Organização de Computadores

Aula 9

Avaliação de Desempenho

Avaliação de desempenho

1. Introdução

2. Métricas básicas

- **Ciclos de clock**
- **Tempo de CPU**
- **Ciclos por instrução**

3. Outras métricas e seus problemas

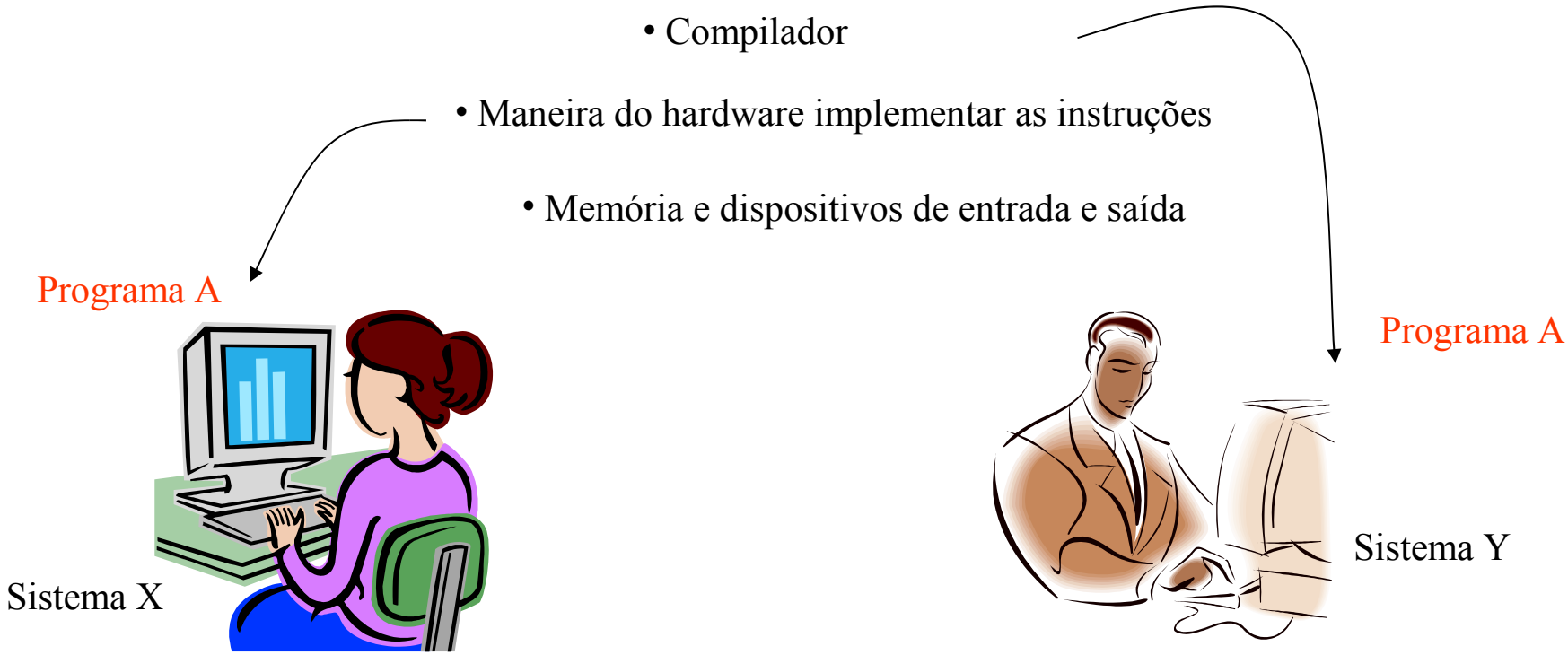
- **MIPS**
- **MFLOPS**

4. Benchmarks

1. Introdução Avaliação de Desempenho

- **Desempenho = eficácia do sistema composto por hardware e software.**

- **Desafio em calcular a performance (desempenho) em sistemas cada vez mais complexos com softwares e hardwares diferentes.**



Exemplo

Avião	Capacidade (passageiros)	Autonomia (milhas)	Velocidade (milhas/hora)	Thoughput Pass * milh/ hor
Boeing 777	375	4630	610	228.750
Boeing 747	470	4150	610	286.700
Concorde	132	4000	1350	178.200
Douglas	146	8720	544	79.424

Qual avião tem melhor desempenho se definirmos desempenho segundo sua velocidade?

- melhor velocidade de cruzeiro (levar 1 unico passageiro mais rapidamente)
- mais passageiros no menor tempo

Qual avião tem melhor desempenho se definirmos desempenho como a capacidade de levar mais passageiros?

Pensando em um usuário no computador…, ele está interessado mais é no tempo de resposta a um certo programa, ou seja, no tempo decorrido do inicio ao final do seu programa.

