

Desempenho: benchmarks

Estudo de caso 1: SPEC CPU

- **Tempos de execução normalizados**
 - Relativos a uma máquina de referência

$$\text{Razão SPEC} = \frac{\text{TempoEx}_{\text{SUN UltraSparc@296MHz}}}{\text{TempoEx}_{\text{medido}}}$$

- **Para os 12 programas inteiros**
 - **CINT2006** = média **geométrica** das razões SPEC
- **Para os 17 programas de ponto flutuante**
 - **CFP2006** = média **geométrica** das razões SPEC

Exemplo: CINT2006 para Opteron X4, modelo 2356 (Barcelona)

Desc ript ion	Na me	Inst ruct ion Coun t $\times 10^9$	CPI	Cloc k cy cle t ime (seco nds $\times 10^9$)	Execut ion Time (seco nds)	Reference Time (seco nds)	SPE Crat io
Interpreted string proces sing	perl	2,11 8	0.75	0.4	637	9,770	15.3
Block-sor ting compression	bzip2	2,389	0.85	0.4	817	9,650	11.8
GNU C compile r	gcc	1,050	1.72	0.4	724	8,050	11.1
Combinatorial optimization	mcf	336	10.00	0.4	1,345	9,120	6.8
Go game (AI)	go	1,658	1.09	0.4	721	10,490	14.6
Search gene sequ ence	hmmer	2,783	0.80	0.4	890	9,330	10.5
Chess game (AI)	sjeng	2,176	0.96	0.4	837	12,100	14.5
Quantum computer simulati on	libquantum	1,623	1.61	0.4	1,047	20,720	19.8
Video compressio n	h264a vc	3,102	0.80	0.4	993	22,130	22.3
Discre te event simulati on library	omnetpp	587	2.94	0.4	690	6,250	9.1
Games/path finding	astar	1,082	1.79	0.4	773	7,020	9.1
XML pars ing	xalancbmk	1,058	2.70	0.4	1,1 43	6,900	6.0
Geometric Mean							11.7

$$f = 2.5 \text{ GHz} \Rightarrow T = 0.4 \cdot 10^{-9} = 0.4 \text{ ns}$$

$$I \times \text{CPI} \times T = t_{\text{ex}} \quad t_{\text{referência}}/t_{\text{ex}} = \text{SPEC ratio}$$

$$\text{CINT2006} = (r_1 \times r_2 \times r_3 \times \dots \times r_{11} \times r_{12})^{1/12}$$

Exemplo: CINT2006 para Opteron X4, modelo 2356 (Barcelona)

Desc ript ion	Na me	Inst ruct ion Coun t $\times 10^9$	CPI	Cloc k cy cle t ime (seco nds $\times 10^9$)	Execut ion Time (seco nds)	Reference Time (seco nds)	SPE Crat io
Interpreted string proces sing	perl	2,11 8	0.75	0.4	637	9,770	15.3
Block-sor ting compression	bzip2	2,389	0.85	0.4	817	9,650	11.8
GNU C compile r	gcc	1,050	1.72	0.4	724	8,050	11.1
Combinatorial optimization	mcf	336	10.00	0.4	1,345	9,120	6.8
Go game (AI)	go	1,658	1.09	0.4	721	10,490	14.6
Search gene sequ ence	hmmer	2,783	0.80	0.4	890	9,330	10.5
Chess game (AI)	sjeng	2,176	0.96	0.4	837	12,100	14.5
Quantum computer simulati on	libquantum	1,623	1.61	0.4	1,047	20,720	19.8
Video compressio n	h264a vc	3,102	0.80	0.4	993	22,130	22.3
Discre te event simulati on library	omnetpp	587	2.94	0.4	690	6,250	9.1
Games/path finding	astar	1,082	1.79	0.4	773	7,020	9.1
XML pa rsing	xalancbmk	1,058	2.70	0.4	1,1 43	6,900	6.0
Geometric Mean							11.7

Pior CPI

Melhor CPI

Exemplo: CINT2006 para Opteron X4, modelo 2356 (Barcelona)

