

Arquiteturas para Alto Desempenho CES-25

Prof. Paulo André Castro

pauloac@ita.br

Sala 110 – Prédio da Computação

www.comp.ita.br/~pauloac

IEC - ITA

Objetivos Gerais da Matéria

- Aprofundar o conhecimento interno sobre o funcionamento interno dos computadores em detalhes
- Estudar tecnologias aplicadas ao melhoramento do desempenho dos computadores.
- Analisar essas tecnologias do ponto de vista de demanda, eficiência e custo.

Estrutura do Curso – 1 Bim.

Cap. 1 Introdução

- 1.0. (Extra): Breve Histórico da Evolução dos Computadores
- 1.1. Apresentação do curso, Conceitos e Classes de computadores
- 1.2. Construção de Chips
- 1.3. Medidas de Desempenho e Lei de Amdahl

Cap. 2 O Funcionamento do Processador

- Controle do Processador
- Princípios e Exemplos de Conjunto de Instruções,

Cap. 3. Exploração de paralelismo em nível de instrução (Pipeline)

- 3.1. Conceitos e dependências
- 3.2. Caminho dos Dados e Atrasos
- 3.3. Controle do pipeline

Cap. 4 Sistemas de Memória

- 4.1. Conceitos e Funcionamento da Hierarquia de Memória
- 4.2. Memória RAM, Memória entrelaçada e Virtual

Semaninha

Estrutura do Curso – 2 Bim.

Cap. 5 Sistemas de E/S

5.1. Armazenamento

5.2. Barramentos

Cap. 6 Processamento Superescalar

Cap. 7 Processamento Superescalar Especulativo e Predição de Desvios

Cap. 8 Computadores Vetoriais e Sistemas MIMD

Cap. 9 Data Center as a Computer ou Warehouse scale computing

Avaliação

- Por bimestre:
 - 2 provas ou 1 projeto e 1 prova
 - Eventualmente listas de exercício
- Um exame final

- As listas de exercícios são individuais; programas e projetos podem ser em equipe.

Bibliografia Básica

- HENNESSY, J.L.; PATTERSON, D.A. Arquitetura de Computadores: Uma abordagem quantitativa. Tradução da 5ª. Edição. Campus Elsevier, 2014
- Barroso, L. ; Clidaras, J.; Hölzle, U. The Data Center as a Computer. Morgan& Claypool. 2013.
- Patterson,D.A; Hennesy, J.L; Organização e Projeto de Computadores. Tradução da 3a. Edição. Rio de Janeiro. Editora Campus, 2005

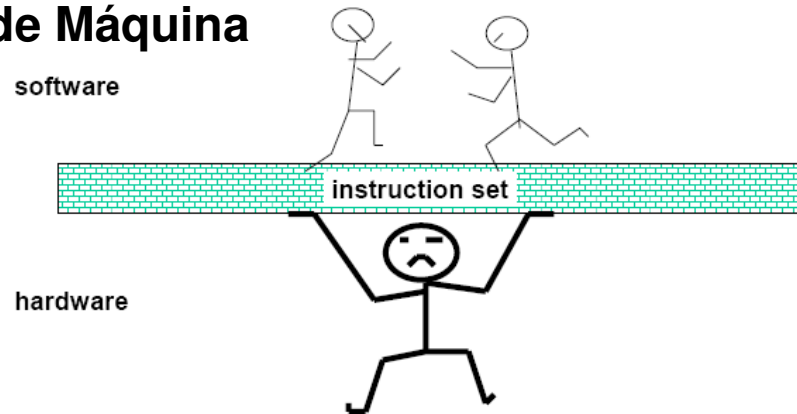
Bibliografia Suplementar

- Tanenbaum, Andrew. *Organização Estruturada de Computadores*. 5a. Edição. Pearson Prentice Hall, 2007.
- STALLINGS, W. *Computer Organization and Architecture*. Upper Saddle River: Prentice Hall, 1996.
- ALMASI, G.S.; GOTTLIEB, A. *Highly Parallel Computing*. Redwood City: Benjamin/Cummings, 1994.
- Messmer, H.P. *The Indispensable PC Hardware Book*. (4a. Ed.), Addison-Wesley Professional; 4 edition. 2001
- STONE, H.S. *High Performance Computer Architecture*. Reading: Addison-Wesley, 1993.
- HAYES, J.P. *Computer Architecture and Organization*. 3.a Edição. Singapore: McGraw Hill, 1998.

