ORGANIZAÇÃO E ARQUITETURA DE COMPUTADORES

Desempenho de Computadores

Profª. Fabiana F F Peres

Apoio: Camile Bordini

Motivação

"O desempenho do hardware normalmente é fundamental para a eficiência do sistema inteiro incluindo hardware e software"

Introdução

- Como medimos o desempenho de um computador?

- Quais os fatores que determina o desempenho de um computador?

Introdução

- É uma tarefa complicada
  - Variedade e **complexidade dos sistemas de software modernos**
    - Diferentes medidas de desempenho podem ser mais significativas e
    - Diferentes aspectos do sistema podem ser os mais determinantes do que outros na medição do desempenho geral do sistema
  - Várias **técnicas para melhorar desempenho** de hw

Introdução

- Alguns fatores que influenciam no **desempenho** do sistema:
  - Utilização por parte do compilador das instruções da máquina
  - Maneira como o hw implementa as instruções
  - Maneira como as memórias e os dispositivos de E/S se comportam
- Medir e comparar diferentes computadores é essencial para compradores e projetistas

- Deve-se tomar cuidado com propagandas
  - Algumas especificações do sistema não refletem melhorias reais de desempenho em certas aplicações

Definindo Desempenho

- Quando dizemos que um computador tem melhor desempenho do que outro o que queremos dizer?

- Embora pareça simples responder esta questão é **relativa**

Avião	Capacidade de passageiros	Autonomia de vôo (milhas)	Velocidade de Vôo (milhas por hora)	Vazão de passageiros (passageiros x mph)
Boeing 777	375	4.630	610	228.750
Boeing 747	470	4.150	610	286.700
BAC/Sud Concorde	132	4.000	1350	178.200
Douglas DC-8-50	146	8.720	544	79.424

- Qual avião tem melhor **desempenho**?

- Antes temos que definir "**desempenho**"
  - Avião com **maior velocidade**?
  - Avião com **maior autonomia**?
  - Avião com **maior capacidade**? ...

Avião	Capacidade de passageiros	Autonomia de vôo (milhas)	Velocidade de Vôo (milhas por hora)	Vazão de passageiros (passageiros x mph)
Boeing 777	375	4.630	610	228.750
Boeing 747	470	4.150	610	286.700
BAC/Sud Concorde	132	4.000	1350	178.200
Douglas DC-8-50	146	8.720	544	79.424

- Supondo que queremos definir **desempenho** em relação a **velocidade**. O "*mais rápido*" é:
  - O que leva um único passageiro de um local a outro no menor tempo:
    - *Concorde*
  - O que consegue levar 450 pessoas de um local ao outro no menor tempo:
    - *Boeing 747*

- O **desempenho** do computador também pode ser definido de várias maneiras:

I. Tempo entre o início e o término de uma tarefa (incluindo acesso a disco, memória, atividades de I/O, overhead do SO...): **tempo de resposta** ou **tempo de execução**
   - Quanto **menor**, melhor!

II. Quantidade total de tarefas realizadas em um intervalo de tempo: **throughput** ou **largura de banda**
   - Quanto **maior**, melhor!

Definindo Desempenho

- Assim, precisamos de diferentes métricas e diferentes tipos de aplicações para comparar *diferentes unidades computacionais*

- **Tempo de execução** → *Computadores embutidos e Desktops*

- **Throughput** → *Servidores*

Throughput x T_E

- Exemplo: as seguintes mudanças no sistema **aumentam o throughput**, **diminuem o tempo de execução** ou as **duas coisas**?

1. Substituir o processador em um computador por uma versão mais rápida
   - **Ambos são melhorados**

2. Incluir processadores adicionais em um sistema que usa múltiplos processadores para tarefas distintas – por exemplo, busca na web
   - **Apenas o throughput aumenta**

Throughput x T_E

- Diminuir o tempo de execução quase sempre melhora o throughput

- No entanto, no caso do exemplo 2, se a demanda para processamento fosse muito grande, o aumento do throughput poderia reduzir o tempo de espera na fila também, e com isso, melhorar o tempo de execução

Assim, em muitos sistemas computacionais, a mudança em um normalmente afeta o outro.

Desempenho Relativo

Desempenho Relativo

- O valor do **desempenho** de um computador X em relação ao seu T_E pode ser definido como:

$$Desempenhox = \frac{1}{TE_x}$$

- Quanto menor T_E , maior o desempenho

Portanto se : $Desempenhox > Desempenhoy$

Então: $TE_x < TE_y$

Desempenho Relativo

- Normalmente queremos comparar o desempenho quantitativamente entre dois computadores

"O computador *X* é *N* vezes *mais rápido* do que o computador *Y*"

"O tempo de execução de *Y* é N vezes maior que em *X*"

$$\frac{Desempenho_x}{Desempenho_y} = N$$

ou

$$\frac{TE_y}{TE_x} = N$$

Desempenho Relativo em %

"O computador *X* é *M% mais rápido* do que o *computador Y*"

$$M = 100 * \left(\frac{Desempenho_x - Desempenho_y}{Desempenho_y}\right)$$

"O tempo de execução de *Y* é *M%* maior que em *X*"

$$M = 100 * \left(\frac{TE_y - TEx}{TE_x}\right)$$

Exercício 1

Se um computador A executa um programa em 10 segundos e o computador B executa o mesmo programa em 15 segundos:

a) Quem tem melhor performance?

b) Qual o desempenho do computador A?

c) Qual o desempenho do computador B?

d) Quanto o computador A é mais rápido do que o computador B? E em porcentagem?

