Universidade de Brasília
Departamento de Ciência da Computação

Aula 3
Desempenho - Benchmarks

“I can make any computer run as fast as you want!
If you remove the constraint of getting correct answers.”

Autor desconhecido

The Power Wall

Histórico da linha Intel:

Salto do Pentium IV

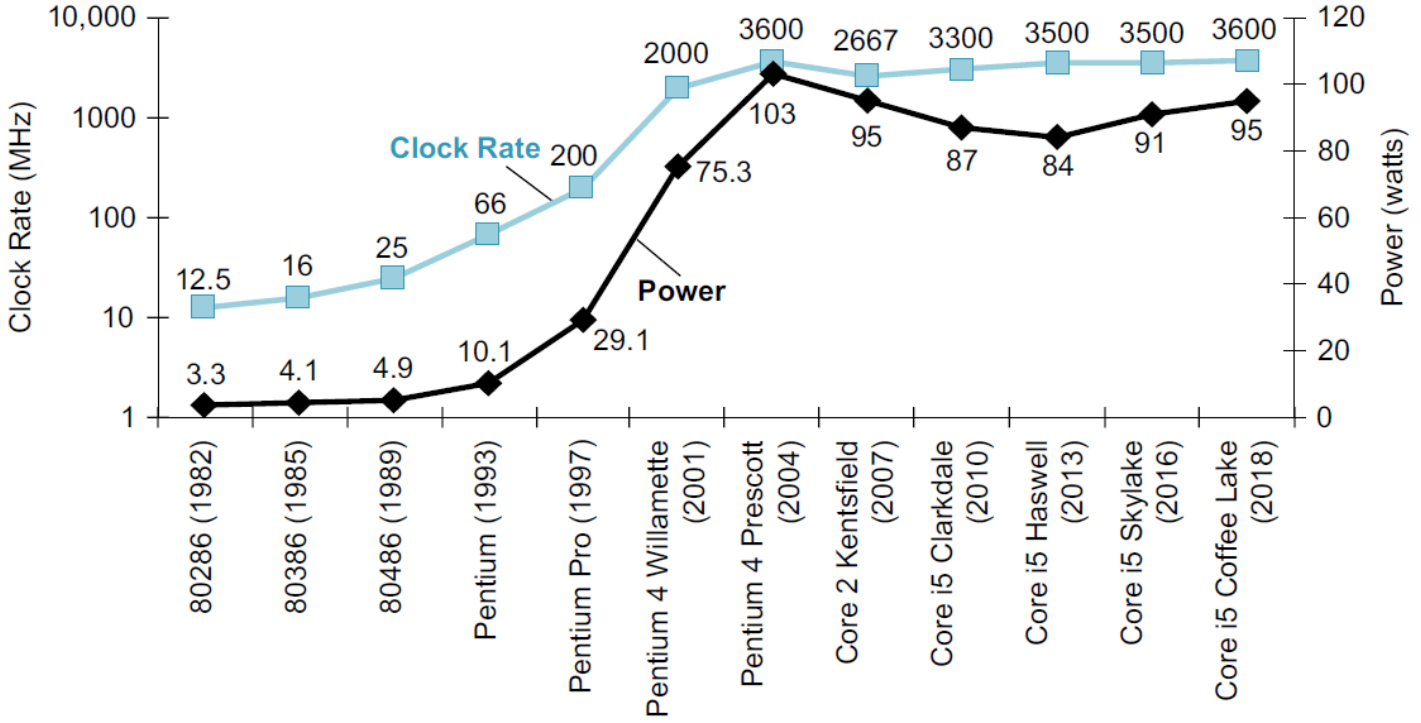
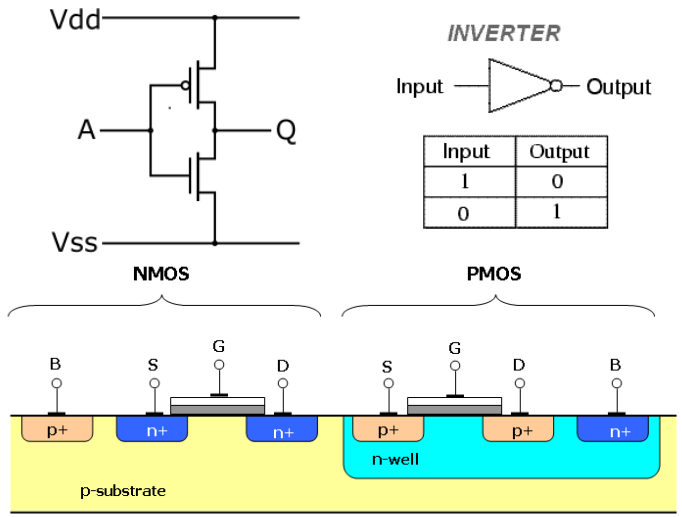