Desc ription	Name	Inst ruction Count $\times 10^9$	CPI	Cloc k cy cle t ime (seco nds $\times 10^9$)	Execut ion Time (seco nds)	Reference Time (seco nds)	SPE Crat io
Interpreted string proces sing	perl	2,11 8	0.75	0.4	637	9,770	15.3
Block-sor ting compression	bzip2	2,389	0.85	0.4	817	9,650	11.8
GNU C compile r	gcc	1,050	1.72	0.4	724	8,050	11.1
Combinatorial optimization	mcf	336	10.00	0.4	1,345	9,120	6.8
Go game (AI)	go	1,658	1.09	0.4	721	10,490	14.6
Search gene sequ ence	hmmer	2,783	0.80	0.4	890	9,330	10.5
Chess game (AI)	sjeng	2,176	0.96	0.4	837	12,100	14.5
Quantum computer simulati on	libquantum	1,623	1.61	0.4	1,047	20,720	19.8
Video compressio n	h264a vc	3,102	0.80	0.4	993	22,130	22.3
Discre te event simulati on library	omnetpp	587	2.94	0.4	690	6,250	9.1
Games/path finding	astar	1,082	1.79	0.4	773	7,020	9.1
XML pars ing	xalancbmk	1,058	2.70	0.4	1,1 43	6,900	6.0
Geometric Mean							11.7

Pior/melhor = 13 vezes

Discussão

- **Dada ISA, como melhorar o desempenho ?**
 - **Aumento da frequência do relógio**
 - **Melhorias na organização que diminuam CPI**
 - **Melhorias no compilador**
 - » **Que reduzam número de instruções ou**
 - » **Que gerem instruções com menor CPI médio**

Discussão

- É um erro comum assumir que:
“A introdução de uma melhoria afetando apenas parte de uma máquina aumente o desempenho total proporcionalmente à melhoria.”

Discussão

- **Exemplo:**
 - Um programa executa em 100ns.
 - As multiplicações contribuem com 80ns.
 - De quanto preciso acelerar a multiplicação para que o programa execute 5× mais rápido ?

$$\text{TempoEx}_{c/\text{melhoria}} = \frac{\text{TempoEx}_{\text{afetado p/melhoria}}}{\text{proporção de melhoria}} + \text{TempoEx}_{\text{não afetado p/melhoria}}$$

$$\frac{100}{5} = \frac{80}{n} + (100 - 80) \Leftrightarrow 20 = \frac{80}{n} + 20 \Leftrightarrow 0 = \frac{80}{n} \quad !?$$

Discussão

- É um erro comum:
“Utilizar apenas parte da equação de desempenho como métrica.”
- Exemplo: MIPS para medir desempenho.
 - Milhões de instruções por segundo
$$\text{MIPS} = \frac{I}{\text{TempoEx} \times 10^6}$$
 - Noção intuitiva, mas:
 - » Não leva em conta diferenças no conj. de instruções
 - » Varia entre programas num mesmo computador
 - » Pode variar inversamente com o desempenho!?

Discussão

- Se $f_r = 4 \text{ GHz}$, qual código é mais rápido ?

classe	CPI
A	1
B	2
C	3

(em bilhões de instruções)

	A	B	C
Compilador 1	5	1	1
Compilador 2	10	1	1

$$\text{TempoEx} = \frac{\text{ciclo}_{\text{SCPU}}}{f} = \frac{\sum_{i=1}^n (\text{CPI}_i \times C_i)}{f}$$

$$\text{MIPS} = \frac{I}{\text{TempoEx} \times 10^6}$$

$$\text{TempoEx}_1 = \frac{(5 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9}{4 \times 10^9} = 2,5 \text{ s}$$

$$\text{MIPS}_1 = \frac{(5 + 1 + 1) \times 10^9}{2,5 \times 10^6} = 2800$$

$$\text{TempoEx}_2 = \frac{(10 \times 1 + 1 \times 2 + 1 \times 3) \times 10^9}{4 \times 10^9} = 3,75 \text{ s}$$

$$\text{MIPS}_2 = \frac{(10 + 1 + 1) \times 10^9}{3,75 \times 10^6} = 3200$$

Cálculo do MIPS (alternativo)

$$\frac{\text{instruções}}{\text{segundo}} = \frac{\text{instruções}}{\text{ciclo}} \times \frac{\text{ciclos}}{\text{segundo}} = \frac{1}{\text{CPI}} \times f = \frac{f}{\text{CPI}}$$

$$\text{MIPS} = \frac{\text{instruções} / \text{s}}{10^6} = \frac{f}{10^6 \times \text{CPI}}$$

