Universidade de Brasília
Departamento de Ciência da Computação

Aula 3
Desempenho - Benchmarks

The Power Wall

Histórico da linha Intel:

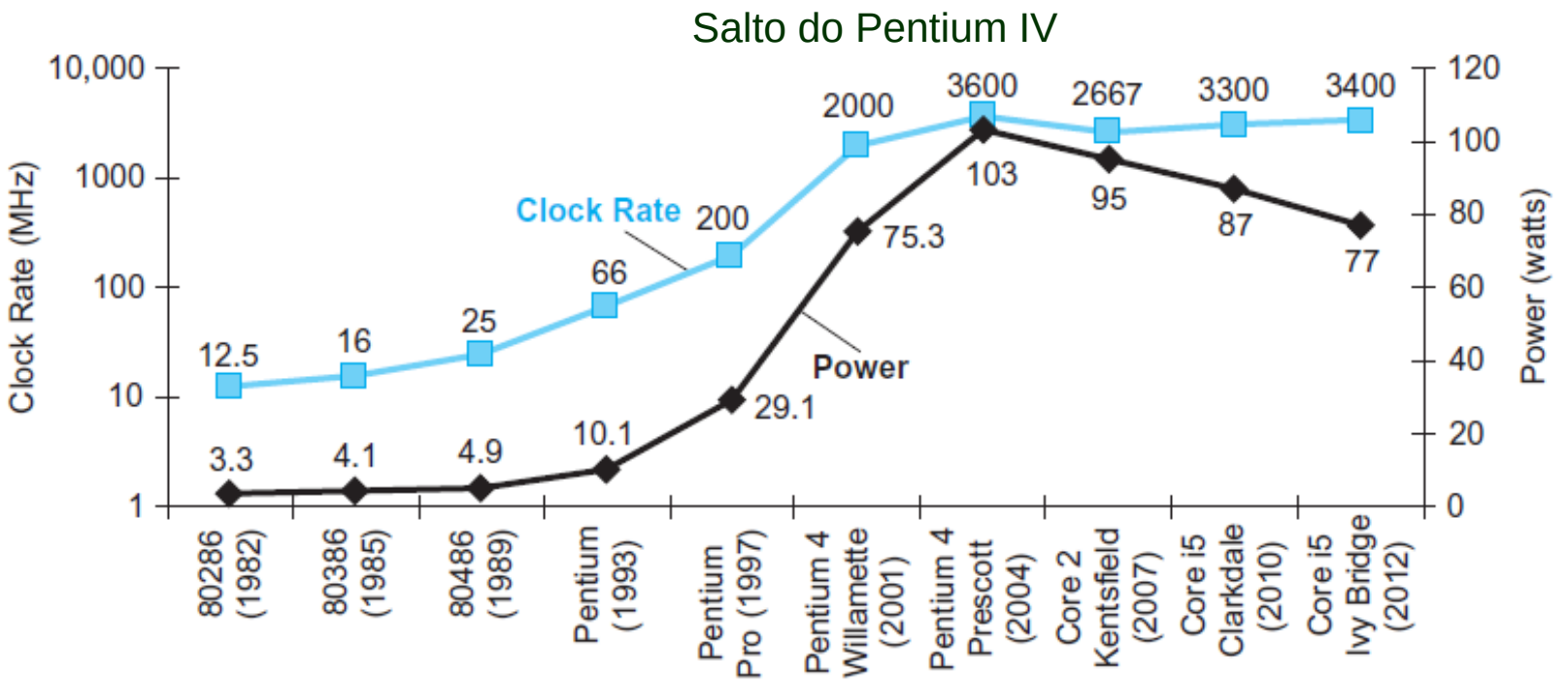
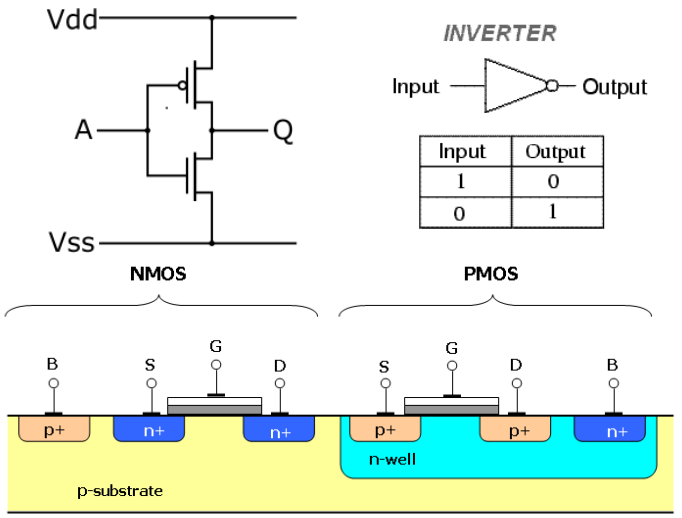