FIGURE 1.16 Clock rate and power for Intel x86 microprocessors over nine generations and 36 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline with lower clock rates and multiple processors per chip. The Core i5 pipelines follow in its footsteps.

Tecnologia CMOS

Complementary Metal-Oxide-Seminconductor (1963)



Dois tipos principais de dissipação de potência nesta tecnologia.

Estática: "vazamento" de corrente através dos transistores desligados e fugas de corrente reversa (diodo).

Hoje é responsável por 40% do consumo! (uso de tensões menores aumentam o estes efeitos!)

Dinâmica: corrente de carga dos capacitores (fanout) e do curto-circuito de chaveamento.

$$P \propto \frac{1}{2} f C V^2$$

P: Potência [W, J/s]
f : Frequência de chaveamento [Hz]
C : Carga capacitiva [F]
V : Tensão [V]

Ex.: Suponha que um novo, e mais simples, processador foi desenvolvido a partir de um processador mais complexo com as seguintes características:

A carga capacitiva é reduzida a 85% da carga original e a tensão e a frequência são reduzida em 15%. Qual o impacto na potência dinâmica dissipada do novo processador em comparação com o original?

Evolução do Desempenho Relativo

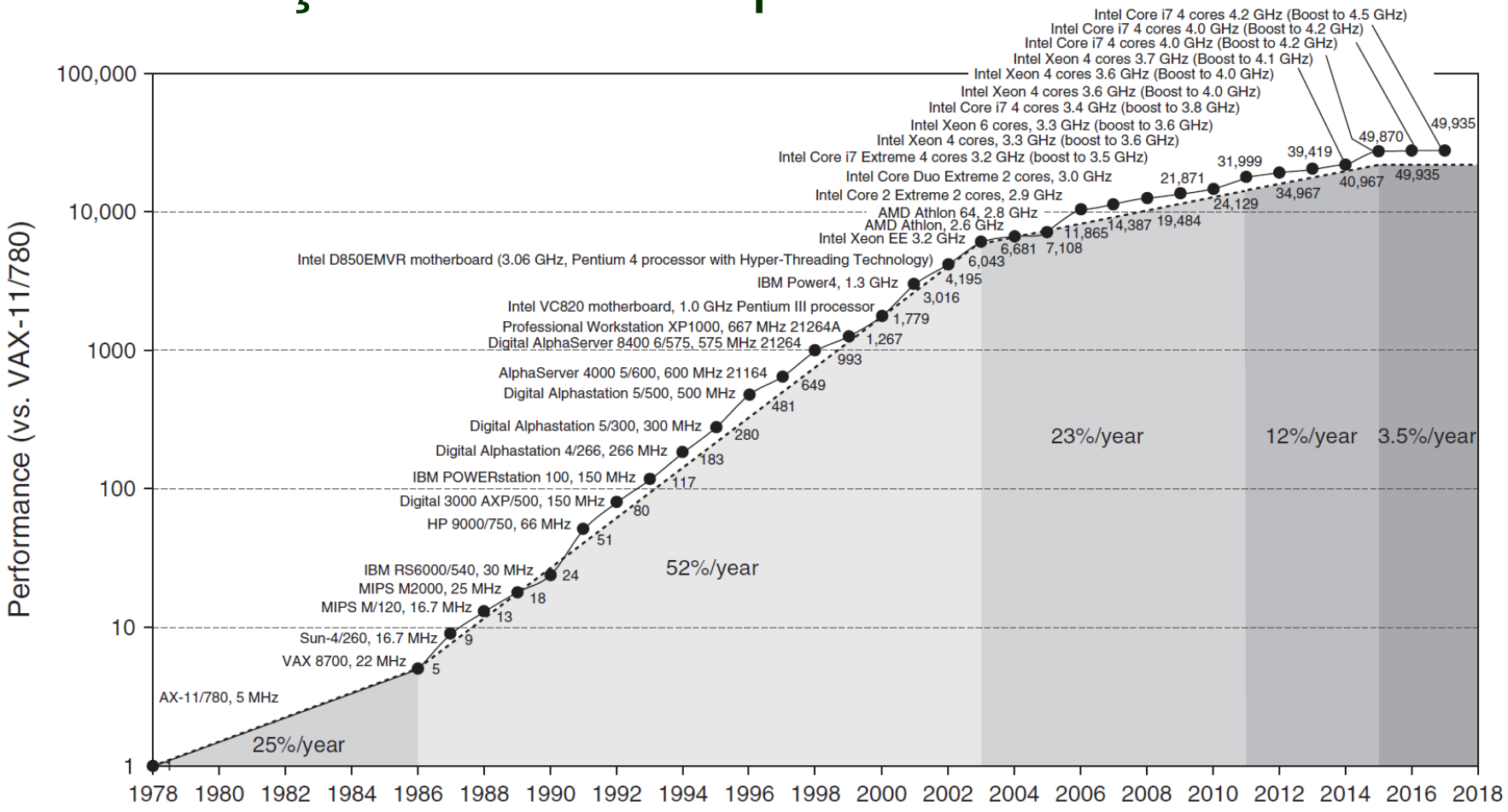
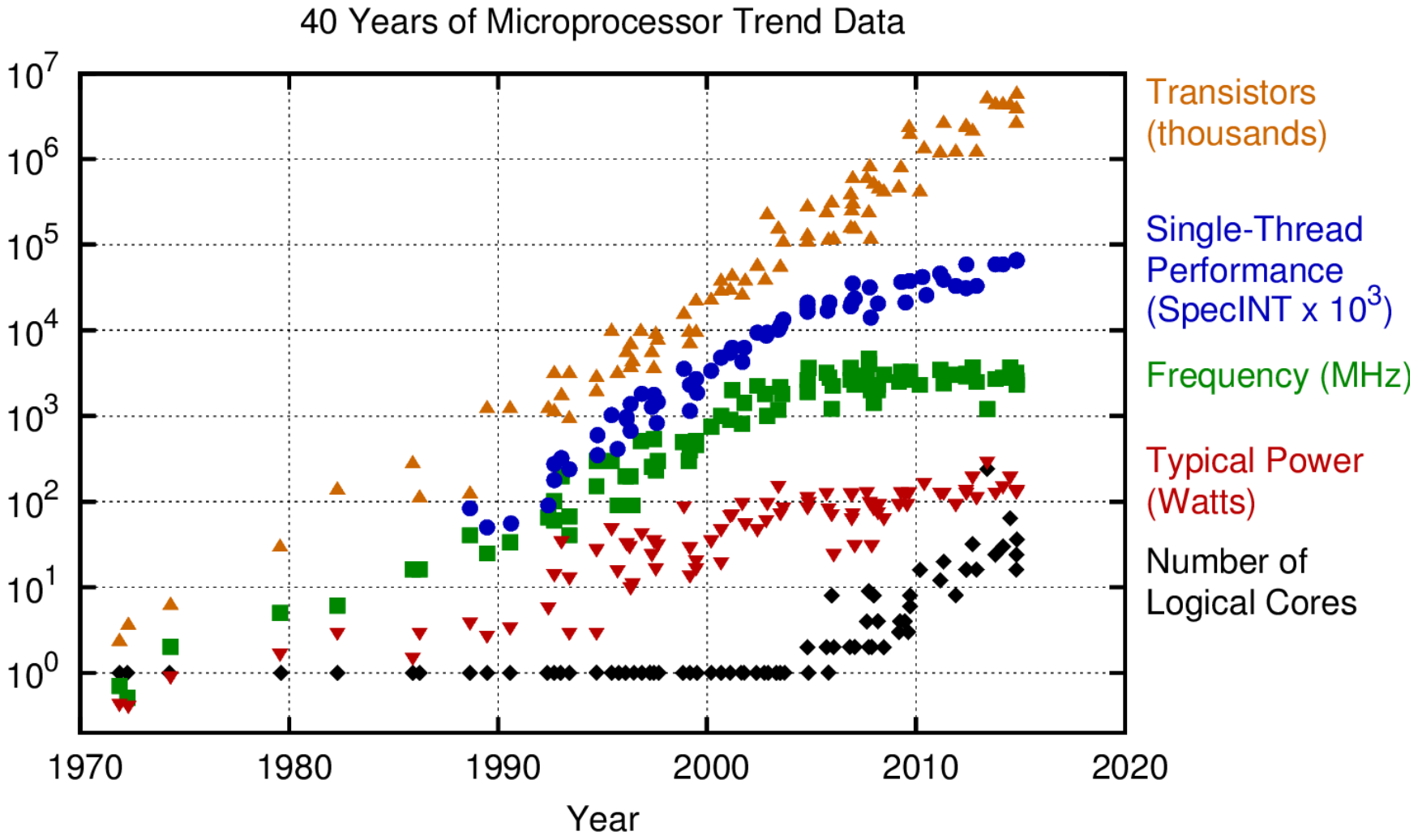