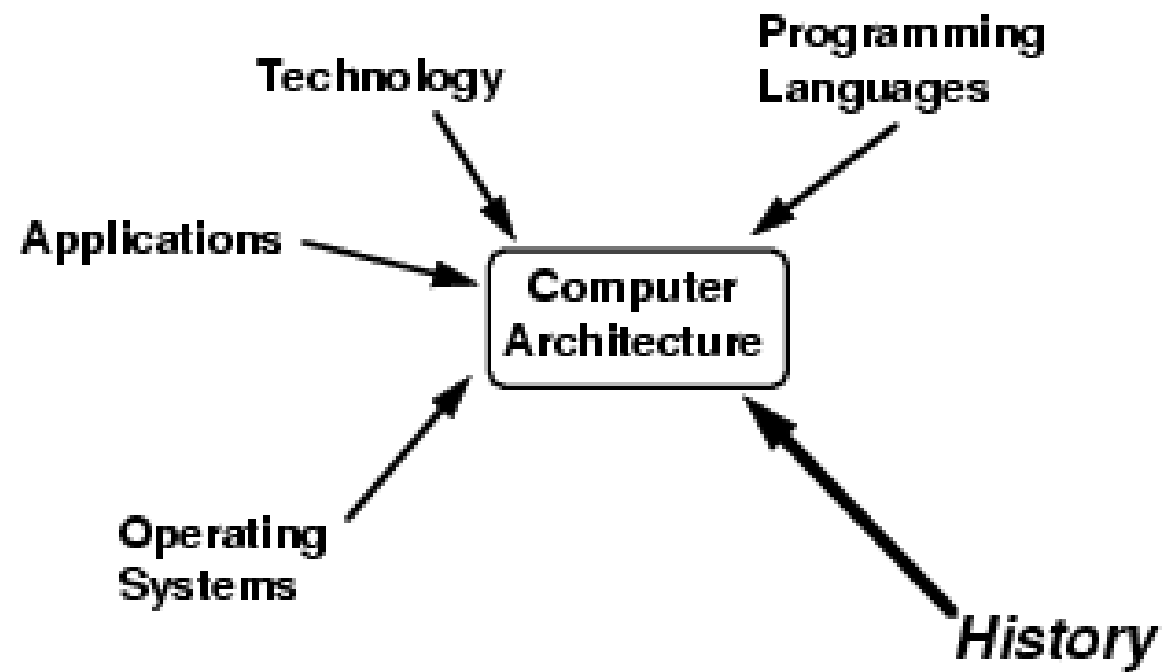
O que é Arquitetura de Computadores ?

Coordenação de um conjunto de níveis de abstração de um computador sob o efeito de um grande conjunto de forças e demandas por mudança

Arquitetura de Computadores = **Arquitetura de Conjuntos de Instruções**
+ **Organização de Máquina**



