

Benchmarks

Lucas Skywalker

20 de setembro de 2015

pg. 35

$$CPU_{time} = Instruction\ count \times CPI \times Clock\ cycle\ time$$

$$CPU_{time} = \frac{Instruction\ count \times CPI}{Clock\ cycle\ time}$$

pg. 48 Wordload é um conjunto de programas rodando em um computador que pode ser tanto uma coleção de aplicações rodadas pelo usuário ou construída de programas reais para aproximar o mix. Um wordload típico especifica ambos os programas e as frequências relativas.^a A elaboração é que quando avaliar dois computadores usando SPECratios, usar média geométrica para dar uma resposta normalizada sempre.

pg. 49 Tempo de execução é o produto de três fatores na tabela; Contagem de Instruções (em bilhões), Relógios (Clocks) por Instrução (CPI), e Tempo de Ciclo de Relógio (em nanosegundos). SPECratio é simplesmente o time de referência, o qual é suportado pelo SPEC, dividido pelo tempo de execução medido. ($SPECratio = \frac{Reference\ Time\ (Seconds)}{Execution\ Time\ (Seconds)}$). Exemplo: $SPECratio(perl) = \frac{9770}{637} = 15,3$

Descrição	Nome	Contagem de Instruções em Bilhões	CPI	Tempo de Ciclo de Relógio (Segundos $\times 10^{-9}$)	Tempo de Execução (Segundos)	Tempo Referencial (Segundos)	SPECratio
Processador interpretador de strings	perl	2118	0,75	0,4	637	9770	15,3

pg. 51 Lei de Amdahl, diz que aumento de desempenho possível dado por uma melhoria é limitado pelo montante que o dispositivo melhorado usa (suporta). É uma medida quantitativa da lei do retorno diminutivo.

$$Execution\ time\ after\ improvement = \frac{Execution\ time\ affected\ by\ improvement}{Amount\ of\ improvement} + Execution\ time\ unaffected$$

^aSPEC significa System Performance Evaluation Cooperative, Sistema Corporativo de Avaliação do Desempenho.

Pitfall

$$Execution\ time\ after\ improvement = \frac{80\ segundos}{n} + (100 - 80\ segundos)$$

$$20\ segundos = \frac{80\ segundos}{n} + (20)$$

$$0 = \frac{80\ segundos}{n}$$

pg. 52 Lei de Amdahl é por vezes, usada para argumentar para limites práticos para o número possível de processadores em paralelo. Ou seja, chega uma hora que existirá uma barreira de processadores trabalhando em paralelo.

Falácia: Computadores de baixo custo utilizam pouca energia.

Eficiência de energia importa bastante quando se fala de computadores custo baixo, por causa que workload em servidores variam muito. Utilização de CPU para servidores da Goodle, por exemplo, está entre 10% e 50% na maioria do tempo e 100% em menos que 1% do tempo.

Server Manufacturer	Micro Processor	<i>Total Cores</i> <i>Sockets</i>	Clock Rate	Peak Performance (ssj_ops)	100% Load Power	50% Load Power	<i>50% Load</i> <i>100% Power</i>	10% Load Power	<i>10% Load</i> <i>100% Power</i>	Active Idle Power	<i>Active/Idle</i> <i>100% Power</i>
HP	Xeon E5440	$\frac{8}{2}$	3.0 Ghz	308022	269 W	227 W	84%	174 W	65%	160 W	59%

Tabela 1: Até mesmo servidores que são utilizados 10%, queimam próximo de dois terços do seu pico de energia.

Desde que o workload de servidores varie mas use uma larga fração do seu pico de energia, Luiz Barroso e Urs Hölzle [2007] argumentaram que nós deveríamos redesenhar hardware para atingir "energia proporcional de computação". Se futuros servidores usaram 10% do seu pico de energia à 10% de worload, nós poderíamos reduzir o custo energético de central de dados, numa era pró-verde e de redução da emissão de CO_2 isso pode vir a calhar.

pg. 53 Milhões de instruções por segundo (MIPS) é uma medida de velocidade da execução de programas baseado no número de milhões de instruções.