Resposta

Se um computador A executa um programa em 10 segundos e o computador B executa o mesmo programa em 15 segundos:

a) Quem tem melhor performance? A

b) Qual o desempenho do computador A? 1/10 = 0,1

c) Qual o desempenho do computador B? 1/15 = 0,0667

d) Quanto o computador A é mais rápido do que o computador B? E em porcentagem?

$$\frac{Desempenho_A}{Desempenho_B} = \frac{TE_B}{TE_A} = \frac{15}{10} = 1{,}5$$

$$M = 100 * (\frac{15 - 10}{10})$$

$$M = 50\%$$

A é 1,5 vezes mais rápido do que B. A é 50% mais rápido do que B

Tempo de Execução (T_E)

- Como já visto, inclui todo tempo gasto para se completar uma tarefa (acesso a disco, memória, atividades de I/O, …)
- Os Computadores são multitarefas e portanto executam vários programas ao mesmo tempo
- É importante fazer distinção entre o **tempo total para execução** de um programa e o **tempo gasto pela CPU** trabalhando para este programa

Tempo de CPU

- ***Tempo de CPU para um programa (T_{CPU})*** *é o* **tempo gasto pelo processador na execução das instruções dedicadas a um programa**, *(não inclui tempo gasto com I/O e nem tempo gasto para execução de outros programas)*

- O **T_{CPU}** pode ainda ser dividido em **Tempo de Usuário(T_U)** e **Tempo de sistema(T_S)** onde

$$T_U + T_S = T_{CPU}$$

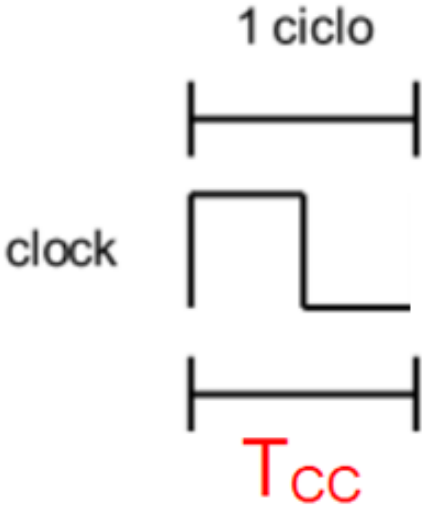
Tempo de CPU

- **Tempo de Usuário** é o tempo gasto pelo processador na execução de instruções do próprio programa
- **Tempo de Sistema** é o tempo gasto pelo SO para executar tarefas em benefício do próprio programa
- **É muito difícil diferenciar este dois tempos**
  - Atribuição de responsabilidades entre SO e programa
  - Diferenças de funcionalidades entre SO's diferentes

Detalhes do Desempenho

- Para avaliar o desempenho de um computador em relação ao Hardware, precisamos de uma **medida relacionada com velocidade do Hardware**:

  - **Tempo de Ciclo de Clock (T_{CC}):** intervalos de tempos discretos (medido em unidades de tempo)



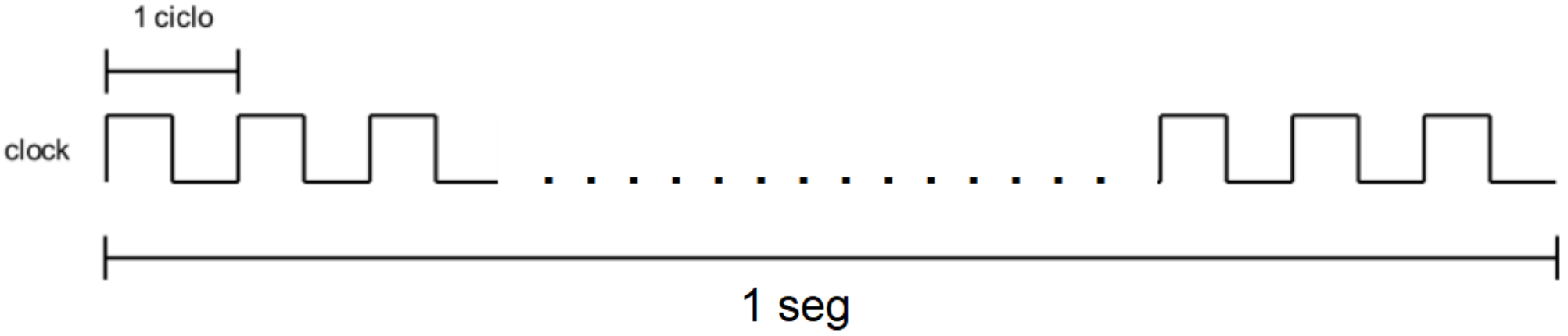
Obs: clocks determinam quando os eventos ocorrem no hardware

Obs: Ciclo de clock ou batidas de clock ou períodos de clock.