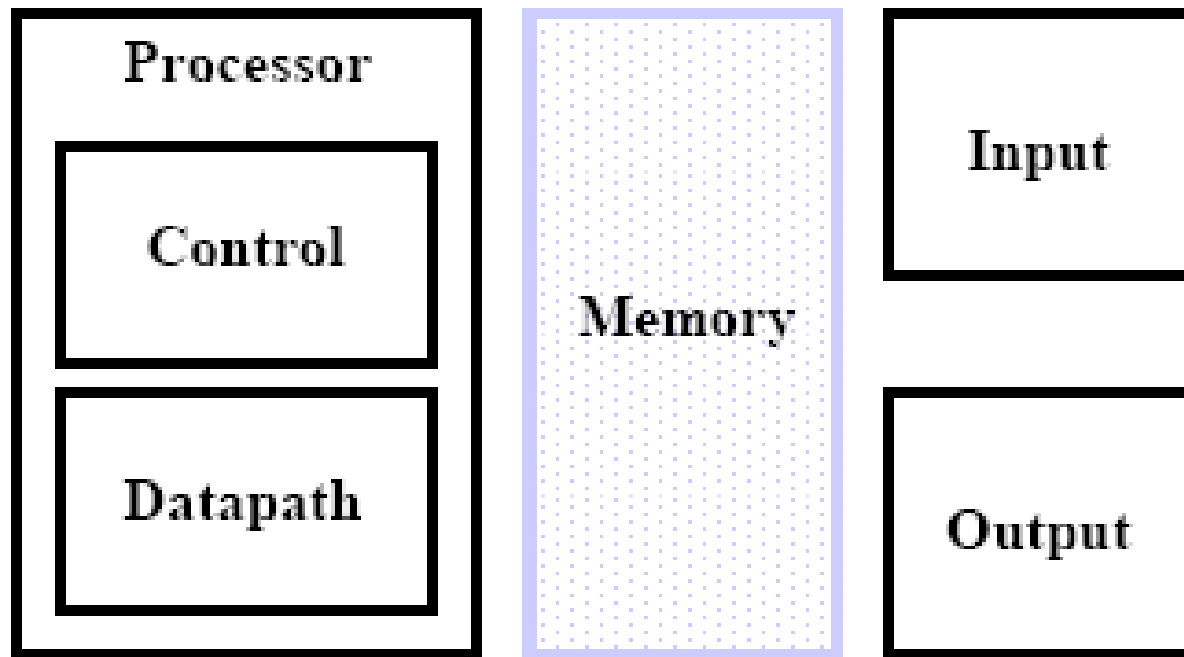
Forças



Demandas Constantes em Computação

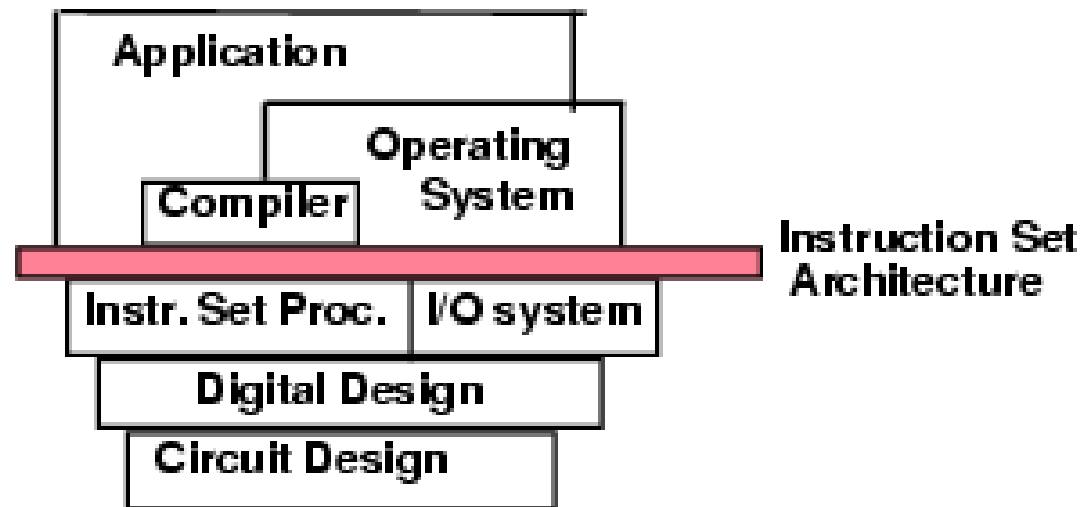
- Demanda por melhor desempenho
 - Software cada vez mais complexo
 - Sistemas gráficos com resolução cada vez maior
 - Aplicações clássicas de Simulação de Sistemas Físicos: Previsão de Tempo, Simulação complexa, Solução de Problemas Matemáticos, etc.
- Demanda por computadores menores e mais leves
- Demanda por maior capacidade de memória e armazenamento
- Demanda por menor consumo de energia (menor dissipação de calor)

Computador em partes: Os cinco componentes clássicos

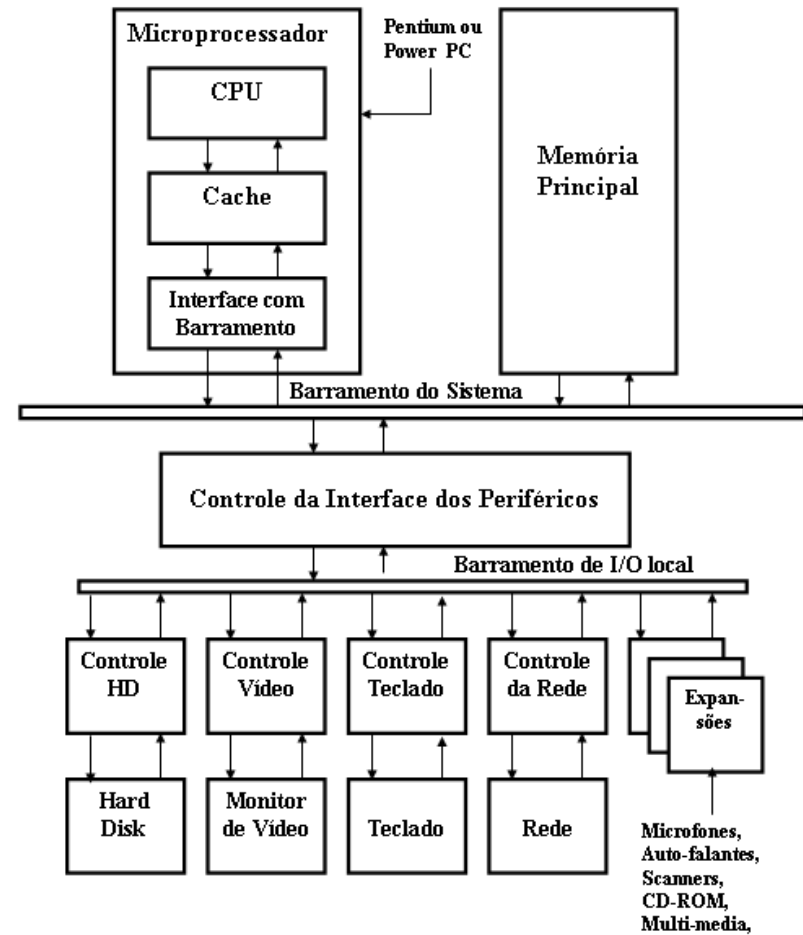
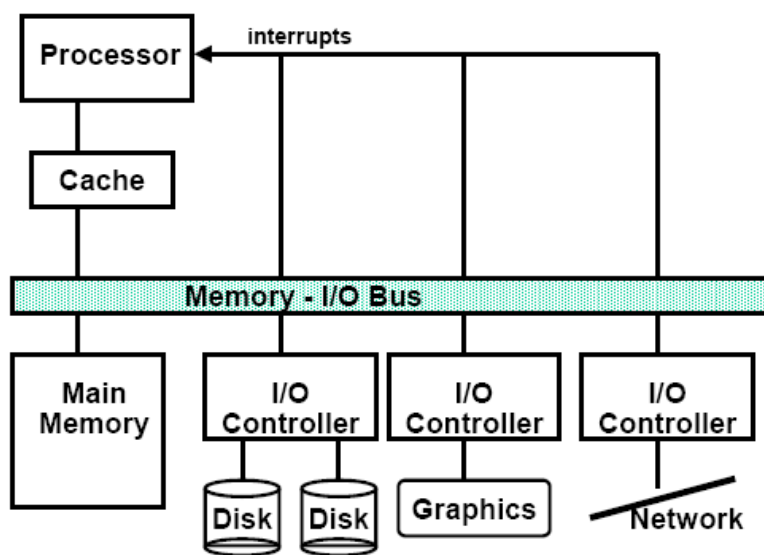