FIGURE 1.17 Growth in processor performance since the mid-1980s. This chart plots performance relative to the VAX 11/780 as measured by the SPECint benchmarks (see Section 1.11). Prior to the mid-1980s, processor performance growth was largely technology-driven and averaged about 25% per year. The increase in growth to about 52% since then is attributable to more advanced architectural and organizational ideas. The higher annual performance improvement of 52% since the mid-1980s meant performance was about a factor of seven larger in 2002 than it would have been had it stayed at 25%. Since 2002, the limits of power, available instruction-level parallelism, and long memory latency have slowed uniprocessor performance recently, to about 3.5% per year.

45 anos de desenvolvimento



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

Níveis de Paralelismo

- Paralelismo ao nível do computador (sistemas distribuídos)
  - Vários computadores ligados em rede, troca de mensagens, executando uma ou mais tarefas.
- Paralelismo ao nível do processador (multicores)
  - Vários processadores (núcleos), compartilhamento de memória, executando uma ou mais tarefas.
- Paralelismo ao nível de tarefa (threads) (superescalar)
  - Um processador executando 2 ou mais tarefas independentes ao mesmo tempo.
- Paralelismo ao nível de Instrução (pipeline)
  - Um processador executando 2 ou mais instruções de um mesmo programa (tarefa) ao mesmo tempo.