Detalhes do Desempenho

- A **quantidade de ciclos de clock** de um processador **num intervalo de tempo de 1 segundo** é chamado de **Frequência de clock (f)** (medido em Hertz)
  - Exemplo: 1GHz ≡ 1 bilhão de ciclos por seg ≡ 10^9 ciclos por seg

$$f = \frac{1}{T_{CC}}$$



Detalhes do Desempenho

- Como estamos interessados em avaliar a CPU, considerando a medida de T_{CC} e de f podemos dizer que:

$$T_{CPU} = N_{cc} * T_{CC} \qquad Ou \qquad T_{CPU} = \frac{N_{cc}}{f}$$

- Sendo:

N_{cc}: ***número de Ciclos de Clock para um programa***

Detalhes do Desempenho

- Essa fórmula deixa claro que o projetista de hardware pode **melhorar o desempenho reduzindo**:
  - O número de ciclos de clock exigidos para um programa (N_{cc})

    ou

  - O tempo de cada ciclo (T_{CC})

$$T_{CPU} = N_{cc} * T_{CC}$$

Obs: Infelizmente os projetistas normalmente têm de escolher entre as duas opções. Muitas técnicas que diminuem o número de ciclos de clock podem aumentar o tempo do ciclo de clock.

Exercício 2

Nosso programa favorito é executado em **10 segundos** no computador A, que possui um clock de **4 GHz (4x10^{9} hertz)**. Estamos tentando ajudar um projetista de computador a construir um computador B que executa esse mesmo programa em **6 segundos**. O projetista determinou que o aumento na velocidade de clock é possível, mas este aumento afetará o restante do projeto da CPU, fazendo com que o computador B exija **1,2 vezes mais ciclos de clock** do que o computador A para este mesmo programa. Que frequência de clock o projetista deverá buscar?

Desempenho da Instrução

Desempenho

- Fatores que irão influenciar no desempenho
  - Tempo de ciclo de clock ($\mathbf{T_{CC}}$)
  - Número de ciclos de clock ($\mathbf{N_{cc}}$) requeridos por um programa

- Sendo que **cada *instrução*** gasta um número de ciclos de clock para ser executada!

- Uma arquitetura tem um conjunto de *instruções*
- Normalmente dividimos as *instruções* em classes e levamos em conta o número médio de ciclos de clock de cada classe

Desempenho

- **CPI (*Ciclos de clock por instrução*)** é o número médio de CC que uma instrução (ou classe de instrução) gasta ao ser executada. Assim:

$$N_{cc} = \sum (CPI * C_i)$$

Então:

$$T_{CPU} = \sum (CPI * Ci) * T_{cc}$$

C_i : *quantidade de instruções da classe i*

Ou:

$$T_{CPU} = \sum (CPI * Ci) / f$$

Desempenho

- Cada compilador gera uma **sequência distinta de instruções** para um programa
  - Portanto o **número de CC** ($\mathbf{N_{cc}}$) requeridos por um programa depende do **código gerado pelo compilador**

Exercício 3

Suponha que temos duas implementações do mesmo conjunto de instruções (dois computadores).

- O computador A tem um T_{CC} de 250ps e um CPI de 2,0 para um determinado programa;
- O computador B tem um T_{CC} 500 ps o um CPI de 1,2 para o mesmo programa.

Que computador é mais rápido para este programa? Quanto mais rápido?

Desempenho

- Qual o CPI mínimo?
  - Embora pareça que o CPI mínimo seja 1,0, alguns processadores buscam e executam **várias instruções em um mesmo ciclo de clock**. Para refletir essa abordagem, alguns projetistas invertem o **CPI** para falar sobre o **IPC** (**Instruções por ciclo de clock**)
- Se um processador executa em média duas instruções por ciclo de clock, então
  - **IPC = 2**
  - **CPI= 0,5**