A arquitetura de conjunto de instruções define as instruções do processador e isso guia o projeto do controle e do datapath

Situando Arquitetura de Conjunto de Instruções em Disciplinas



Situando Organização de Máquina



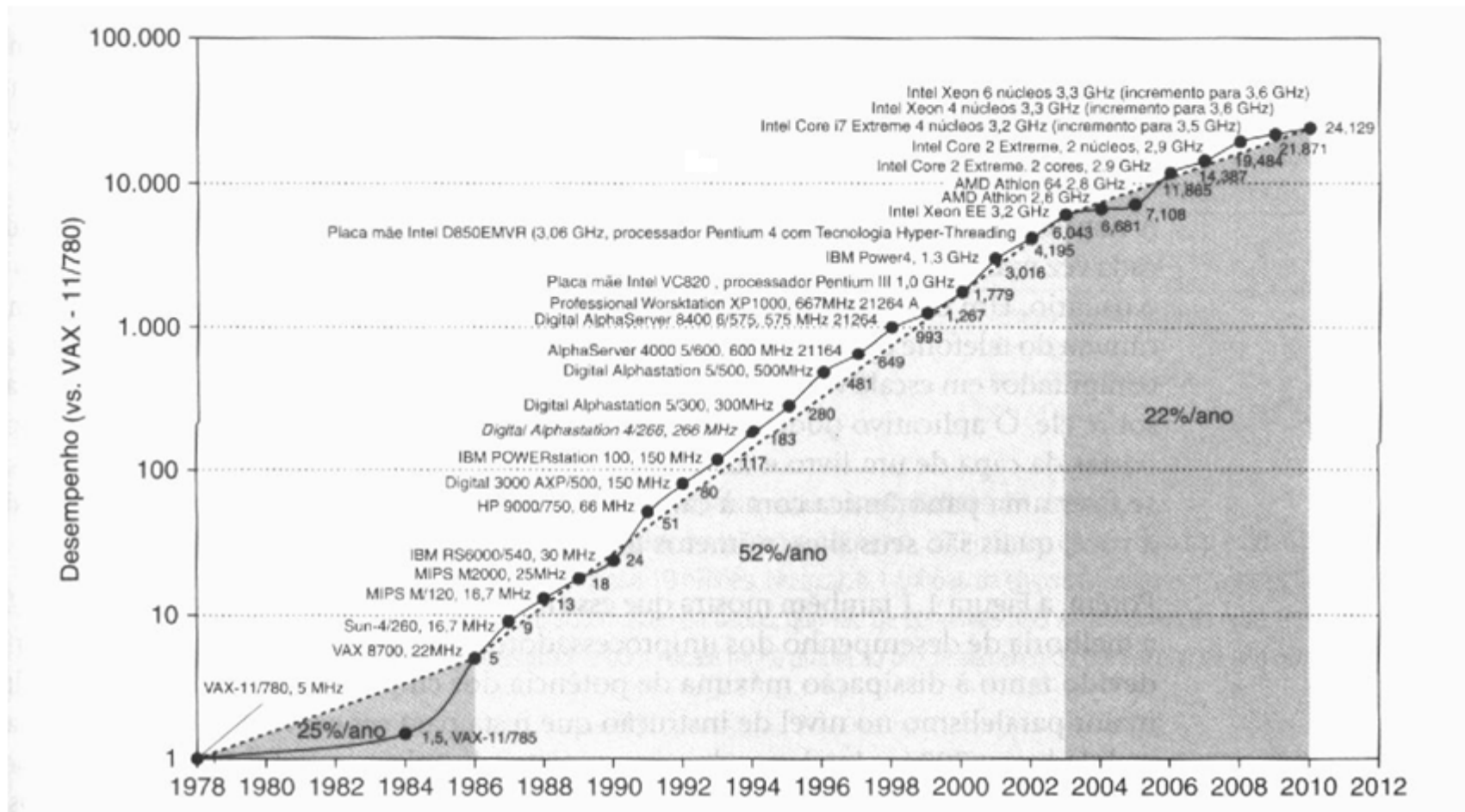
Classes de Computadores e suas Principais Características

Recurso	Dispositivo pessoal móvel (PMD)	Desktop	Servidor	Clusters/computador de armazenamento em escala	Embarcados
Preço do sistema	\$100-\$1.000	\$300-\$2.500	** \$5.000-\$10.000.000	\$100.000-\$200.000.000	\$10-\$100.000
Preço do microprocessador	\$10-\$100	\$50-\$500	\$200-\$2.000	\$50-\$250	\$0,01-\$100
Questões críticas de projeto do sistema	Custo, energia, desempenho de mídia, capacidade de resposta	Preço-desempenho, consumo de energia, desempenho de gráficos	Throughput, disponibilidade, escalabilidade, energia	Preço-desempenho, throughput, proporcionalidade de energia	Preço, consumo de energia, desempenho específico da aplicação

Classe	PMD	Desktop	Servidor	Embedded
Unid. vendidas em 2010	1,8 Bi	350 mi	20 mi	19 Bi (6,1 Bi baseados em ARM, inclui PMD)