FIGURE 1.16 Clock rate and Power for Intel x86 microprocessors over eight generations and 25 years. The Pentium 4 made a dramatic jump in clock rate and power but less so in performance. The Prescott thermal problems led to the abandonment of the Pentium 4 line. The Core 2 line reverts to a simpler pipeline 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! (tensões menores)

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

$$P = \alpha . f . C . V^2$$

P: Potência [W, J/s]
f: Frequência de chaveamento [Hz]
C: Carga capacitiva [F]
V: Tensão [V]
α: Fator de uso do transistor.

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?

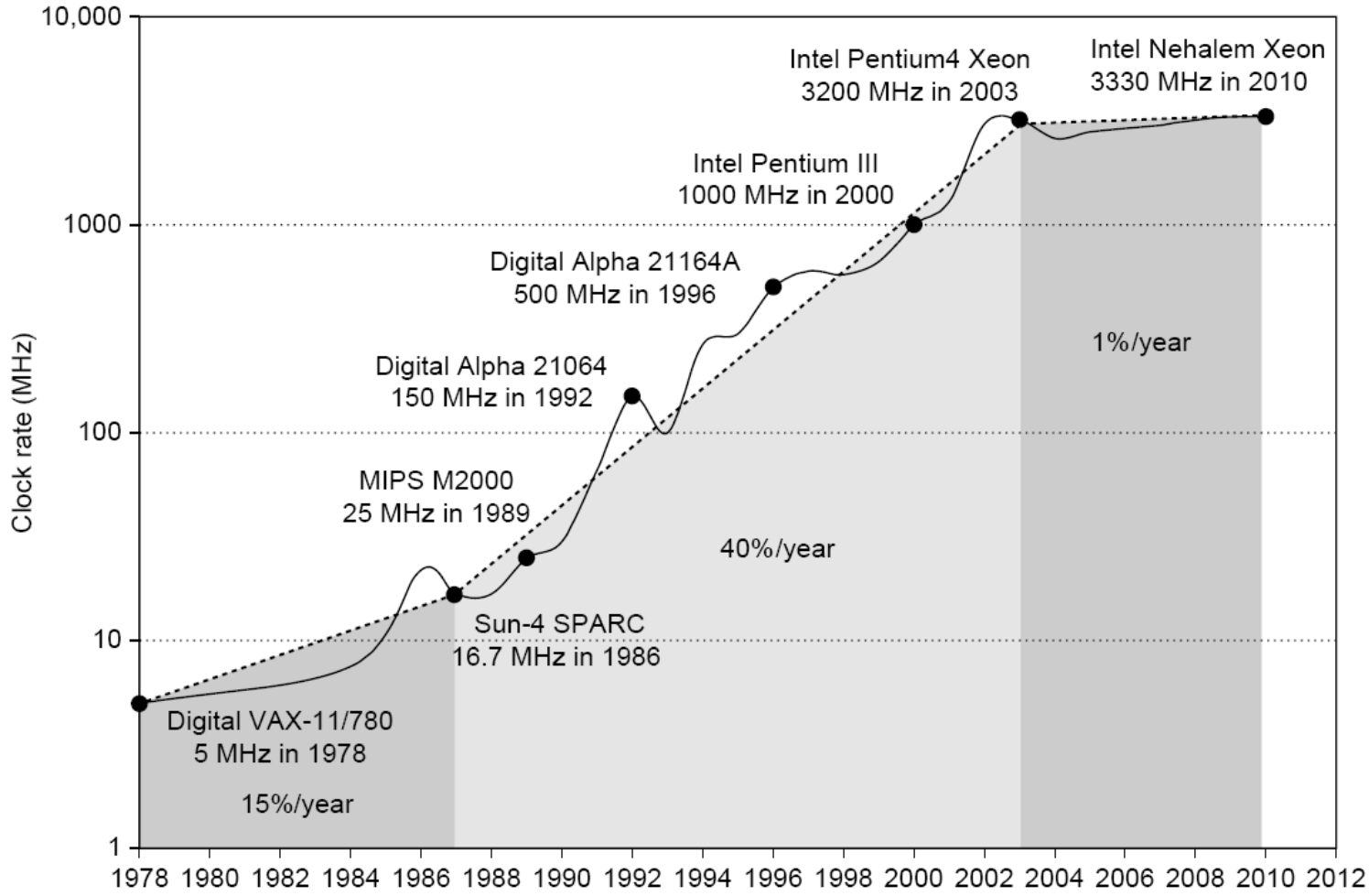


Figure 1.11 Growth in clock rate of microprocessors in Figure 1.1. Between 1978 and 1986, the clock rate improved less than 15% per year while performance improved by 25% per year. During the "renaissance period" of 52% performance improvement per year between 1986 and 2003, clock rates shot up almost 40% per year. Since then, the clock rate has been nearly flat, growing at less than 1% per year, while single processor performance improved at less than 22% per year.