Resumindo

$$Performanœ_x = \frac{1}{T_{E_x}}$$

$$performance_x > performance_y$$

$$T_{E_x} < T_{E_y}$$

$$N = \frac{T_{E_y}}{T_{E_x}}$$

$$N = performance_x / performance_y$$

$$M = \left(100 * \left(performance_x - performance_y\right)/performance_y\right)$$

$$f_{cc} = 1/T_{CC}$$

$$T_{CPU} = n_{cc} * T_{cc}$$

$$T_{CPU} = \frac{n_{cc}}{f_{cc}}$$

$$n_{cc} = \sum (CPI * C_i)$$

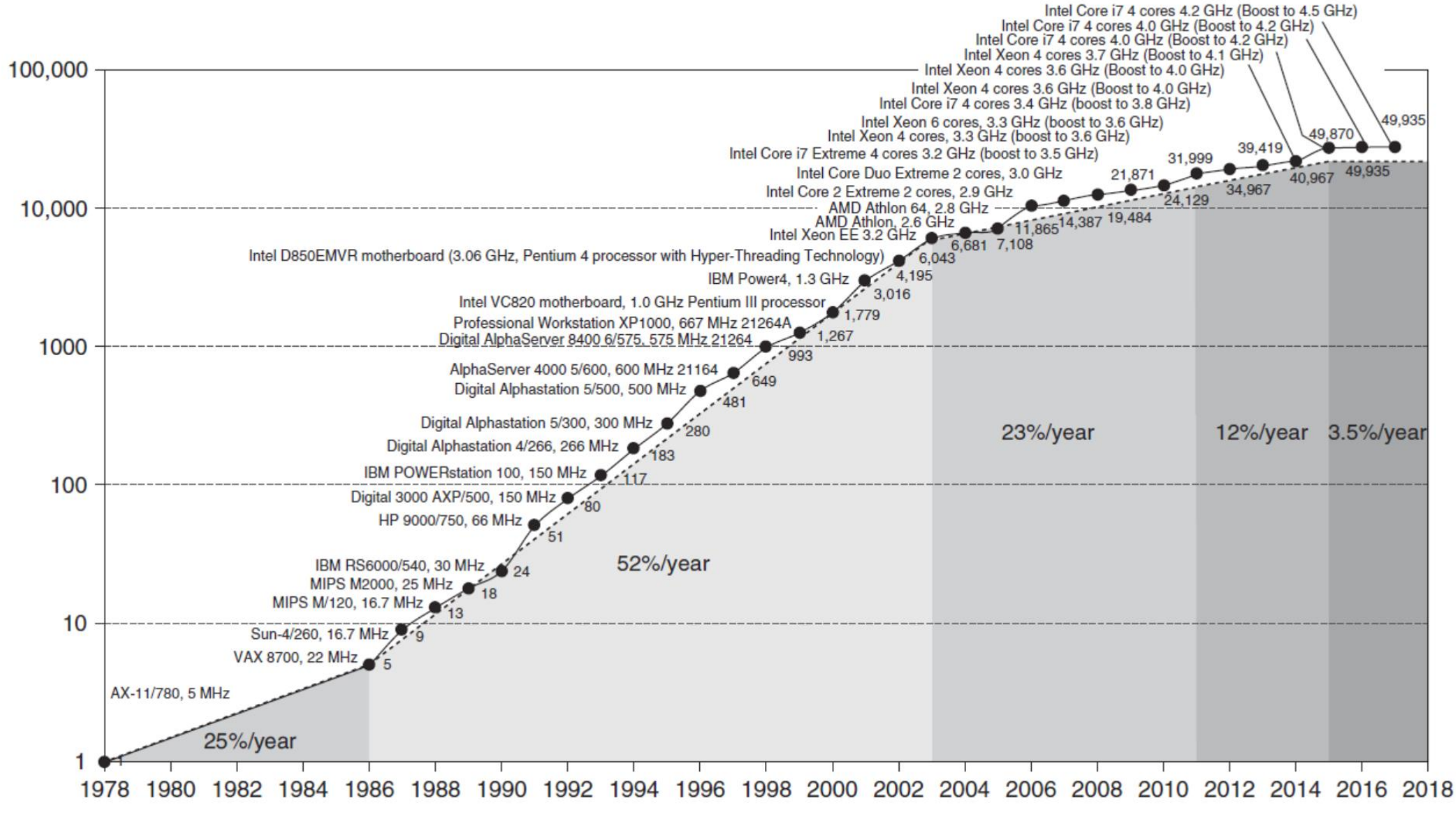
$$CPI \neq IPC$$

Instruções por ciclo de clock