Introdução - Parâmetros

PARAMETROS DE MEDIDAS

- **Tempo de relógio real necessário para execução de um programa (ou tempo de resposta) inclui ...**
  - **acessos a disco**
  - **atividades de I/O**
  - **overhead do sistema operacional, da rede, ...**
- **No entanto, seguidamente um sistema executa diversos programas simultaneamente**
- **Throughput , vazão, é uma medida do número de tarefas executada por unidade de tempo**
- **O que otimizar?**
  - **Tempo de resposta de um programa isolado?**
  - **Throughput do sistema?**

Introdução - Tempo de Execução x Throughput

Para um dado sistema de computação:

- **Subtituição de um processador por outro mais rápido, diminui o tempo de resposta e quase sempre melhora o throughput.**

- **Alocação de processadores adicionais a um sistema que utiliza varios processadores para executar programas diferentes, não faz com que as tarefas sejam realizadas mais rápidas pois os processadores são os mesmos, porém aumenta o throughput caso a demanda de processamento anteriormente fosse maior que a possivel.**

Primeiramente iremos discutir tempo de resposta:

- **Desempenho =** $\dfrac{1}{\text{Tempo de execução}}$

Introdução

- **Distinção entre …**
  - **tempo de relógio real gasto em um programa, e**
  - **tempo do processador CPU efetivamente gasto com tarefas específicas do programa**
    - **não inclui tempo de execução de outros programas, tempo de espera por I/O, ...**

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

Introdução

- **Exemplo no UNIX com o comando *time***

90.7s 12.9s 2:39m 65% **(90.7 + 12.9) / 2:39 = 0.65**

tempo do processador

tempo do sistema

tempo decorrido

percentual do processado pelo decorrido

Tempo em segundos e minutos.

*** Note que a maioria do tempo é gasto em entrada e saida e/ou rodando outros programas.**

- **Tendo em vista os objetivos desta disciplina de Organização, desempenho de CPU será medido apenas em função do tempo de CPU do usuário, ou seja, o tempo do processador gasto para a execução das instruções do programa em questão.**

Introdução

- **Desempenho é conseqüência de otimizações feitas em 3 dimensões do hardware:**

- **Arquitetura**
  - **processadores RISC**
  - **instruções otimizadas**

- **Organização**
  - **paralelismo**
  - **memória cache**

- **Tecnologia**
  - **velocidade de chaveamento das portas lógicas**
  - **tempo de acesso à memória**

- **Desempenho também depende no entanto do compilador**
  - **geração de código otimizada**
  - **bom aproveitamento dos recursos da organização**

2. Métricas básicas – Período do Relógio

- **<u>Ciclos de relógio</u> = intervalos básicos de tempo nos quais são executadas as operações elementares de uma instrução**
  - **transferências de valores entre registradores**
  - **operações aritméticas na ALU**

- **<u>Período do relógio</u> T = tempo de duração de um ciclo de clock**
  - **p.ex. 1 ns**

- **<u>Freqüência de operação</u> f = número de ciclos de clock por unidade de tempo**

$$f = 1 / T$$

se T = 1 ns

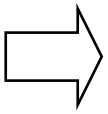
Então $f = 1 / 1.10^{-9} = 10^{9} = 1$ **GHz**

Métricas básicas – Tempo de CPU de um programa

tempo de CPU de um programa	=	nº de ciclos de clock do programa	X	período do clock

por exemplo:

– programa gasta 10.10^6 ciclos
– período do clock é 1 ns



tempo de CPU do programa =

$10.10^6 \times 1.10^{-9} = 10.10^{-3} = 10$ ms

alternativamente:

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

Formas de Aumento de Desempenho

Formas de aumento do desempenho:

- **– diminuir o período do clock**
  - **tecnologia, organização**
- **– diminuir nº de ciclos necessários para execução do programa**
  - **organização, arquitetura**

Objetivos muitas vezes conflitantes.

Métricas básicas – Exercício Frequência

Exercício

- programa roda em 10 s na máquina A, que tem clock de 1 GHz
- queremos rodá-lo em 6 s numa máquina B com nova tecnologia e nova organização
- a máquina B pode ter um aumento substancial de freqüência de clock
- no entanto, a organização da máquina B exigirá 1.2 vezes mais ciclos de clock para executar instruções do que a máquina A
- qual é a freqüência de clock necessária para a máquina B ?

Métricas básicas – Tempode CPU

Solução do exercício

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

$$10\ s = \frac{\text{nº de ciclos de clock em A}}{10^9 \text{ ciclos / s}}$$

$$\text{nº de ciclos de clock em A} = 10 \cdot 10^9 \text{ ciclos}$$

$$6\ s = \frac{1.2 \times \text{nº de ciclos de clock em A}}{\text{freqüência do clock em B}} = \frac{1.2 \times 10 \cdot 10^9}{\text{freqüência do clock em B}}$$

freqüência do clock em B = 2 GHz

Métricas básicas: CPI

nº de ciclos de clock do programa	=	nº instruções do programa	X	nº médio de ciclos de clock por instrução

CPI = nº médio de ciclos de clock por instrução
é uma média sobre todas as instruções executadas no programa.