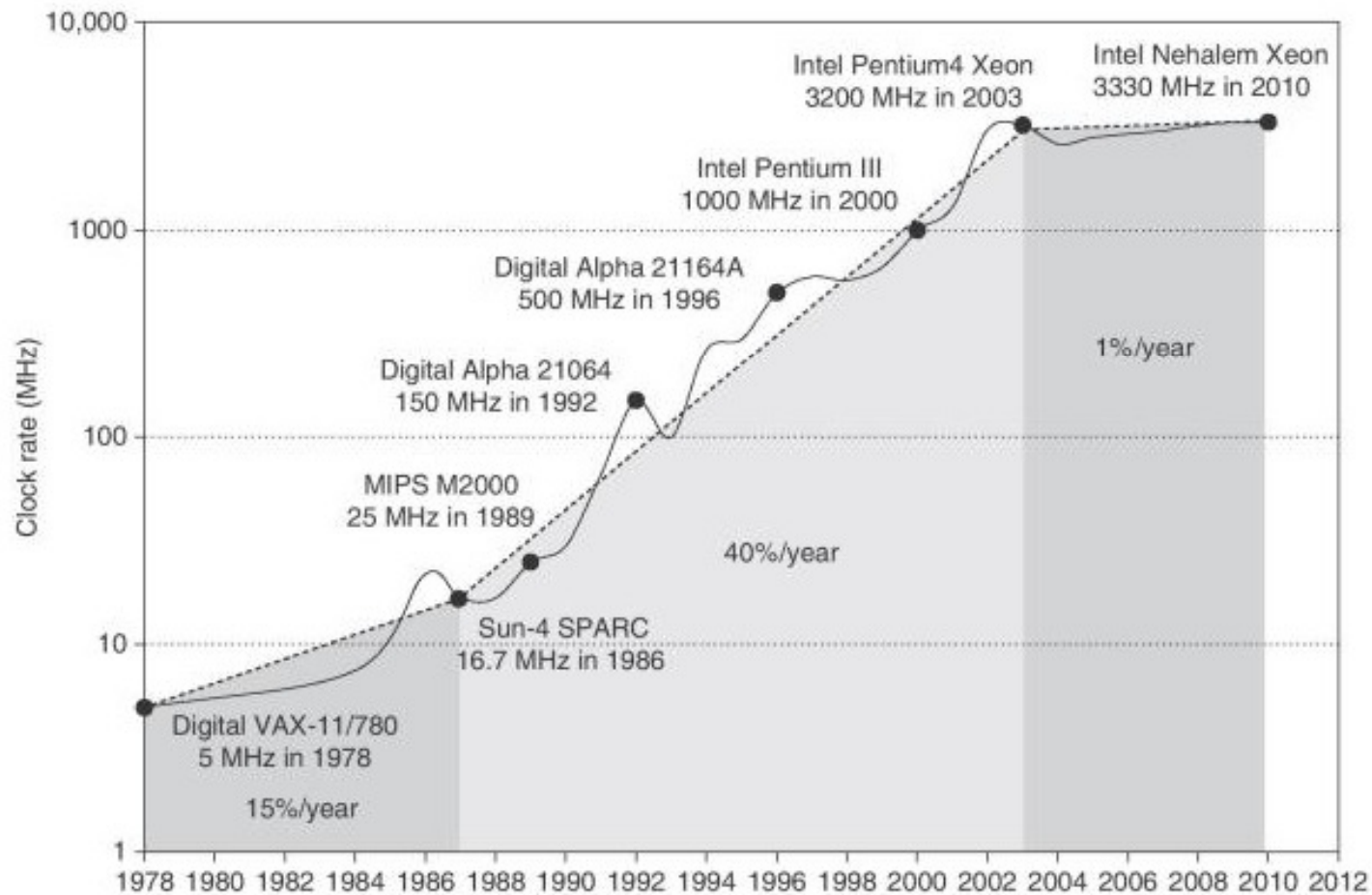
Fonte: (Hennessy, Patterson, 2014)

Evolução do desempenho dos processadores



Fonte: (Hennessy, Patterson, 2014)

Evolução da taxa de clock



Evolução do desempenho da memória

CA:AQA Edition	Year	DRAM growth rate	Characterization of impact on DRAM capacity
1	1990	60%/year	Quadrupling every 3 years
2	1996	60%/year	Quadrupling every 3 years
3	2003	40%–60%/year	Quadrupling every 3 to 4 years
4	2007	40%/year	Doubling every 2 years
5	2011	25%–40%/year	Doubling every 2 to 3 years

Por que estudar Arquitetura de Computadores?