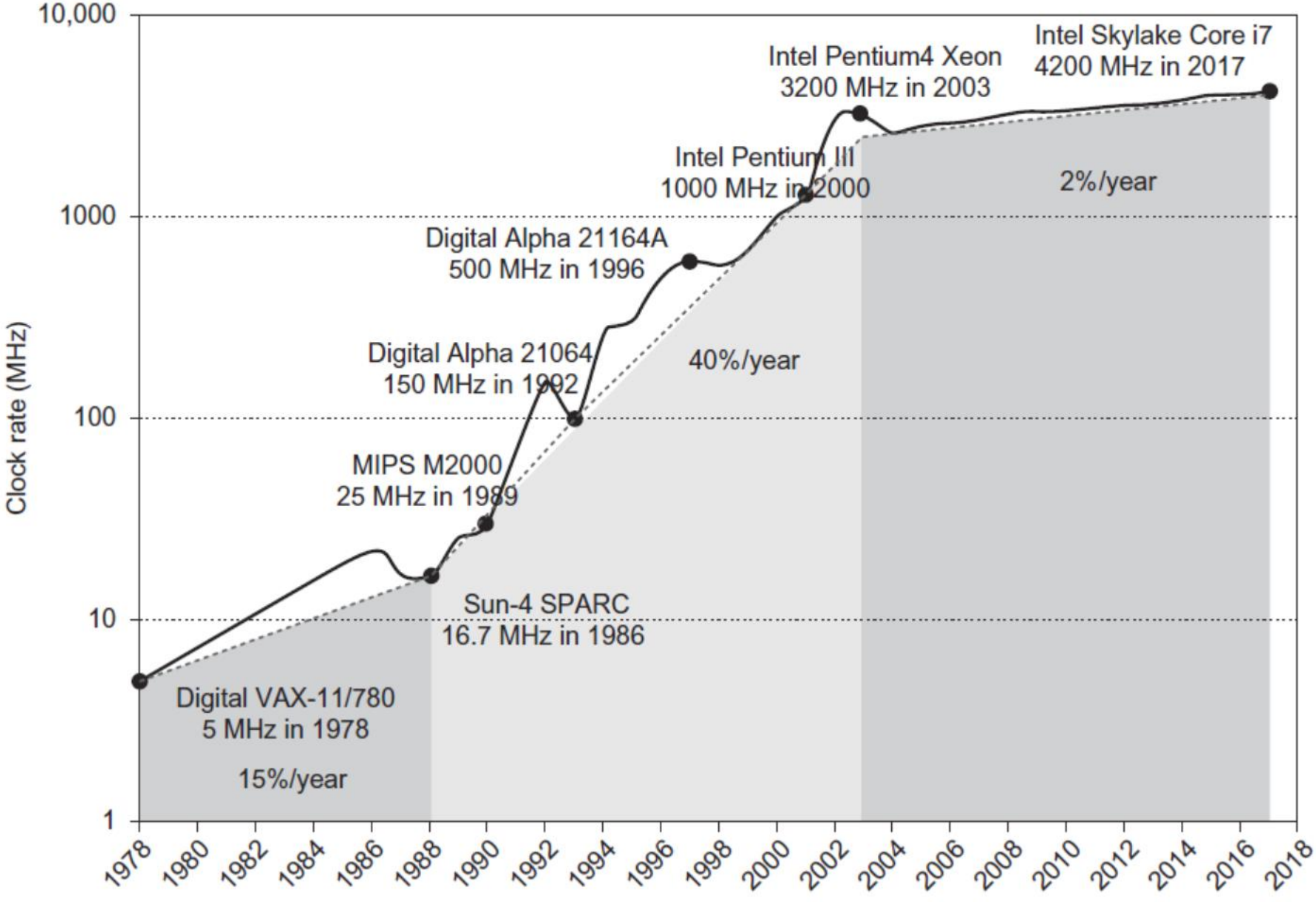
Observações

- Embora o tempo de ciclo do clock seja *tradicionalmente fixo*, (para economizar energia ou aumentar temporariamente o desempenho), os processadores de hoje podem *variar* suas taxas de clock,
- Então para os cálculos precisaríamos usar a **taxa de clock média** para um programa.
  - Ex: O Intel Core i7 aumenta temporariamente a frequência em cerca de 10% até que o chip aqueça muito (modo Turbo da Intel)