Exercício:

- **Duas máquinas implementam o mesmo conjunto de instruções**
- **Um mesmo programa é rodado em ambas as máquinas**
- **Máquina A tem período de clock de 1 ns e CPI = 2.0 para este programa**
- **Máquina B tem período de clock de 2 ns e CPI = 1.2 para o mesmo programa**

- **Qual máquina é mais rápida ?**

Métricas básicas CPI

Solução do exercício

nº de ciclos de clock do programa	=	N	(nº instruções do programa)	X	CPI

nº de ciclos de clock de A = 2.0 x N

nº de ciclos de clock de B = 1.2 x N

tempo de CPU de um programa	=	nº de ciclos de clock do programa	X	período do clock

tempo de CPU de A = 2.0 x N x 1 ns = 2 N ns

tempo de CPU de B = 1.2 x N x 2 ns = 2.4 N ns

portanto: máquina A é 1.2 vezes mais rápida do que B na execução deste programa

Métricas básicas: Tempo de CPU

tempo de CPU = nº de instruções X CPI X período do clock

$$\text{tempo de CPU} = \frac{\text{nº de instruções} \times \text{CPI}}{\text{freqüência do clock}}$$

- **Como obter estes valores?**
  - **Tempo de CPU é medido executando-se o programa**
  - **Período (ou freqüência) do clock é divulgado pelo fabricante**
  - **nº de instruções e CPI são mais difíceis de obter**
    - **nº de instruções depende da arquitetura e do compilador**
    - **CPI depende da organização e do programa**
  - **Conhecendo-se nº de instruções ou CPI, pode-se determinar o outro a partir da fórmula**

Métricas básicas

Nº de ciclos de clock de um programa pode ser determinado ...

– Pela análise dos diferentes tipos de instruções executados pelo programa

– Pelo nº de ciclos de clock necessários para cada tipo de instrução

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

onde
n = nº de tipos de instruções
CPI_i = ciclos de clock por instrução do tipo i
C_i = nº de instruções do tipo i no programa

Métricas básicas

Exercício:

– O projetista de um compilador deseja decidir entre duas possíveis seqüências de código para a resolução de um problema

– Dados os tipos de instruções e o nº de ciclos por instrução de cada tipo, qual seqüência é mais rápida?

tipo de instrução	CPI
A	1
B	2
C	3

código	nº de instruções (x N)		
	tipo A	tipo B	tipo C
1	2	1	2
2	4	1	1

Métricas básicas

Solução do exercício

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 x 1) + (1 x 2) + (2 x 3) = 10 ciclos

Nº ciclos de clock para código 2 = (4 x 1) + (1 x 2) + (1 x 3) = 9 ciclos

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

CPI código 1 = 10 / 5 = 2.0

CPI código 2 = 9 / 6 = 1.5

Código 2 é mais rápido, mesmo que execute uma instrução a mais, pois tem CPI bem mais baixo.

3. Outras métricas: MIPS

$$MIPS = \frac{\text{nº instruções}}{\text{tempo de CPU X } 10^6} = \frac{\text{nº instruções}}{\text{ciclos X período X } 10^6}$$

$$= \frac{\text{nº instruções X freqüência}}{\text{nº instruções X CPI X } 10^6}$$

$$MIPS = \frac{\text{freqüência}}{\text{CPI X } 10^6}$$

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**

MIPS

Exemplo:

tipo de instrução	CPI
A	1
B	2
C	3

código	nº de instruções (x N)		
	tipo A	tipo B	tipo C
compil. 1	5	1	1
compil. 2	10	1	1

freqüência do clock = 1 GHz

Calcular o desempenho do código gerado por cada compilador:

- **em MIPS**
- **em tempo de CPU**

MIPS

$$MIPS = \frac{\text{freqüência}}{CPI \times 10^6}$$

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

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

$$CPI = \frac{\sum (CPI_i \times C_i)}{\text{nº instruções}}$$

$$CPI_1 = \frac{(5x1 + 1x2 + 1x3) . 10^6}{(5 + 1 + 1) \times 10^6} = \frac{10}{7} = 1.43$$

$$CPI_2 = \frac{(10x1 + 1x2 + 1x3) . 10^6}{(10 + 1 + 1) \times 10^6} = \frac{15}{12} = 1.25$$

$$MIPS_1 = \frac{1\ GHz}{1.43 \times 10^6} = 699 \qquad MIPS_2 = \frac{1\ GHz}{1.25 \times 10^6} = 800$$

MIPS

$$\text{tempo de CPU} = \frac{\text{nº instruções X CPI}}{\text{freqüência}}$$

$$\text{tempo}_1 = \frac{(5 + 1 + 1) \cdot 10^6 \times 1.43}{1 \cdot 10^9} = 10 \text{ ms}$$

$$\text{tempo}_2 = \frac{(10 + 1 + 1) \cdot 10^6 \times 1.25}{1 \cdot 10^9} = 15 \text{ ms}$$

O código 2 gasta portanto mais tempo.

Apesar de ter MIPS com maior valor, o código 2 gasta bem mais instruções.