- **Fundamental para o desenvolvimento de sistemas computacionais de alto desempenho (High performance computing) e processamento paralelo**
 - Veremos alguns exemplos de sistemas de Alto Desempenho (cap. 1.2.)
- Para estar apto a fazer decisões de compra de hardware melhor informado sobre as necessidades do software alvo e as características do hardware disponível
- **Para entender a estrutura e o funcionamento dos computadores**
- **Para aprender a utilizar técnicas de otimização de desempenho em sistemas computacionais (software ou hardware)**

Medidas de Desempenho e a Lei de Amdahl

CES-25 – Arquiteturas para Alto Desempenho

Prof. Paulo André Castro

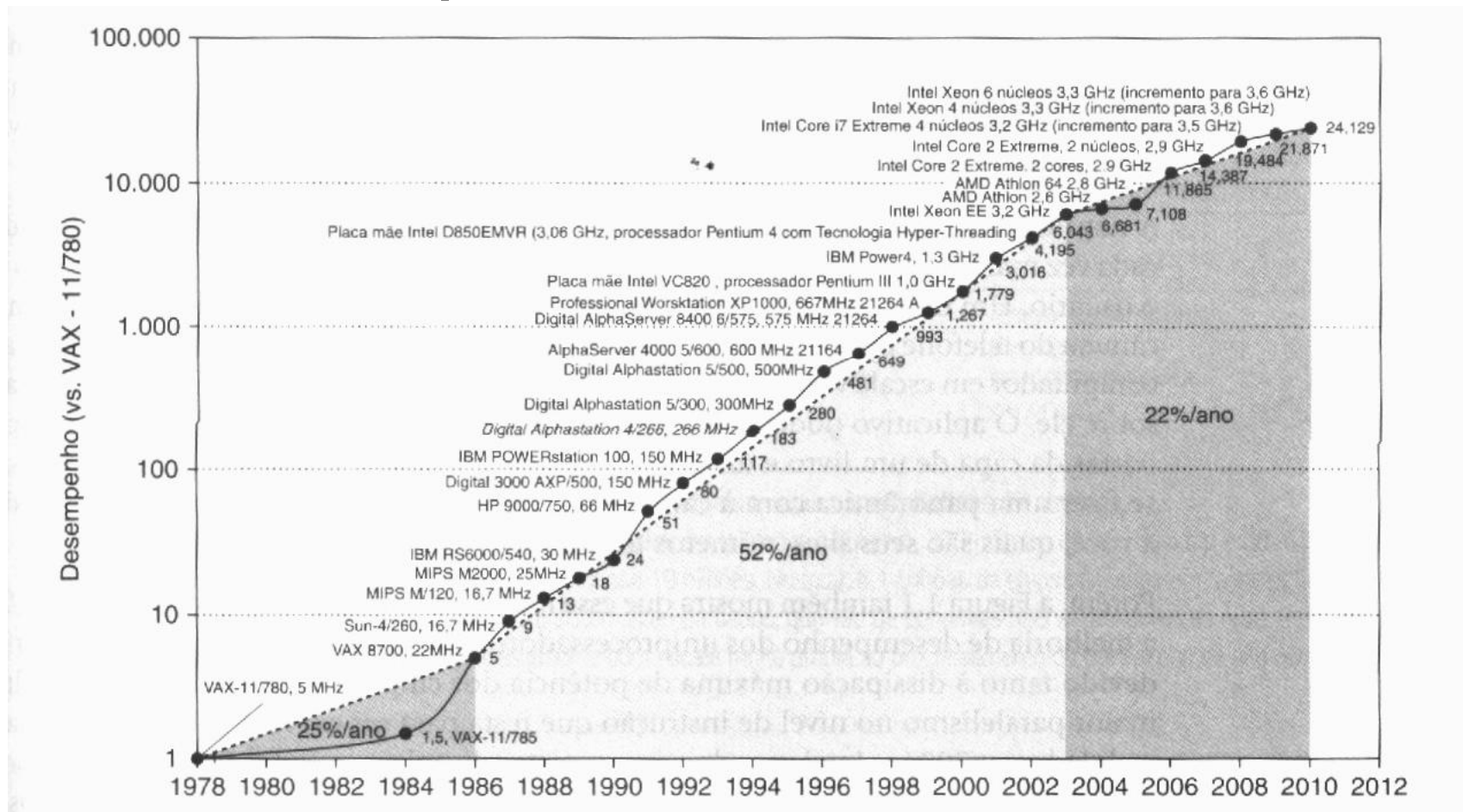
pauloac@ita.br

Sala 110 – Prédio da Computação

www.comp.ita.br/~pauloac

IEC - ITA

Evolução do desempenho dos processadores



Fonte: (Hennessy, Patterson, 2014)

Como medir desempenho?

- Como se mede o impacto de um melhoria no desempenho de um computador?
- Clock? FLOPS?
- Desempenho (performance) deve ser medida através de um conjunto de tarefas executadas por unidade de tempo ?
 - Quanto maior melhor
- Desempenho deve ser medida pelo tempo gasto para realizar uma determinada tarefa ?
 - Quanto menor melhor

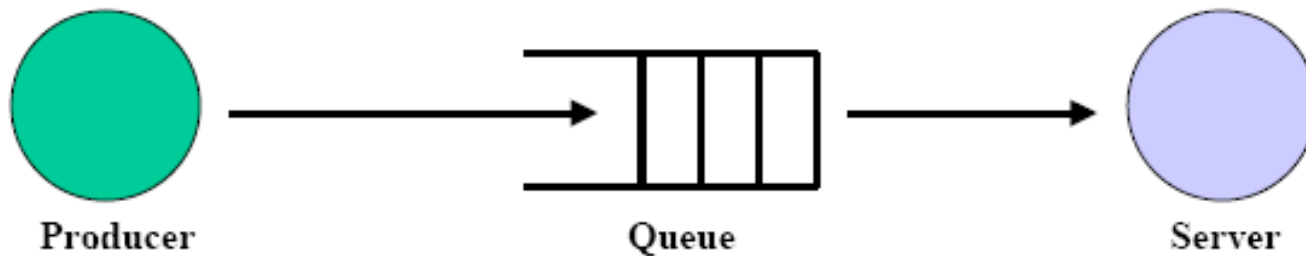
Medidas de Desempenho

- Throughput: Número de tarefas concluídas por unidade de tempo. Por exemplo, instruções/segundo. MFLOPs , Mbps.
- Tempo de resposta (latência): Tempo consumido para executar uma determinada tarefa ou conjunto de tarefas

Perguntas ?