Evolução do Desempenho Relativo

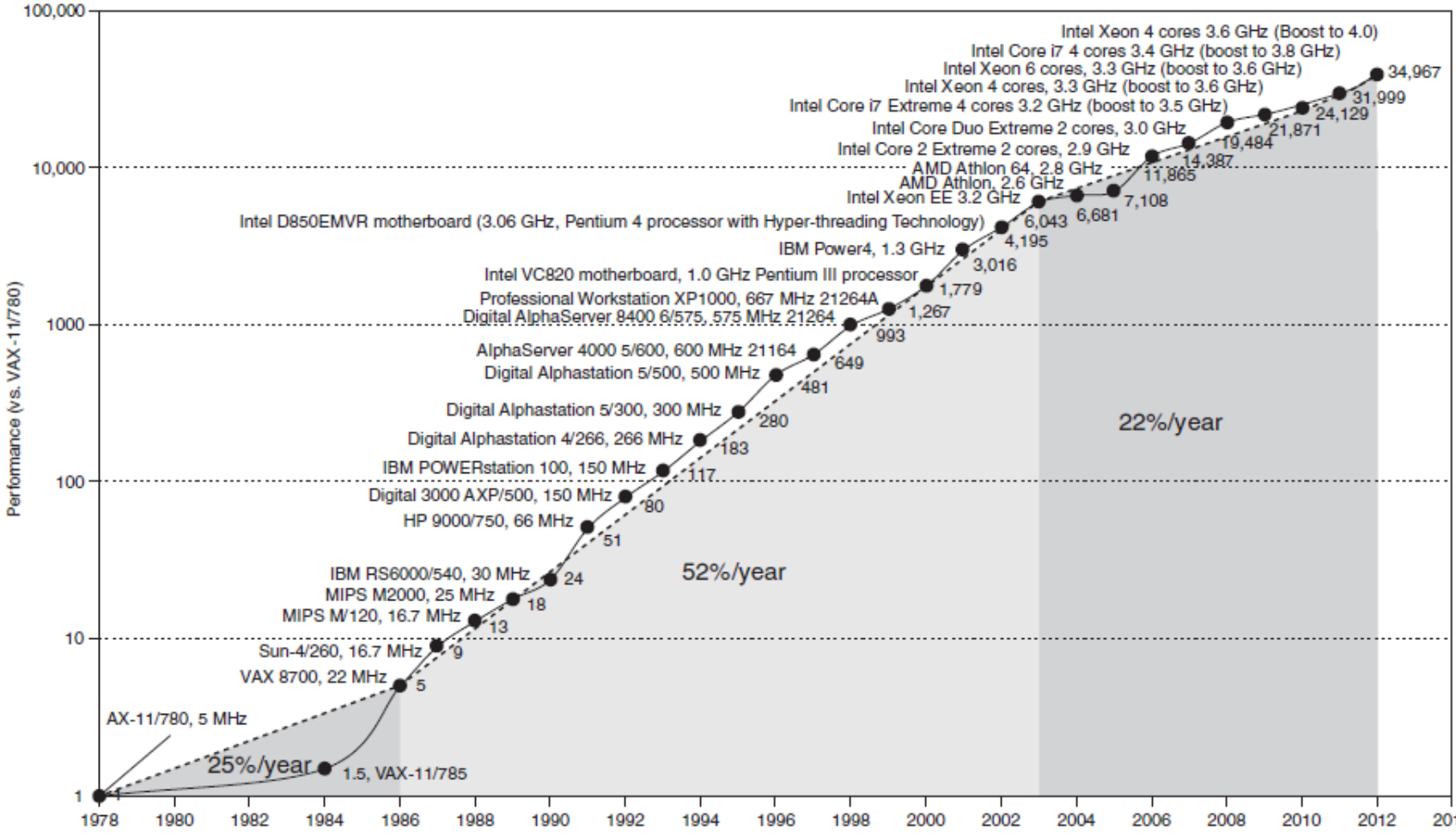
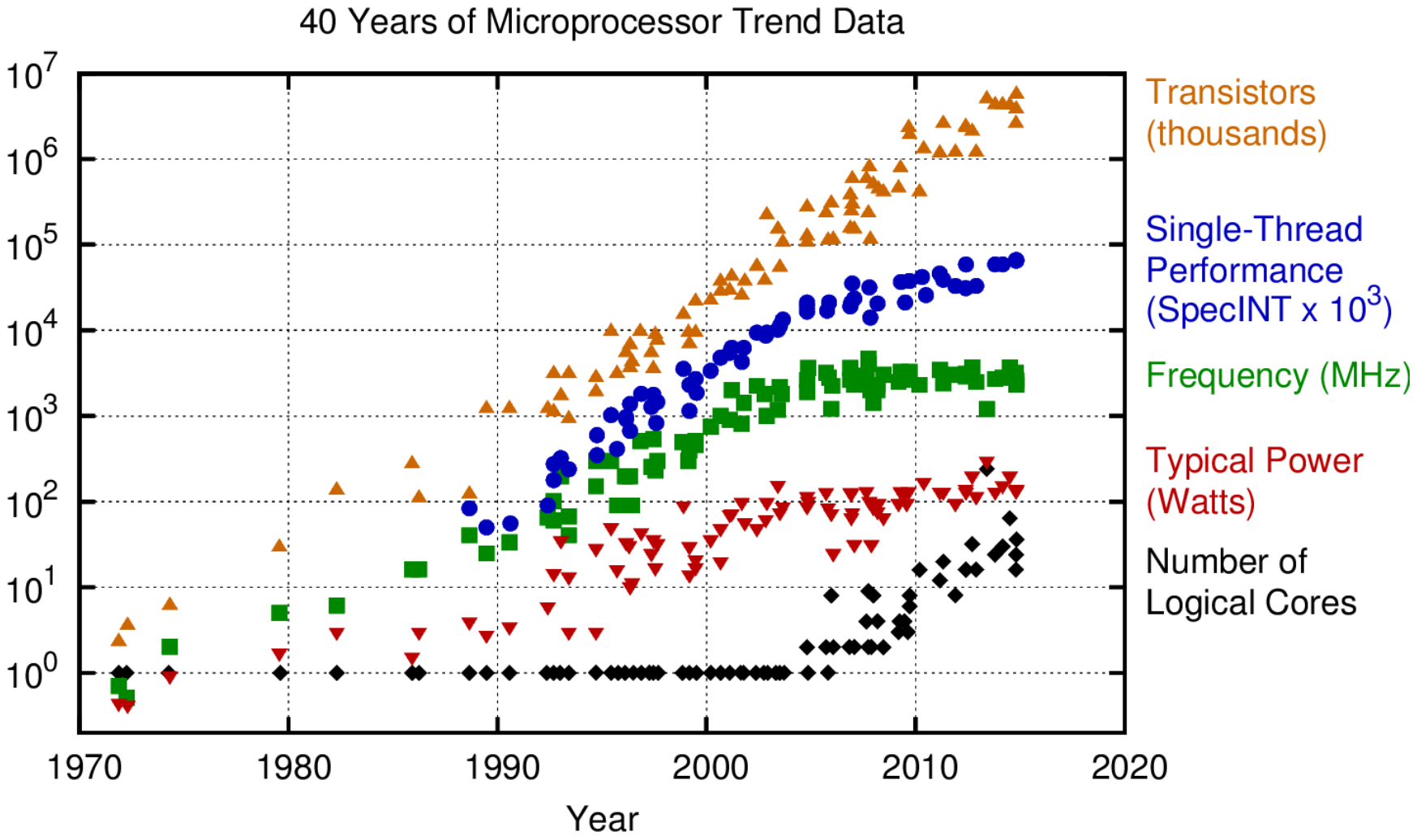


FIGURE 1.17 Growth in processor performance since the mid-1980s. This chart plots performance relative to the VAX 11/780 as measured by the SPECint benchmarks (see Section 1.10). Prior to the mid-1980s, processor performance growth was largely technology-driven and averaged about 25% per year. The increase 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 higher 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 22% 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

A mudança de uniprocessadores para multiprocessadores

- 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.