Benchmarks

- Avaliação de desempenho ideal: Aplicação Real

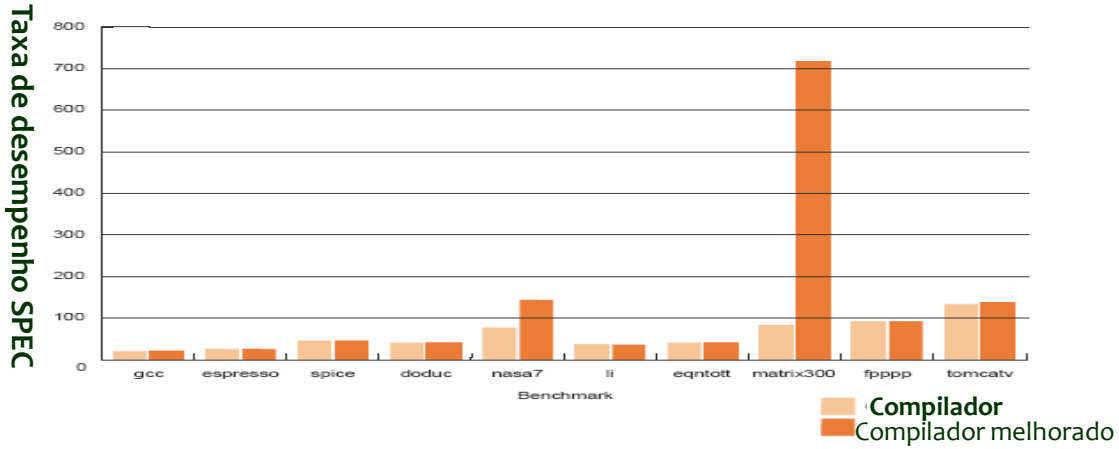
- *Workload*: Conjunto de programas típicos que caracterizam a utilização da máquina
  - Depende do usuário (ex. engenharia, desenvolvimento, finanças, jogos, etc)
  - Difícil padronização para fins de comparação

- *Benchmarks*: Conjuntos de programas especificamente escolhidos para medir o desempenho em determinada categoria de aplicações.
  - Workload que o usuário espera que preveja o desempenho no workload real

Benchmarks

- Real
  - Aplicações reais
  - Grande complexidade e variação das instruções
  - Difícil de medir pois depende muito de I/O

- Sintético
  - Pequenos fragmentos de código
  - Fáceis de padronizar, aplicar e medir
  - Adequado na etapa de desenvolvimento (arquitetura, compilador)
  - Problema: Facilmente otimizável (forçado)



Qual benchmark usar?

- Desempenho:
  - Tempo de resposta
    - Aplicações científicas, matemáticas, gráficas, etc.
    - Tempo de CPU
  - Vazão
    - Aplicações em servidores (Banco de dados, Web server, etc.)
    - Throughput
  - Outros fatores de desempenho: Potência, custo,….

Exemplo

- Dada a tabela comparativa abaixo do tempo de execução de dois programas em três computadores diferentes.

	Computador A	Computador B	Computador C
Programa Prog1	1	10	20
Programa Prog2	1000	100	20

A é 10 vezes mais rápido que B para o Prog1
B é 10 vezes mais rápido que A para o Prog2
A é 20 vezes mais rápido que C para o Prog1
C é 50 vezes mais rápido que A para o Prog2
B é 2 vezes mais rápido que C para o Prog1
C é 5 vezes mais rápido que B para o Prog2

Como resumir?

	Computador A	Computador B	Computador C
Prog1	1	10	20
Prog2	1000	100	20
Tempo total (seg)	1001	110	40

- Tempo Total de Execução

B é 9.1 vezes mais rápido que A para os programas Prog1 e Prog2

C é 25 vezes mais rápido que A para os programas Prog1 e Prog2

C é 2.75 vezes mais rápido que B para os programas Prog1 e Prog2

Considera que cada um dos N programas do workload é executado o mesmo número de vezes.

Logo:

Média Aritmética

$$tempo_A = \frac{1}{N}\sum_{i=1}^{N} tempo_i$$

Porém:

- Se o programa Prog1 for muito mais frequentemente executado na aplicação real que o programa Prog2?

- Média Ponderada:

$$tempo_A = \sum_{i=1}^{N} P_i \times tempo_i \qquad \sum_{i=1}^{N} P_i = 1$$

Considera que cada um dos N programas do workload é executado de acordo com a sua Probabilidade de Ocorrência (P_i) (Frequência Relativa).

