{r}}}$$

	I	CPI	T_r
Programa	X		
Compilador	X	(X)	
Conjunto de Instruções	X	X	
Organização		X	X
Tecnologia			X

Número médio de ciclos por instrução

$$\text{tempo}_{\text{CPU}} = \text{T}_{\text{r}} \times \sum_{i=1}^{n} \text{CPI}_i \times \text{I}_i$$

$$\text{CPI} = \frac{\text{número de ciclos}}{\text{número de instruções}} = \frac{\text{tempo}_{\text{CPU}} / \text{T}_{\text{r}}}{\text{I}}$$

$$\text{CPI} = \frac{\text{tempo}_{\text{CPU}}}{\text{I} \times \text{T}_{\text{r}}} = \frac{1}{\text{I}} \sum_{i=1}^{n} \text{CPI}_i \times \text{I}_i$$

Freqüência de ocorrência de instruções

$$CPI = \sum_{i=1}^{n} CPI_i \times F_i \text{ , onde } F_i = \frac{I_i}{I}$$

- **Princípio: Invista em recursos onde mais tempo é gasto!**
- **Exemplo: cálculo de CPI médio**

Máquina (Reg / Reg)

Op	Freq	Ciclos	CPI_i	(% tempo)
ALU	50%	1	0.5	(33%)
Load	20%	2	0.4	(27%)
Store	10%	2	0.2	(13%)
Branch	20%	2	0.4	(27%)
			1.5	

"mix" típico

CA, A1, slide 11
Luiz C. V. dos Santos INE/CTC/UFSC