- Aumentar o throughput sempre melhora o tempo de resposta?
- Diminuir o tempo de resposta sempre melhora o throughput?

Modelo Simples Produtor-Consumidor

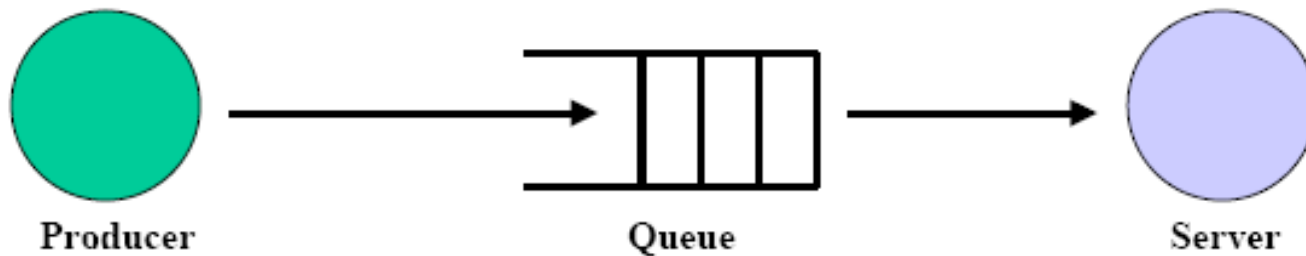


- **Latência** (Tempo de Resposta): tempo médio para a conclusão de uma tarefa.

- Para **minimizar**:

- A fila deveria estar vazia
- O servidor deveria estar ocioso(idle)

Modelo Simples Produtor-Consumidor



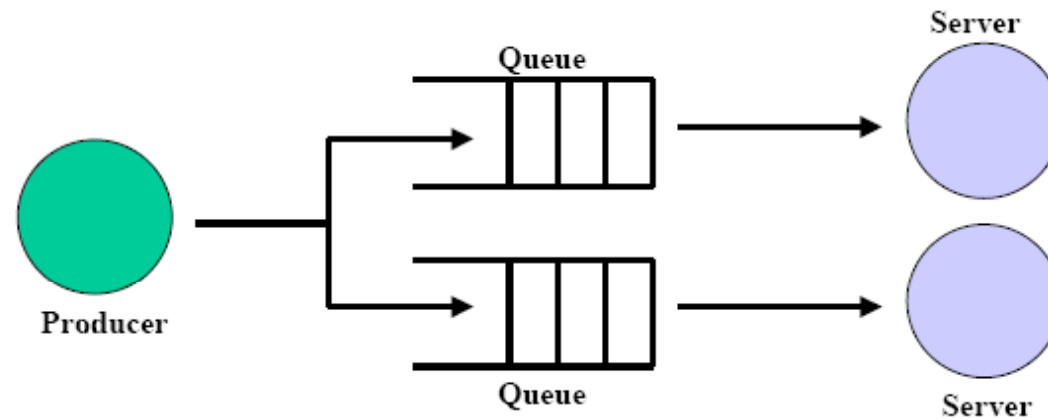
- **Throughput** (“Taxa de Finalização”): Número de tarefas concluídas por unidade de tempo

- Para **maximizar**:

- A fila **nunca** deveria estar vazia
- O servidor **nunca** deveria estar ocioso(idle)