Exemplo: programas	Máquinas			Ponderações		
	A	B	C	P_1	P_2	P_3
Prog1	1	10	20	0.5	0.909	0.999
Prog2	1000	100	20	0.5	0.091	0.001

Média P_1	500.5	55	20
Média P_2	91.91	18.19	20
Média P_3	2	10.09	20

P_1 Média Aritmética
P_2 Normalização para Máquina B
P_3 Normalização para Máquina A

} Considera que cada programa ocupa a mesma quantidade tempo naquela máquina

Tempo de Execução Normalizado

- Nova abordagem para mix desigual de programas no workload
- Escolhe-se uma máquina para ser a Máquina Base

$$\bar{t}_i = \frac{tempo_{A_i}}{tempo_{Base_i}}$$

Tempo de Execução Normalizado do programa i executado na maquina A em relação à Maquina Base

Tempo de Execução Normalizado Médio

Média Geométrica dos tempos normalizados

$$\bar{t_A} = \sqrt[N]{\prod_{i=1}^{N} \bar{t_i}}$$

Porque não usar a média aritmética ponderada?
Pq depende das ponderações usadas na máquina base.
Na Média Geométrica não ocorre isso!

$$\frac{MG(X_i)}{MG(Y_i)} = MG\left(\frac{X_i}{Y_i}\right)$$

	Normalizado para A			Normalizado para B			Normalizado para C		
	A	B	C	A	B	C	A	B	C
Prog1	1.0	10.0	20.0	0.1	1.0	2.0	0.05	0.5	1.0
Prog2	1.0	0.1	0.02	10.0	1.0	0.2	50.0	5.0	1.0
Média Aritm.	1.0	5.05	10.01	5.05	1.0	1.1	25.03	2.75	1.0
Média Geom.	1.0	1.0	0.63	1.0	1.0	0.63	1.58	1.58	1.0

- Média Aritmética de tempos de execução normalizado não é consistente (depende da Máquina Base)
- Média Geométrica é consistente! Porém não reflete o tempo de execução.

  A é 9.1 vezes mais lento que B; B é 2.75 vezes mais lento que C para workloads iguais

Fator Importante para comparação:

- REPRODUTIBILIDADE
  - Fornecer subsídios para que outros consigam repetir o mesmo experimento validando o resultado apresentado.
  - Necessário
    - Descrição detalhada do workload
    - Descrição das diretivas de compilação
    - Descrição do Sistema Operacional
    - Configuração completa da máquina

Ex.:
AMD_Sempron_Benchmarks_May06.pdf
AMD_Athlon_64_Benchmarks_May06.pdf

Benchmarks

Padrões da Indústria (Auditáveis e Verificáveis):

- Standard Performance Evaluation Corporation (SPEC)
- Business Applications Performance Corporation (BAPCo)
- Transaction Processing Performance Council (TPC)
- Embedded Microprocessor Benchmark Consortium (EEMBC)

Há vários outros comerciais e OpenSource:

Dhrystone: Desempenho em operações com inteiros
Whetstone: Desempenho em operações com ponto flutuante
LINPACK: Usados no TOP500 (xFLOPS)
…

Benchmarks

- SPEC (*Standard Performance Evaluation Corporation*)
  - Criada em 1988 (SPEC89)
  - Função padronizar workloads para diversos tipos da análises de desempenho baseado no paradigma da aplicação real.
    - **CPU** (CINT e CFP)
    - **Graphics and Workstation Performance**
    - **High Performance Computing, OpenMP, MPI**
    - **Java Client/Server**
    - **Mail Servers**
    - **Storage**
    - **Power**
    - **Web Servers**
  - Máquina Base:
    - SPEC95: Sun SPARC 10 (até Jul. 2000)
    - SPEC2000: Sun UltraSPARC 5 (até Fev. 2007)
    - SPEC2006: Sun Ultra Enterprise 2 (até Jan 2018)
    - SPEC2017: Sun Fire V490 with 2100 MHz UltraSPARC-IV+ chip