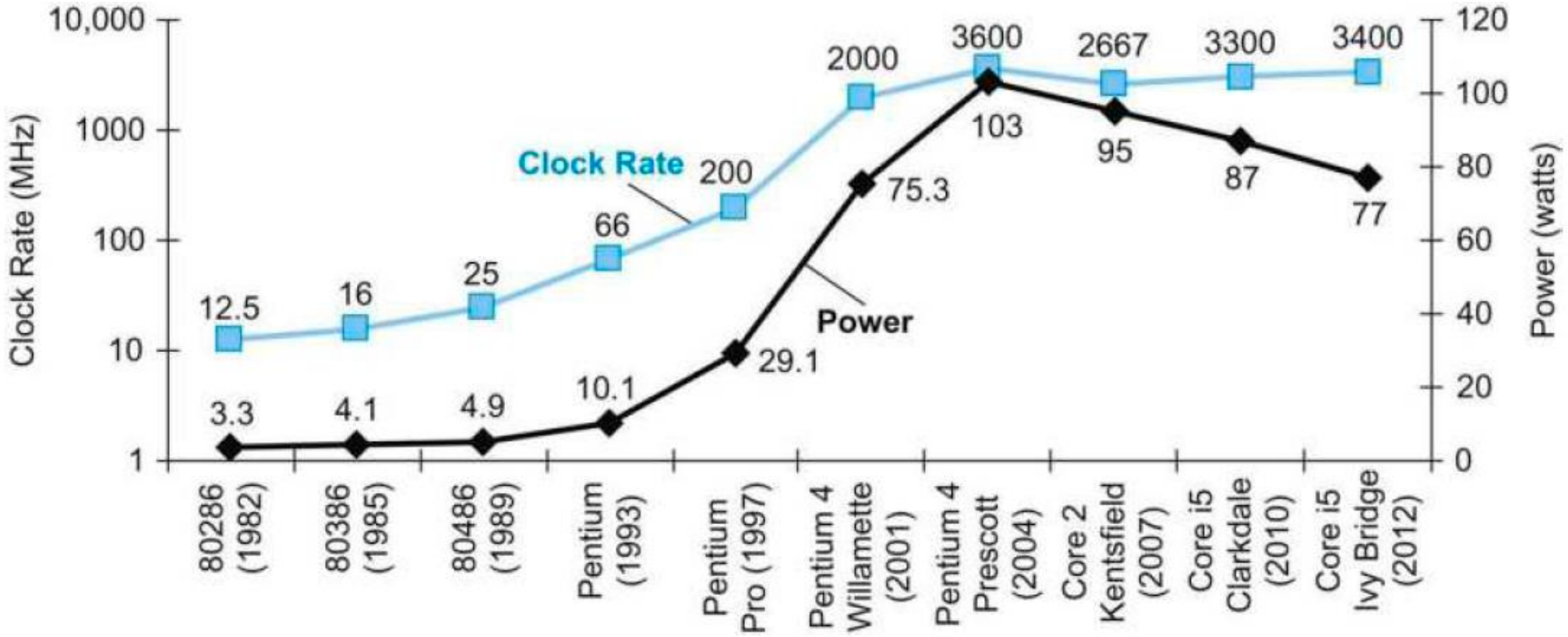
Desempenho do processador x Ano



Frequência de clock x Ano



Frequência de clock x família de processador



Computação em um Universo Paralelo

Até agora, a maioria dos softwares são como músicas escritas por um artista solo. Com a atual geração de chips estamos adquirindo um pouco de experiência com duetos, quartetos e outros pequenos conjuntos; mas compor uma obra para uma grande orquestra é um desafio diferente.

Brian Hayes, 2007

Benchmark

Benchmark

- [1]É o principal método para medir o desempenho de uma máquina física
  - Execução de um conjunto de programas representativos em diferentes computadores e redes
  - Os resultados são usados para avaliar o desempenho de um determinado sistema com um ***workload*** (conjunto de trabalho) bem definido
  - Bons benchmarks têm objetivos de desempenho e *workloads* bem definidos, que são mensuráveis e reprodutíveis.

1 https://www.pearsonhighered.com/assets/samplechapter/0/1/3/0/0130659037.pdf

Benchmark

- **Wikipédia:** (na computação) [...] ato de executar um programa de computador, um conjunto de programas ou outras operações, a fim de avaliar o desempenho relativo [...].

- **Michaelis:** marca de referência.

- **Patterson e Hennesy:** programas escolhidos para medir o desempenho de computadores.

Benchmark

- Quais os programas que devem ser usados para medir o desempenho de um computador?
  - **Idealmente, programas reais de um determinado usuário**

- Benchmarks são compostos por ***workloads*** *que* procuram estimar a performance dos *workloads* reais utilizados por determinados grupos de usuários.

Workloads

- Conjunto de programas executados em um computador
  - Coleção real de aplicativos executados por um usuário ou
  - Aplicativos construídos a partir de programas reais para aproximar a coleção de programas reais.

- Um **workload** típico especifica os programas e as frequências relativas deles

Workload -Tipos de Programas

a. **Sintéticos**: o código não faz nenhuma computação útil, não representam nenhuma aplicação, somente exercita alguns componentes do computador.

b. **Kernel:** baseia-se no fato de que a maior parte da computação de um programa é concentrada em uma pequena parte do código (Kernel) que é extraída do programa e usada como benchmark.

Workload -Tipos de Programas

c. **Algoritmos**: lógicas bem definidas, geralmente implementações de métodos conhecidos em computação numérica, como por exemplo os métodos de resolução de equações lineares, otimização combinatória

d. **Aplicações:** programas completos que resolvem problemas científicos bem definidos.

Pergunta

"Como comparar e resumir o desempenho através do *workload* escolhido?"

Exemplo

Considere o seguinte *workload*:

	Computador A	Computador B
Programa 1 (s)	1	10
Programa 2 (s)	1000	100
Tempo total (s)	1001	110

- Para o programa 1 a máquina A é **10 vezes** mais rápida que a B.
- Para o programa 2 a máquina B é **10 vezes** mais rápida que a A.

Exemplo (cont.)

- Utilizando o conceito de **performance relativa** a máquina B é **9,1 vezes** mais rápida que a A...
- ...se os dois programas são executados um mesmo número de vezes!