Outras métricas: 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, raíz 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)**
- **como na medida de MIPS, pode-se ter gasto maior em tempo mesmo tendo mais MFLOPS**

4. Benchmarks

- **Benchmarks são conjuntos de programas representativos da carga de trabalho de uma máquina**
  - **Utilizados para avaliação de desempenho, segundo métricas já discutidas**
  - **Geralmente de domínio público**
  - **Procuram explorar repertório de recursos da arquitetura**

- **Problemas com benchmarks**
  - **Melhor avaliação seria feita com a carga de trabalho efetivamente utilizada em cada máquina (workload)**
  - **Escolha dos benchmarks e aplicação rigorosa da metodologia de avaliação**
  - **Fabricantes podem tentar otimizar arquitetura e/ou organização e/ou compilador para executar de forma mais eficiente apenas os benchmarks**

Tipos de benchmarks

- **Aplicações reais ou modificadas**
  - **Quais ?**
  - **Dados de entrada ?**
  - **Exemplos:**
    - **programas do SPEC (empresa de padronização de avaliação)**
    - **iCOMP (benchmarks da Intel, aplicações sob Windows, PC-workload)**
    - **transações de banco de dados (TPC-A, TPC-B, etc.)**
    - **simuladores de equações diferenciais (NAS: benchmarks NASA)**
- **Kernels**
  - **Linpack: rotinas de resolução de sistema de equações lineares**
  - **Livermore Loops, NAS kernels (simulação numérica) …**
- **Benchmarks de "brinquedo"**
  - **Puzzle, Quicksort, Sieve, …**
- **Benchmarks sintéticos**
  - **Dhrystone, Whetstone, ...**

Exemplos de benchmarks

- **Dhrystone**
  - desenvolvido em 1984 e escrito em ADA, C ou Pascal
- **Whetstone**
  - primeiro grande programa de benchmark sintético
  - exemplo de uma tabela de resultados:

www.dl.ac.uk/TCSC/disco/Benchmarks/whetstone.html

- **SPEC**
  - *Standard Performance Evaluation Corporation*

www.spec.org/benchmarks.html

  - **SPEC2000** www.specbench.org/cpu2000
    - CINT2000: 12 programas para cálculos intensivos com inteiros
    - CFP2000: 14 programas para cálculos intensivos de ponto flutuante
    - máquina de referência: SUN Ultra 5_10 com processador SPARC de 300MHz e 256MB de memória
  - **SPEC2004** www.specbench.org/cpu2004

What is SPEC?

SPEC is the Systems Performance Evaluation Cooperative.

www.spec.org

- **The current membership list includes 16 companies: AT&T, Control Data Corp., Data General, Digital Equipment Corp., DuPont, Hewlett-Packard, IBM, Intel, Intergraph, MIPS Computer Systems, Motorola, Multiflow, Solbourne, Stardent, Sun and Unisys.**

- **The SPEC applications represent a large body of code (over 14 megabytes) which span a range of application arenas.**
- **The membership to SPEC is open to any interested company.**
- **SPEC is not devoted to any single architecture nor any particular philosophy of computing systems.**
- **SPEC has created a framework in which a wide variety of applications can be tested by a very large audience of computer users.**