$$MIPS = \frac{Contagem\ de\ instr}{Tempo\ de\ exec} \times 10^6$$

$$MIPS = \frac{Contagem\ de\ instr}{\frac{Contagem\ de\ instr \times CPI}{Tempo\ de\ exec}} \times 10^6$$

$$MIPS = \frac{Contagem\ de\ instr}{\frac{Contagem\ de\ instr \times CPI}{Clock\ Rate}} \times 10^6$$

pg. 54 **Tempo de execução** é a única opção impecável de medição da performance.

$$\frac{\text{Tempo (Segundos)}}{\text{Programa}} = \frac{\text{Contagem Instr.}}{\text{Programa}} \times \frac{\text{Ciclos de Relog.}}{\text{Contagem Instr.}} \times \frac{\text{Tempo (Segundos)}}{\text{Ciclos de Relog.}}$$

Individualmente esses fatores não definem performance. Mas o produto deles igual ao tempo de execução, vira então, uma medida confiável de performance.

1.12.1 Achar o CPI se temos um ciclo de relógio é 0,333 nanosegundos.

Name	Intr. Count 10 ⁹	Execution Time	Reference Time
a. bzip2	2389	750 s	9650 s
b. go	1658	700 s	10490 s

bzip2

$$CPUfreq = \frac{1}{0,333 \times 10^{-9}}$$

$$CPUfreq = 5,005005005 \times 10^9 \text{ Hz} = 5 \text{ Ghz}$$

$$CPI = 5 \times 10^9 \times \frac{ExecutionTime}{Instr. Count \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9}$$

$$CPI = 5 \times 10^9 \times \frac{750}{2389 \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9} = 1,569694433 \rightarrow 1,57 \text{ Ciclos por instr.}$$

go

$$CPUfreq = \frac{1}{0,333 \times 10^{-9}}$$

$$CPUfreq = 5,005005005 \times 10^9 \text{ Hz} = 5 \text{ Ghz}$$

$$CPI = 5 \times 10^9 \times \frac{ExecutionTime}{Instr. Count \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9} = 1,569694433 \rightarrow 1.6 \text{ Ciclos por instr.}$$

$$CPI = 5 \times 10^9 \times \frac{700}{1658 \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9} = 2,110977081 \rightarrow 2.1 \text{ Ciclos por instr.}$$

1.12.2

bzip2

$$CPUfreq = \frac{1}{0,333 \times 10^{-9}}$$

$$CPUfreq = 5,005005005 \times 10^9 \text{ Hz} = 5 \text{ Ghz}$$

$$CPI = 5 \times 10^9 \times \frac{ExecutionTime}{Instr. Count \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9}$$

$$CPI = 5 \times 10^9 \times \frac{750}{2389 \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9} = 1,569694433 \text{ Ciclos por instr.}$$

go

$$CPUfreq = \frac{1}{0,333 \times 10^{-9}}$$

$$CPUfreq = 5,005005005 \times 10^9 \text{ Hz} = 5 \text{ Ghz}$$

$$CPI = 5 \times 10^9 \times \frac{ExecutionTime}{Instr. Count \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9} = 1,569694433 \rightarrow 1.6 \text{ Ciclos por instr.}$$

$$CPI = 5 \times 10^9 \times \frac{700}{1658 \times 10^9} = 5 \times 10^9 \times 0,313938887 \times 10^{-9} = 2,110977081 \rightarrow 2.1 \text{ Ciclos por instr.}$$

1.12.3 bzip2

$$SPECratio = \frac{RefTime}{ExecTime} = \frac{9650}{750} = 12,86$$

go

$$SPECratio = \frac{RefTime}{ExecTime} = \frac{10490}{700} = 14,98$$

média geométrica

$$GEOmean = \sqrt[3]{12,866666667 \times 14,985714286} = \sqrt[3]{192,81} = 13,88$$

1.14.1 A seção 1.8 cita como pitfall a utilização de um subconjunto da equação da performance como uma medida de desempenho. Para ilustrar isso, considere as configurações abaixo para execução de um programa em diferentes processadores.