  - Pode-se utilizar também:
    - Média aritmética ponderada
    - Média geométrica

Média Geométrica

- É frequentemente utilizada quando compara-se diferentes itens
- Determina uma única "figura representativa" para esses itens (quando esses itens possuem múltiplas propriedades que possuem diferentes escalas numéricas).
- Determina uma "média" significativa para comparações
- Seu uso normaliza os valores

https://pt.wikipedia.org/wiki/M%C3%A9dia_geom%C3%A9trica

Resumindo o Desempenho

- Relatório que deve conter:
  - Especificação do hardware e do software de cada sistema computacional testado
  - Detalhes do *workload* utilizado

Tipos de Benchmark's

1. Benchmark para desktops
2. Benchmark para servidores
3. Benchmark para embarcados

Benchmarks - Exemplos

- SPEC
- Dhrystone
- MiBench
- EEMBC
- TPC
- LINPACK
- Bonnie++
- Phoronix Test Suite
- BDTI (*Berkeley Design Technology*)

SPEC

(*System Performance Evaluation Cooperative*)

- Esforço financiado e apoiado por vários fornecedores de computador para criar conjuntos padrão de benchmarks para sistemas computacionais

- Em 1989, a SPEC (SPEC89) criou um conjunto de benchmarks com foco no desempenho do processador
  - Evoluiu por várias gerações: SPEC89, SPEC92, SPEC95, SPEC2000, SPEC2006 E SPEC2017

https://www.spec.org/cpu/

SPEC2006

Consiste em um conjunto de:

- ...12 programas de **inteiros** (CINT2006)
  - Parte de um compilador C
  - Programa de xadrez
  - Simulação de computador quântico

- ...e 17 programas de **ponto flutuante** (CFP2006):
  - códigos para modelagem de elementos finitos
  - códigos de método de partículas para dinâmica molecular
  - códigos de álgebra linear esparsos para dinâmica de fluidos

SPEC

- Para simplificar o marketing, o SPEC decidiu informar um único número para resumir todos os 12 benchmarks de inteiros

- Usa média geométrica para comparar desempenho por apresentar uma série de vantagens como, por exemplo:
  - a escolha do computador de referência é irrelevante

Comparando CPUs

$$\frac{1600}{51,5} = 31,067$$

Benchmarks	Ultra 5 time (sec)	Opteron time (sec)	SPECRatio	Itanium 2 time (sec)	SPECRatio	Opteron/Itanium times (sec)	Itanium/Opteron SPECRatios
wupwise	1600	51.5	31.06	56.1	28.53	0.92	0.92
swim	3100	125.0	24.73	70.7	43.85	1.77	1.77
mgrid	1800	98.0	18.37	65.8	27.36	1.49	1.49
applu	2100	94.0	22.34	50.9	41.25	1.85	1.85
mesa	1400	64.6	21.69	108.0	12.99	0.60	0.60
galgel	2900	86.4	33.57	40.0	72.47	2.16	2.16
art	2600	92.4	28.13	21.0	123.67	4.40	4.40
equake	1300	72.6	17.92	36.3	35.78	2.00	2.00
facerec	1900	73.6	25.80	86.9	21.86	0.85	0.85
ammp	2200	136.0	16.14	132.0	16.63	1.03	1.03
lucas	2000	88.8	22.52	107.0	18.76	0.83	0.83
fma3d	2100	120.0	17.48	131.0	16.09	0.92	0.92
sixtrack	1100	123.0	8.95	68.8	15.99	1.79	1.79
apsi	2600	150.0	17.36	231.0	11.27	0.65	0.65
Geometric mean			20.86		27.12	1.30	1.30