SPEC95 Programs

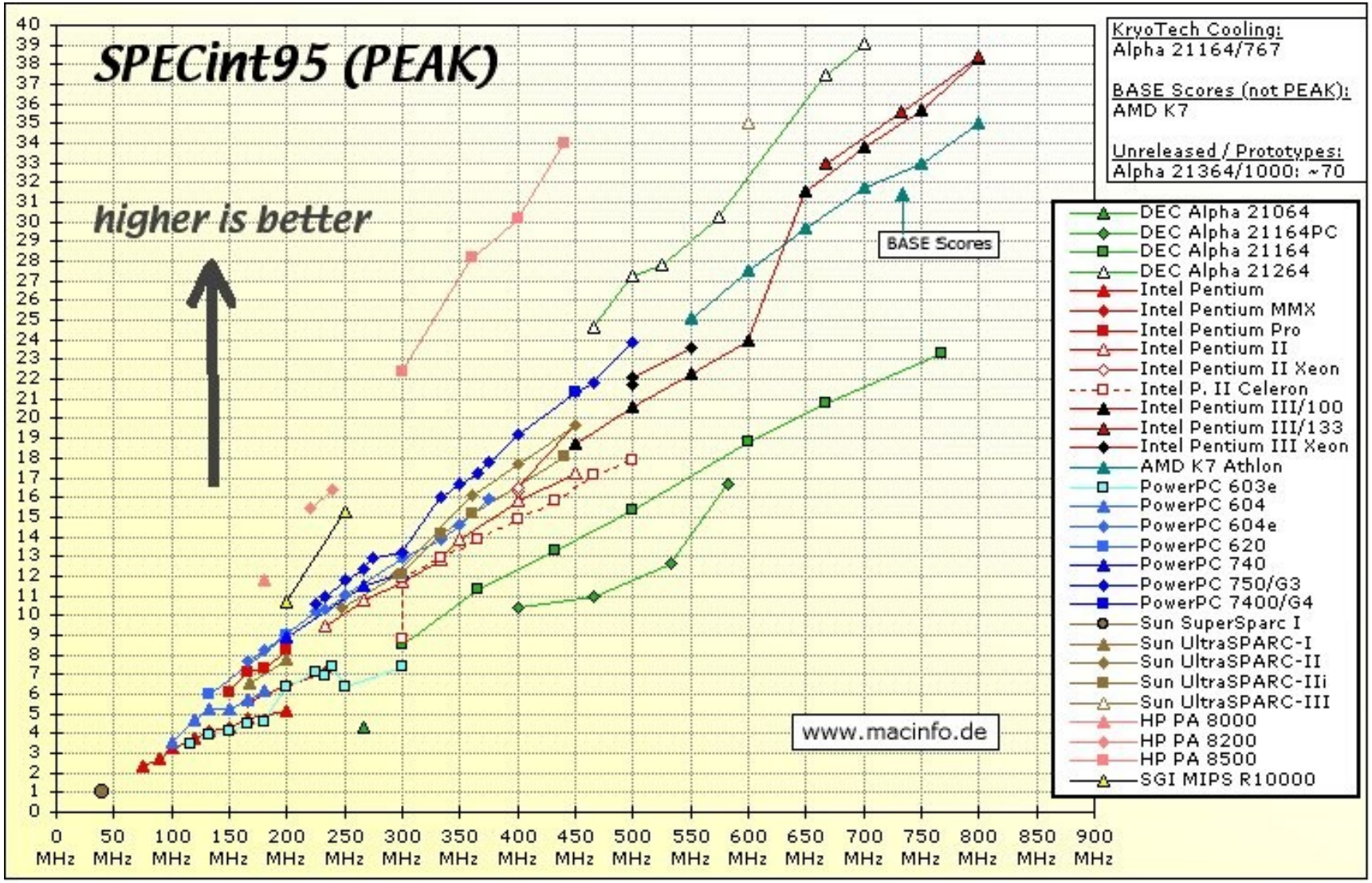
	Benchmark	Description
Integer	go	Artificial intelligence; plays the game of Go
	m88ksim	Motorola 88k chip simulator; runs test program
	gcc	The Gnu C compiler generating SPARC code
	compress	Compresses and decompresses file in memory
	li	Lisp interpreter
	ijpeg	Graphic compression and decompression
	perl	Manipulates strings and prime numbers in the special-purpose programming language Perl
	vortex	A database program
Floating Point	tomcatv	A mesh generation program
	swim	Shallow water model with 513 x 513 grid
	su2cor	Quantum physics; Monte Carlo simulation
	hydro2d	Astrophysics; Hydrodynamic Naiver Stokes equations
	mgrid	Multigrid solver in 3-D potential field
	applu	Parabolic/elliptic partial differential equations
	trub3d	Simulates isotropic, homogeneous turbulence in a cube
	apsi	Solves problems regarding temperature, wind velocity, and distribution of pollutant
	fpppp	Quantum chemistry
	wave5	Plasma physics; electromagnetic particle simulation

Medidas de Tempo de Execução no SPEC

- **As medidas de tempo de execução são normalizadas por meio da divisão pelo tempo gasto para executar o benchmark na estação de trabalho SUN SPARC station 10/40, pelo tempo de execução onde a medição esta sendo realizada. = razão SPEC.**