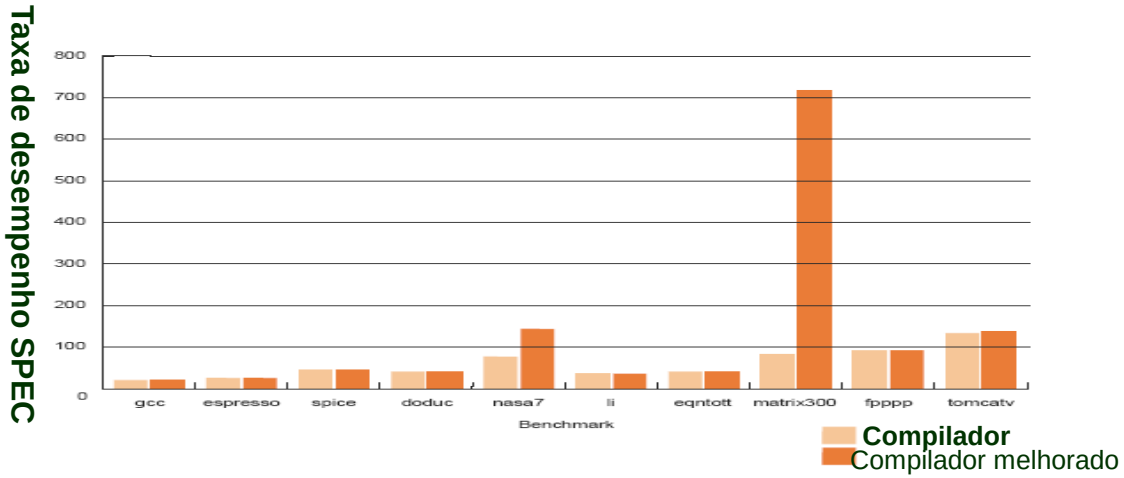
Benchmarks

- 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 P1	1	10	20
Programa P2	1000	100	20

A é 10 vezes mais rápido que B para o programa P1
B é 10 vezes mais rápido que A para o programa P2
A é 20 vezes mais rápido que C para o programa P1
C é 50 vezes mais rápido que A para o programa P2
B é 2 vezes mais rápido que C para o programa P1
C é 5 vezes mais rápido que B para o programa P2

Como resumir?

	Computador A	Computador B	Computador C
Programa P1	1	10	20
Programa P2	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 P1 e P2

C é 25 vezes mais rápido que A para os programas P1 e P2

C é 2.75 vezes mais rápido que B para os programas P1 e P2

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 P1 for muito mais frequentemente executado na aplicação real que o programa P2?

Como resumir?

- 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)
P1	1	10	20	0.5	0.909	0.999
P2	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

$$\overline{t_i} = \frac{tempo\_A_i}{tempo\_Base_i}$$

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 (usada no SPEC)

$$Razão\_Média = \sqrt[N]{\prod_{i=1}^{N} \overline{t_i}}$$

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
P1	1.0	10.0	20.0	0.1	1.0	2.0	0.05	0.5	1.0
P2	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)

Outros:

HINT
Fhourstones
Khornerstone
Aquamark
GL Excess
John the Ripper
The BRL-CAD Benchmark
…

Open source benchmarks

Dhrystone: integer arithmetic performance
Whetstone: floating-point arithmetic performance
} Sintéticos

ApFloat: floating point
Linpack / LAPACK
GliBench: a Gui based benchmarking tool to check CPU and hard disk performance.
BYTEmark benchmark suite
STREAM: memory bandwidth benchmark
MemPerf: memory bandwidth
LLCBench: a group of benchmark for cache, MPI,etc.
LMbench: a suite of simple, portable benchmarks
nbench for Linux/Unix : Memory, integer and floating point comparison with AMD K6 233MHz
Ubench - Unix Benchmark Utility
NAS parallel benchmarks
PovRay: 3D render
Iozone file I/O
Bonnie++: File I/O
netperf : network throughput and latency benchmark
GENESIS distributed memory benchmark suite
Himeno Benchmark
STREAM : measures sustainable memory bandwidth the corresponding computation rate for simple vector kernels.