	Processador	Clock Rate	CPI	Nro. Instr.
a.	P1	4 Ghz	0.9	5.00E+06
	P2	3 Ghz	0.75	1.00E+06
b.	P1	3 Ghz	1.1	3.00E+06
	P2	2.5 Ghz	1.0	0.50E+06

Uma falácia comum, é considerar o computador com a maior frequência como tendo garantidamente a melhor performance. Veja se isso é verdade para os processadores da tabela acima.

$$Tempo_{a.P1} = Nro. Instr \times \frac{CPI}{ClockRate} = 5 \times 10^6 \times \frac{0.9}{4 GHz} = 1,125 \times 10^{-3} \text{ segundos}$$

$$Tempo_{a.P2} = Nro. Instr \times \frac{CPI}{ClockRate} = 1 \times 10^6 \times \frac{0.75}{3 GHz} = 0,25 \times 10^{-3} \text{ segundos}$$

$$Tempo_{b.P1} = Nro. Instr \times \frac{CPI}{ClockRate} = 3 \times 10^6 \times \frac{1.1}{3 GHz} = 1,1 \times 10^{-3} \text{ segundos}$$

$$Tempo_{b.P2} = Nro. Instr \times \frac{CPI}{ClockRate} = 0,5 \times 10^6 \times \frac{1.0}{2,5 GHz} = 0,2 \times 10^{-3} \text{ segundos}$$

Vemos que $Tempo_{a.P1} > Tempo_{a.P2}$. Tempo maior quer dizer que performance de $Tempo_{a.P1} < \text{performance de } Tempo_{a.P2}$.

Vemos também que $Tempo_{b.P1} > Tempo_{b.P2}$. Tempo maior quer dizer que performance de $Tempo_{b.P1} < \text{performance de } Tempo_{b.P2}$.

1.14.2 Outra falácia é considerar que o processador executando o maior número de instruções vai precisar de maior Tempo da CPU. Considerando que o processador P1 está executando a sequência de 10^6 instruções e que o CPI dos processadores P1 e P2 não mudam, **determine o número de instruções que o P2 pode executar** no mesmo tempo que P1 precisa para executar 10^6 instruções.

$$Tempo_{a.P1} = Nro. Instr \times \frac{CPI}{ClockRate} = 10^6 \times \frac{0.9}{4 GHz} = 0,225 \times 10^{-3} \text{ segundos}$$

$$Tempo_{a.P2} = Nro. Instr \times \frac{CPI}{ClockRate} = Nro. Instr \times \frac{0.75}{3 GHz} = 0,225 \times 10^{-3} \text{ segundos}$$

$$Nro. Instr = \frac{3 GHz \times 0,225 \times 10^{-3} \text{ segundos}}{0,75} = 0,9 \times 10^6 \text{ instruções}$$

$$Tempo_{b.P1} = Nro. Instr \times \frac{CPI}{ClockRate} = 10^6 \times \frac{1,1}{3 GHz} = 0,36 \times 10^{-3} \text{ segundos}$$

$$Tempo_{b.P2} = Nro. Instr \times \frac{CPI}{ClockRate} = Nro. Instr \times \frac{1,1}{2,5 GHz} = 0,36 \times 10^{-3} \text{ segundos}$$

$$Nro. Instr = \frac{3 GHz \times 0,225 \times 10^{-3} \text{ segundos}}{0,75} = 0,8 \times 10^6 \text{ instruções}$$

1.14.2a. Número de instruções para a.P2 aumentou muito pouco para a nova configuração.

1.14.2b. Número de instruções para b.P2 aumentou muito pouco para a nova configuração.

Comentário: Quer dizer que é uma falácia considerar genericamente que o processador com maior número de instruções levará (por causa disso) maior tempo de CPU.

1.14.3 Calcule os MIPS para cada processador.

$$MIPS_{a.P1} = \frac{ClockRate}{CPI \times 10^6} = \frac{4 \text{ GHz}}{0,9 \times 10^6} = 4444$$

$$MIPS_{a.P2} = \frac{ClockRate}{CPI \times 10^6} = \frac{3 \text{ GHz}}{0,75 \times 10^6} = 4000$$

$$MIPS_{b.P1} = \frac{ClockRate}{CPI \times 10^6} = \frac{3 \text{ GHz}}{1,1 \times 10^6} = 2727$$

$$MIPS_{b.P2} = \frac{ClockRate}{CPI \times 10^6} = \frac{2,5 \text{ GHz}}{1,0 \times 10^6} = 2500$$

Vemos que $MIPS_{a.P1} > MIPS_{a.P2}$. Porém $MIPS_{a.P1}$ leva mais tempo para executar 5 milhões de instruções do que $MIPS_{a.P2}$ que leva menos tempo para executar 1 milhão de instruções.

Vemos também que $MIPS_{b.P1} > MIPS_{b.P2}$.