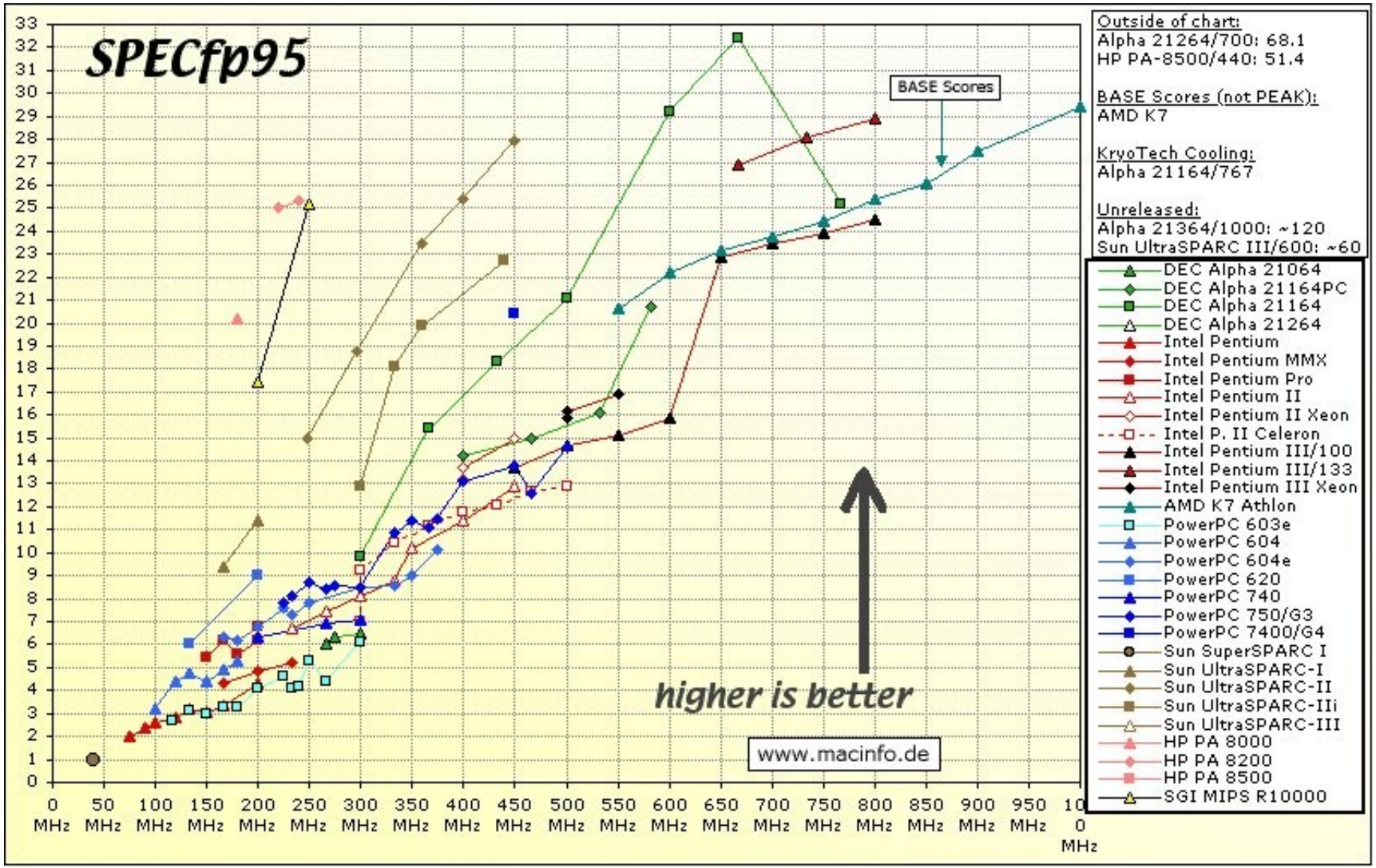
- **A medida final de performance é medida pela média geométrica das razões SPEC para cada programa do conjunto, por exemplo SPECint95 e SPECfp95.**

Sample SPECint95 Results



Source URL: http://www.macinfo.de/bench/specmark.html

Sample SPECfp95 Results



Source URL: http://www.macinfo.de/bench/specmark.html

SPEC CPU2000

- **CINT 2000**

Benchmark	Função	Fonte
gzip	compressão de arquivos	C
vpr	posicionamento e roteamento de circuitos FPGA	C
gcc	compilador C	C
mcf	otimização combinacional	C
crafty	jogo: xadrez	C
parser	parser	C
eon	computação gráfica	C++
perlmbk	linguagem de programa PERL	C
gap	linguagem e biblioteca para computação em grupo	C
vortex	banco de dados orientado a objetos	C
bzip2	compressão de arquivos	C
twolf	simulador de posição e rota	C

SPEC CPU2000

- **CFP 2000**

Benchmark	Função	Fonte
wupwise	computação da propagação de quarks	Fortran77
swim	modelo da superfície d'água com 513x513 pontos	Fortran77
mgrid	soluções multigrid em campos 3D	Fortran77
apply	equações parciais diferenciais	Fortran77
mesa	biblioteca gráfica 3-D	C
galgel	dinâmica de fluídos	Fortran90
art	reconhecimento de imagens / redes neurais	C
equake	simulação de propagação de ondas sísmicas	C
facerec	processamento de imagens: reconhecimento de faces	Fortran90
ammp	modelo de um grande sistema de moléculas	C
lucas	teoria de números / teste de números primos	Fortran90
fma3d	simulação de ruídos elétricos em elementos finitos	Fortran90
sixtrack	energia nuclear: acelerador de partículas	Fortran77
apsi	problemas com temperatura, vento e poluição	Fortran77

Top 20 SPEC CPU2000 Results (As of March 2002)

Top 20 SPECint2000

#	MHz	Processor	int peak	int base
1	1300	POWER4	814	790
2	2200	Pentium 4	811	790
3	2200	Pentium 4 Xeon	810	788
4	1667	Athlon XP	724	697
5	1000	Alpha 21264C	679	621
6	1400	Pentium III	664	648
7	1050	UltraSPARC-III Cu	610	537
8	1533	Athlon MP	609	587
9	750	PA-RISC 8700	604	568
10	833	Alpha 21264B	571	497
11	1400	Athlon	554	495
12	833	Alpha 21264A	533	511
13	600	MIPS R14000 500	483	675
14	675	SPARC64 GP 478	449	900
15	900	UltraSPARC-III	467	438
16	552	PA-RISC 8600	441	417
17	750	POWER RS64-IV	439	409
18	700	Pentium III Xeon	438	431
19	800	Itanium	365	358
20	400	MIPS R12000 353	328	400