Classe	CPI	Fração
A	5	0,33
B	2	0,50
C	4	0,10
D	4	0,07

f = 200 MHz

$$\text{CPI} = 5 \times 0,33 + 2 \times 0,50 + 4 \times 0,10 + 4 \times 0,07 = 3,33$$

$$\text{MIPS} = \frac{200 \times 10^6}{10^6 \times 3,33} = 60$$

Consumo de energia e potência

- **Computação móvel**
 - Celulares e outros dispositivos móveis
 - » Maximizar tempo de bateria \Rightarrow
 - » Minimizar consumo de **energia**
- **Computação de uso geral**
 - Laptops
 - » Maximizar tempo de bateria \Rightarrow
 - » Minimizar consumo de energia
 - Laptops, desktops, servers
 - » Viabilizar resfriamento
 - » \Rightarrow Minimizar consumo de **potência**
 - Servers
 - » Maximizar a **eficiência energética** (operações/J)

Estudo de caso 2: SPEC Power

- **Começou com**
 - *SPEC benchmark for Java business applications*
- **Consumo de potência de servidores sob diferentes níveis de carga**
 - Desempenho: operações/s (ssj_ops)
 - Potência: Watts (Joules/s)
- **Métrica de eficiência energética**
 - Quantas operações por Joule?

Exemplo: SPECpower_ssj2008 Opteron X4, modelo 2356

Target Load %	Performance (ssj_ops)	Average Power (Watts)
100 %	231,86 7	295
90%	211 ,282	286
80%	185,80 3	275
70%	163,42 7	265
60%	140,16 0	256
50%	118,32 4	246
40%	92,03 5	233
30%	70,50 0	222
20%	47,12 6	206
10%	23,06 6	180
0%	0	141
Overall Sum	1,283, 590	2,605
$\sum \text{ssj_ops} / \sum \text{power} =$		493

$$\text{Overall ssj_ops per Watt} = \left(\sum_{i=0}^{10} \text{ssj_ops}_i \right) / \left(\sum_{i=0}^{10} \text{power}_i \right)$$

Discussão

- É um erro comum assumir que:
“Computadores com baixa utilização consomem pouca potência”

Ser ver Man ufact urer	Micro- proc esso r	Total Cor es/ Sock ets	Cloc k Rat e	Pe ak Pe rfor man ce (ssj_op s)	100 % Loa d Po wer	50% Loa d Po wer	50% Loa d/ 100 % Po wer	10% Loa d Po wer	10% Loa d/ 100 % Po wer	Act ive Idle/ Po wer	Act ive Idle/ 100 % Po wer
HP	Xeon E5440	8/2	3.0 GHz	308, 022	269 W	227 W	84%	174 W	65%	160 W	59%
Dell	Xeon E5440	8/2	2.8 GHz	305, 413	276 W	230 W	83%	173 W	63%	157 W	57%
Fujitsu Sei mens	Xeon X3220	4/1	2.4 GHz	143, 742	132 W	110 W	83%	85 W	65%	80 W	60%

A utilização de CPU dos Servidores da Google:

- 100% cerca de 1% do tempo
- Entre 10% e 50% a maior parte do tempo

Discussão

- É um erro comum assumir que:
“Computadores com baixa utilização consomem pouca potência”

Ser ver Man ufact urer	Micro- proc esso r	Total Cor es/ Sock ets	Cloc k Rat e	Pe ak Pe rfor man ce (ssj_op s)	100 % Loa d Po wer	50% Loa d Po wer	50% Loa d/ 100 % Po wer	10% Loa d Po wer	10% Loa d/ 100 % Po wer	Act ive Idle/ Po wer	Act ive Idle/ 100 % Po wer
HP	Xeon E5440	8/2	3.0 GHz	308, 022	269 W	227 W	84%	174 W	65%	160 W	59%
Dell	Xeon E5440	8/2	2.8 GHz	305, 413	276 W	230 W	83%	173 W	63%	157 W	57%
Fujitsu Sei mens	Xeon X3220	4/1	2.4 GHz	143, 742	132 W	110 W	83%	85 W	65%	80 W	60%

A utilização de CPU dos Servidores da Google:

- 100% cerca de 1% do tempo
- Entre 10% e 50% a maior parte do tempo

Por que não gastar 10% da potência com 10% de carga?

→ **Green Computing,
Green Data Centers**

Mais informações?

- **Acesso aos SPEC benchmarks:**
 - <http://www.spec.org/cpu2006/> (CPU)
 - http://www.spec.org/power_ssj2008/ (Power)
- **Green computing:**
 - IEEE Computer Magazine, Dezembro 2007
 - » www.computer.org/computer
 - Referência: “The Case for Energy-Proportional Computing”
 - » Luiz André Barroso, Google
 - » Urs Hölze, Google