Microsoft Windows benchmarks

Lavalys EVEREST
SiSoftware Sandra
Futuremark 3DMark
Futuremark PCMark
Futuremark SPMark
BAPCo Mobilemark
BAPCo SYSmark
BAPCo Webmark
REALiX HWiFO32
DocMemory Diagnostic software
CD Speed 99
CPUmark
InfoTool
WinBench 99
Whetstone
PiFast
Super pi (2^{32} dígitos)
Índice de Experiência do Windows

- •Acesso à RAM
- •Cálculo CPU central processing unit
- •Acesso ao HD
- •Elementos Gráficos (Desktop)
- •Gráficos 3D

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**
    - **Network File System**
    - **Power**
    - **Web Servers**
  - Máquina Base:
    - SPEC95: Sun SPARC 10 (até Jul. 2000)
    - SPEC2000: Sun UltraSPARC 5 (até Fev. 2007)
    - SPEC2006: Sun Ultra Enterprise 2
  - Precisa comprar licença para utilizar ☹
  - Resultados são gratuitos ☺

Ex.: SPECINT2006 em um Core i7 920, 2.66GHz

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

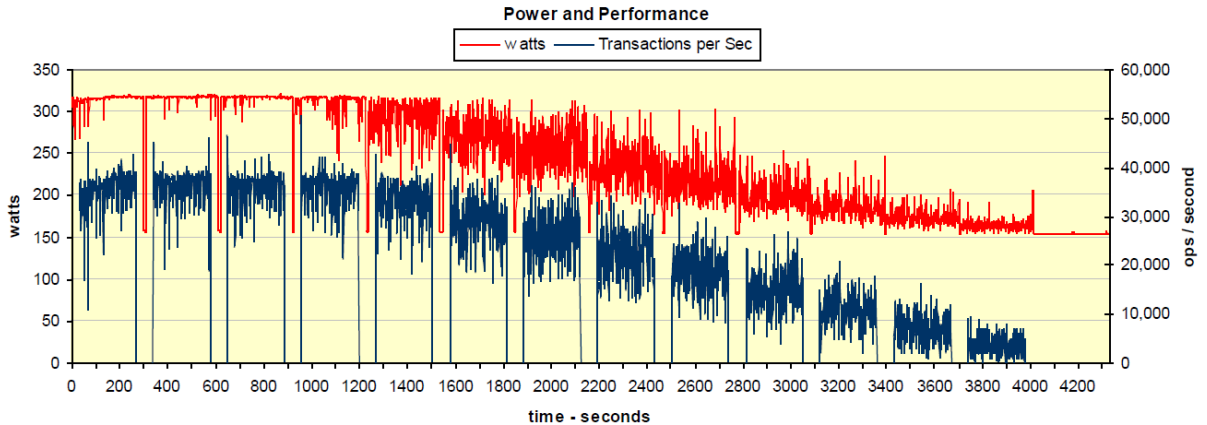
FIGURE 1.18 SPECINTC2006 benchmarks running on a 2.66 GHz Intel Core i7 920. As the equation on page 35 explains, execution time is the product of the three factors in this table: instruction count in billions, clocks per instruction (CPI), and clock cycle time in nanoseconds. SPECratio is simply the reference time, which is supplied by SPEC, divided by the measured execution time. The single number quoted as SPECINTC2006 is the geometric mean of the SPECratios.

Ex.: Power benchmark

Análise realizada, geralmente, com passos de 10% da carga.

Target Load %	Performance (ssj_ops)	Average Power (watts)
100%	865,618	258
90%	786,688	242
80%	698,051	224
70%	607,826	204
60%	521,391	185
50%	436,757	170
40%	345,919	157
30%	262,071	146
20%	176,061	135
10%	86,784	121
0%	0	80
Overall Sum	4,787,166	1922
∑ssj_ops / ∑power =		2490

FIGURE 1.19 SPECpower_ssj2008 running on a dual socket 2.66 GHz Intel Xeon X5650 with 16 GB of DRAM and one 100 GB SSD disk.



Resultados do SPEC

http://www.spec.org

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

Mostrar Sisoft Sandra
http://www.sisoftware.net/

Mostrar página: Tom´s Hardware
http://www.tomshardware.com/

AMD vs Intel

Comparação Técnica (tendenciosa?)

Athlon64DxPentiumD.pdf Athlon64FXxPentiumED840.pdf

Comparação de desempenho (tendenciosa?)

AMD_v_Intel_Performance_Comp_Q2-2006.pdf

Lei de Amdahl

$$Tempo\_execução\_após\_melhoria = Tempo\_de\_execução\_não\_afetado + \frac{Tempo\_de\_execução\_afetado}{Quantidade\_de\_melhoria}$$

- 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?

Exemplos

- 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