Top 20 SPECfp2000

MHz	Processor	fp peak	fp base
1300	POWER4	1169	1098
1000	Alpha 21264C	960	776
1050	UltraSPARC-III Cu	827	701
2200	Pentium 4 Xeon	802	779
2200	Pentium 4	801	779
833	Alpha 21264B	784	643
800	Itanium	701	701
833	Alpha 21264A	644	571
1667	Athlon XP	642	596
750	PA-RISC 8700	581	526
1533	Athlon MP	547	504
600	MIPS R14000	529	499
	SPARC64 GP	509	371
	UltraSPARC-III	482	427
1400	Athlon	458	426
1400	Pentium III	456	437
500	PA-RISC 8600	440	397
450	POWER3-II	433	426
500	Alpha 21264	422	383
	MIPS R12000	407	382

Source: http://www.aceshardware.com/SPECmine/top.jsp

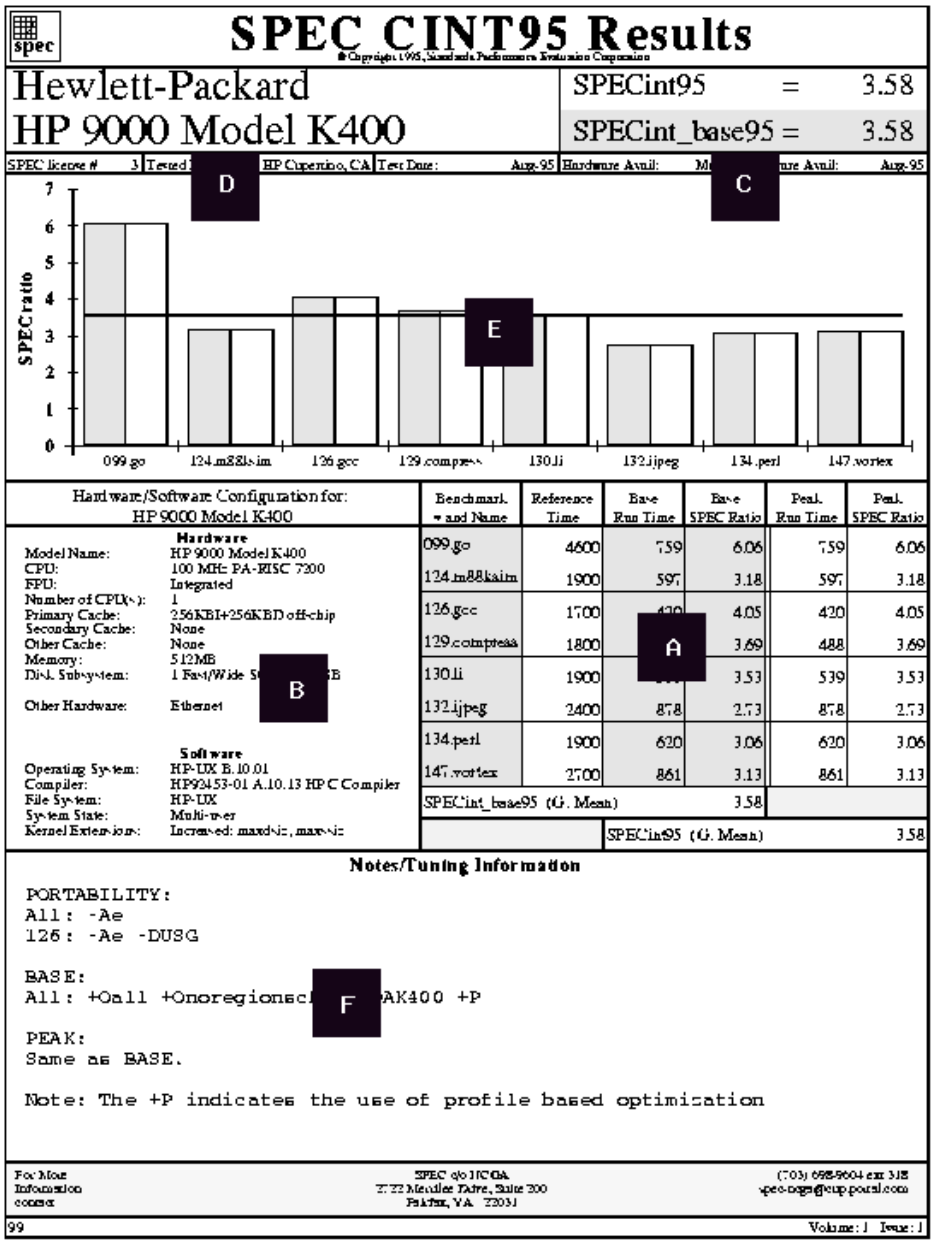
Resultados do benchmark Whetstone-97

www.dl.ac.uk/TCSC/disco/Benchmarks/whetstone.html

Rank	Máquina	TotalCPU (s)	MWIPS
1	Pentium 4/2666 (ifc)	10.4	3532
2	HP RX5670 Madison/1500 (+)	10.6	3532
3	IBM pSeries 690Turbo/1.7	10.8	3472
4	Compaq Alpha ES45/1250	10.9	3441
5	IBM Regatta-HPC/1300	11.5	3281
6	IBM pSeries 690Turbo/1.3	11.7	3260
7	Compaq Marvel EV7/1000	13.0	2894
8	Compaq Alpha ES45/1000	13.6	2778
9	IBM pSeries 630/1000	15.4	2461
10	HP RX2600 Itanium2/1000	15.9	2369
11	AMD Opteron244/1800 (pgi)	15.9	2358
12	SGI Onyx 300/R14k-600	16.4	2280
…	…	…	…