  - Precisa comprar licença para utilizar ☹
  - Resultados são gratuitos ☺

Ex.: SPECINT2017 em um Intel Xeon E5-2650L

Description	Name	Instruction Count x 10^9	CPI	Clock cycle time (seconds x 10^{-9})	Execution Time (seconds)	Reference Time (seconds)	SPECratio
Perl interpreter	perlbench	2684	0.42	0.556	627	1774	2.83
GNU C compiler	gcc	2322	0.67	0.556	863	3976	4.61
Route planning	mcf	1786	1.22	0.556	1215	4721	3.89
Discrete Event simulation - computer network	omnetpp	1107	0.82	0.556	507	1630	3.21
XML to HTML conversion via XSLT	xalancbmk	1314	0.75	0.556	549	1417	2.58
Video compression	x264	4488	0.32	0.556	813	1763	2.17
Artificial Intelligence: alpha-beta tree search (Chess)	deepsjeng	2216	0.57	0.556	698	1432	2.05
Artificial Intelligence: Monte Carlo tree search (Go)	leela	2236	0.79	0.556	987	1703	1.73
Artificial Intelligence: recursive solution generator (Sudoku)	exchange2	6683	0.46	0.556	1718	2939	1.71
General data compression	xz	8533	1.32	0.556	6290	6182	0.98
Geometric mean	–	–	–	–	–	–	2.36

FIGURE 1.18 SPECspeed 2017 Integer benchmarks running on a 1.8 GHz Intel Xeon E5-2650L. As the equation on page 35 explains, execution time is the product of the three factors in this table: instruction count in billions, clocks per instruction (CPI), and clock cycle time in nanoseconds. SPECratio is simply the reference time, which is supplied by SPEC, divided by the measured execution time. The single number quoted as SPECspeed 2017 Integer is the geometric mean of the SPECratios. SPECspeed 2017 has multiple input files for perlbench, gcc, x264, and xz. For this figure, execution time and total clock cycles are the sum running times of these programs for all inputs.

Alguns resultados on-line

http://www.spec.org

http://www.cpubenchmark.net/high_end_cpus.html

http://www.sisoftware.net/

http://www.tomshardware.com/

Lei de Amdahl (1967)

$$t_{exec}\text{após melhoria} = t_{exec}\text{não afetado} + \frac{t_{exec}afetado}{quantidade\ de\ melhoria}$$

- Exemplo:

  “Suponha que um programa seja executado em 100 segundos em uma máquina, com multiplicação responsável por 80 segundos desse tempo. O quanto precisamos melhorar a velocidade da multiplicação se queremos que o programa seja executado 4 vezes mais rápido?”

  Que tal torná-lo 5 vezes mais rápido?

Exercícios

- Suponha que melhoramos uma máquina fazendo todas as instruções de ponto flutuante serem executadas cinco vezes mais rápido. Se o tempo de execução de algum benchmark antes da melhoria do ponto flutuante é 10 segundos, qual será o aumento de velocidade se metade dos 10 segundos é gasta executando instruções de ponto flutuante?

- Estamos procurando um benchmark para mostrar a nova unidade de ponto flutuante descrita acima e queremos que o benchmark geral mostre um aumento de velocidade de 3 vezes. Um benchmark que estamos considerando é executado durante 100 segundos com o hardware de ponto flutuante antigo. Quanto do tempo de execução as instruções de ponto flutuante teriam que considerar para produzir nosso aumento de velocidade desejado nesse benchmark?

- Suponha que estejamos considerando um aperfeiçoamento para o processador de um sistema servidor Web. A nova CPU é 10 vezes mais rápida que o processador original para computação do serviço Web. Supondo que a CPU original esteja ocupada 40% do tempo e fique esperando por E/S em 60% do tempo, qual será a aceleração global obtida com o aperfeiçoamento?

Conclusões

- O desempenho é específico a um determinado programa
  - O tempo de execução total é um resumo consistente do desempenho

- Para uma determinada arquitetura com 1 único processador, os aumentos de desempenho vêm de:
  - aumentos na frequência de clock (sem efeitos de CPI adversos)
  - melhorias na organização do processador que diminuem a CPI
  - melhorias no compilador que diminuem a CPI e/ou a contagem de instruções
  - escolhas de algoritmo/linguagem que afetam a contagem de instruções

- Embora tenhamos nos detido nas análises de desempenhos temporais, projetos reais devem considerar outros fatores de desempenho.

  Projeto de Alto Desempenho x Projeto de Baixo Custo