CINT2006 no Intel Core i7

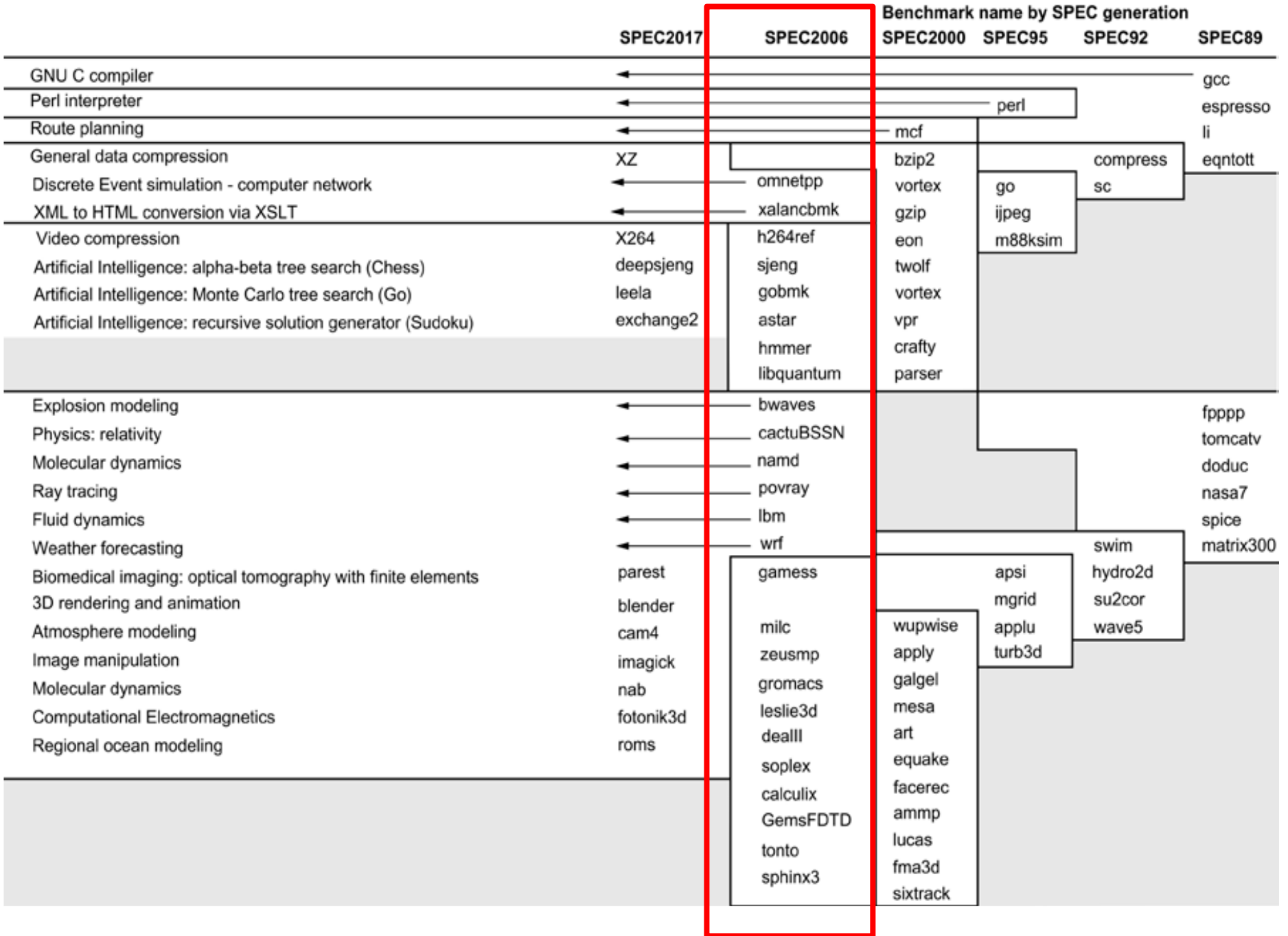
Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Interpreted string processing	perl	2252	0.60	0.376	508	9770	19.2
Block-sorting compression	bzip2	2390	0.70	0.376	629	9650	15.4
GNU C compiler	gcc	794	1.20	0.376	358	8050	22.5
Combinatorial optimization	mcf	221	2.66	0.376	221	9120	41.2
Go game (AI)	go	1274	1.10	0.376	527	10490	19.9
Search gene sequence	hmmer	2616	0.60	0.376	590	9330	15.8
Chess game (AI)	sjeng	1948	0.80	0.376	586	12100	20.7
Quantum computer simulation	libquantum	659	0.44	0.376	109	20720	190.0
Video compression	h264avc	3793	0.50	0.376	713	22130	31.0
Discrete event simulation library	omnetpp	367	2.10	0.376	290	6250	21.5
Games/path finding	astar	1250	1.00	0.376	470	7020	14.9
XML parsing	xalancbmk	1045	0.70	0.376	275	6900	25.1
Geometric mean	–	–	–	–	–	–	25.7

SPEC

$$\frac{\text{SPECRatio}_A}{\text{SPECRatio}_B} = \frac{\dfrac{\text{Execution time}_{\text{reference}}}{\text{Execution time}_A}}{\dfrac{\text{Execution time}_{\text{reference}}}{\text{Execution time}_B}} = \frac{\text{Execution time}_B}{\text{Execution time}_A} = \frac{\text{Performance}_A}{\text{Performance}_B} = 1.25$$

$$\text{Geometric mean} = \sqrt[n]{\prod_{i=1}^{n} sample_i}$$

SPEC2017 e sua evolução



SPEC CPU®2017 Overview / What's New?

$Id: overview.html 6491 2020-08-19 23:11:28Z JohnHenning $

This document introduces SPEC CPU®2017 via a series of questions and answers. SPEC CPU 2017 is a product of the SPEC® non-profit corporation (about SPEC) .

https://www.spec.org/cpu2017/Docs/overview.html

Referências Bibliográficas

- Patterson, David A. & Hennessy, John L. “Computer Organization & Design: the hardware/software interface”. 3nd ed., Morgan Kaufmann Publishers Inc., San Francisco, CA, 2005;