interações do Benchmark / s

Exemplo de relatório de resultados do SPEC



SPEC CINT95 Results

Hewlett-Packard
HP 9000 Model K400

SPECint95 = 3.58
SPECint_base95 = 3.58

SPEC license # 3 | Tested: | HP Cupertino, CA | Test Date: Aug-95 | Hardware Avail: | Software Avail: Aug-95

D

C

E

SPECratio

099.go 124.m88ksim 126.gcc 129.compress 130.li 132.ijpeg 134.perl 147.vortex

Hardware/Software Configuration for:
HP 9000 Model K400

Hardware

Model Name: HP 9000 Model K400
CPU: 100 MHz PA-RISC 7200
FPU: Integrated
Number of CPU(s): 1
Primary Cache: 256KBI+256KBD off-chip
Secondary Cache: None
Other Cache: None
Memory: 512MB
Disk Subsystem: 1 Fast/Wide S[illegible]B
Other Hardware: Ethernet

B

Software

Operating System: HP-UX B.10.01
Compiler: HP92453-01 A.10.13 HP C Compiler
File System: HP-UX
System State: Multi-user
Kernel Extensions: Increased: maxdsiz, maxssiz

Benchmark Name	Reference Time	Base Run Time	Base SPEC Ratio	Peak Run Time	Peak SPEC Ratio
099.go	4600	759	6.06	759	6.06
124.m88ksim	1900	597	3.18	597	3.18
126.gcc	1700	420	4.05	420	4.05
129.compress	1800	488	3.69	488	3.69
130.li	1900	[illegible]	3.53	539	3.53
132.ijpeg	2400	878	2.73	878	2.73
134.perl	1900	620	3.06	620	3.06
147.vortex	2700	861	3.13	861	3.13
SPECint_base95 (G. Mean)			3.58		
		SPECint95 (G. Mean)			3.58

A

Notes/Tuning Information

```
PORTABILITY:
All: -Ae
126: -Ae -DUSG

BASE:
All: +Oall +Onoregionsc[illegible]AK400 +P

PEAK:
Same as BASE.

Note: The +P indicates the use of profile based optimization
```

F

For More Information contact SPEC c/o NCGA 2722 Merrilee Drive, Suite 200 Fairfax, VA 22031 (703) 698-9604 ext 318 spec-ncga@cup.portal.com

99 Volume: 1 Issue: 1

A: seção de resultados

B: configuração do sistema

C: disponibilidade

D: patrocinador (*sponsor*)

E: gráfico

F: seção de notas

Resumo:

- **Qual a métrica mais importante? MIPS? MFLOPS?**
- **Como escolher uma CPU?**
  - **Spec int?**
  - **Spec FP?**
  - **Média dos dois?**
  - **Whetstone?**
- **Que outros fatores impactam a escolha de uma CPU?**

END

Exercício 1

Para um dado programa:

Classe	CPI em M1	CPI em M2	Codigo por C1	Codigo por C2	Codigo por terceiros
A	4	2	30%	30%	50%
B	6	4	50%	20%	30%
C	8	3	20%	50%	20%

A máquina M1 tem um clock de 400MHz e a M2 tem um clock de 200MHz.

Usando o compilador C1 na maquina M1 e o compilador C2 na máquina M2, qual é a mais rapida e de quanto comparado com a outra?

Se tiveres que comprar a máquina M2, qual compilador deverias usar para conseguir uma melhor performance para o programa em questão.

Exercício 2

- **Considere duas implementações diferentes M1 e M2 para o mesmo conjunto de instruções. Existem 4 classes de instruções.**

Classe	CPI para M1	CPI para M2
A	1	2
B	2	2
C	3	4
D	4	4

- **M1 tem um clock de 500MHz e M2 tem um clock de 750MHz.**
- **Suponha que o desempenho de pico é definida como a taxa mais elevada que uma máquina possa executar uma seqüência de instruções, escolhida para maximizar essa taxa. Determine o desempenho de pico para M1 e M2 dada em instruções executadas por segundo.**
- **Se o numero de instruções de um programa for dividido igualmente entre as classes, qual máquina executa o programa mais rapidamente?**

Exercício 3

- **A tabela a seguir mostra o numero de operações em ponto flutuante executadas por dois programas diferentes além do tempo de execução em 3 máquinas distintas.**

Programa	Operações em ponto flutuante	Tempo de execução (s)		
		Computador A	Computador B	Computador C
Programa 1	10.000.000	1	10	20
Programa 2	100.000.000	1.000	100	20

Qual das máquinas é mais rápida considerando o tempo de execução. E quanto mais rápida ela é comparada as outras duas?