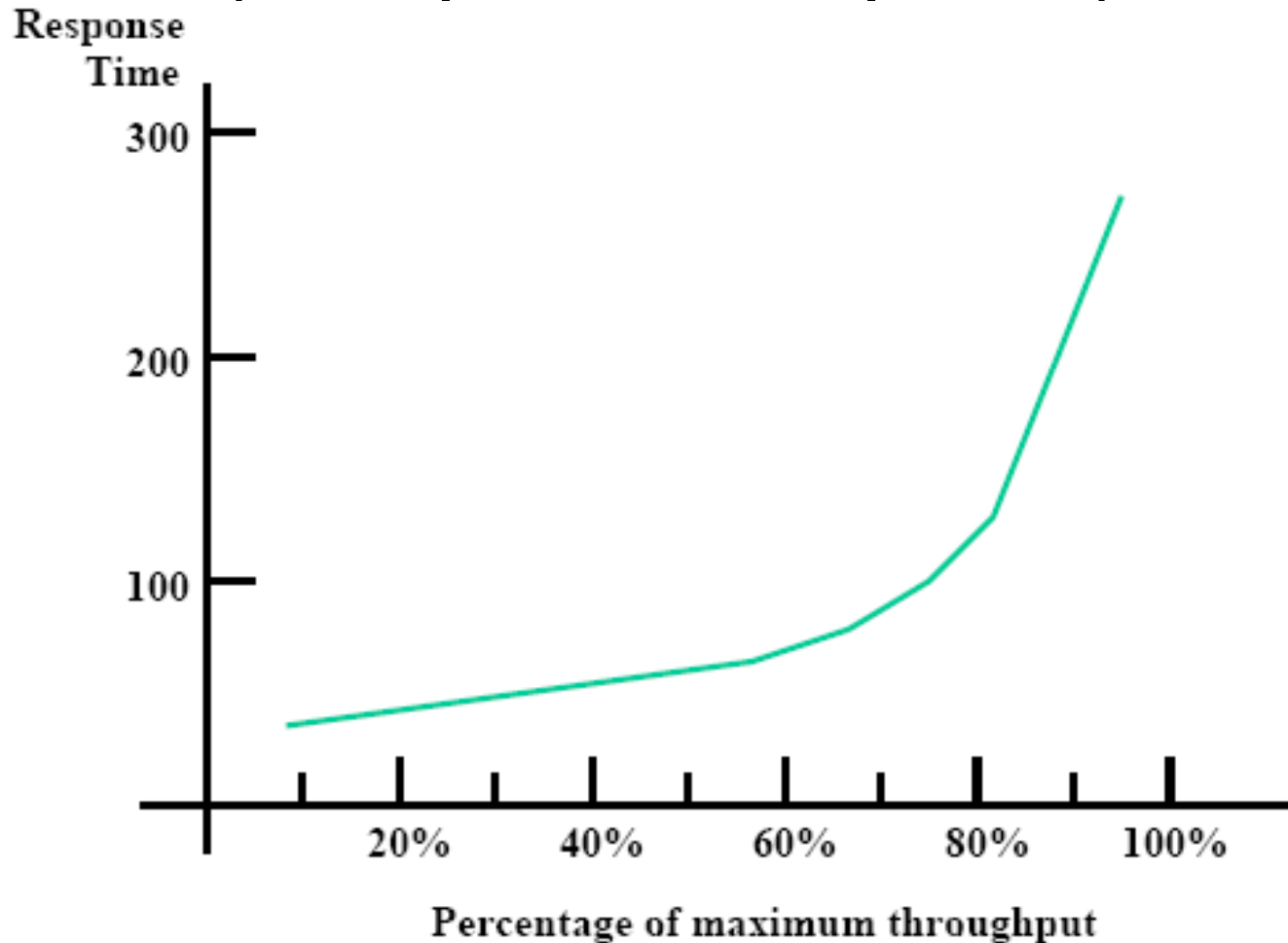
Aumentando o Throughput

Throughput Enhancement



- Em geral, throughput pode ser incrementado
 - Colocando mais hardware (reduzindo latência relacionada a carga)
- Tempo de resposta é muito mais difícil de reduzir
 - É preciso otimizar a arquitetura e/ou tecnologia de implementação

Gráfico qualitativo – Aumento de Throughput X Latência (Tempo de Resposta)



Desempenho

- Estejamos preocupados com throughput ou latência, o mais importante é o tempo total de conclusão das tarefas de interesse do usuário.
 - O computador mais rápido é aquele que atende a seu cliente em menor tempo
- Quando o número de tarefas é muito elevado e resultados parciais relevantes para o cliente podem ser produzidos, é possível “ocultar a latência”.
 - Pense um pouco, o que isso significa ?

Definindo Desempenho de um Sistema em uma “Tarefa”

- Considerando tempo de execução, temos:
 - $\text{Desempenho}(x) = 1/\text{TempoExecução}(X)$
- Logo, dizer que X é n vezes mais rápido que Y, significa:

$$\frac{\text{Desempenho}(x)}{\text{Desempenho}(y)} = n$$

ou,

$$\frac{\text{Tempo_Execução}(Y)}{\text{Tempo_Execução}(X)} = n$$

Medindo o aumento de desempenho

Aumento de desempenho ocorrido, devido a uma melhoria E.

$$\text{speedup}(E) = \frac{\text{ex_time}(\quad)}{\text{ex_time}(E)} = \frac{\text{performance}(E)}{\text{performance}(\quad)}$$

Existem várias técnicas, componentes e meios de comunicação que podem ser alvo de aperfeiçoamentos. Por outro lado, existem também vários “tipos” de instruções quais devem ser melhoradas preferencialmente.

O que deve ser priorizado para possível melhoria?

Caso comum e a Lei de Amdahl

Lei de Amdahl

- O ganho de desempenho possível de um dado melhoramento é limitado pela fração de tempo que a característica melhorada é usada.
- Suponha que um aperfeiçoamento ofereça um ganho de desempenho S para uma fração do tempo F . Qual o ganho de desempenho?

$$ex_time(E) = ((1-F) + F/S) \times ex_time()$$

$$speedup(E) = \frac{1}{(1-F) + F/S}$$

Exemplo

- Se um computador executa um programa P em 100 segundos, onde 30% das operações são acessos a memória com tempo médio de acesso de 80ms e 60% são operações de ponto flutuante.
- A) Qual o impacto no desempenho global do sistema, ao se diminuir o tempo médio de acesso a memória pela metade ?
- B) Qual o impacto no desempenho global, ao dobrar o desempenho das op. de ponto flutuante ?

$$A) \quad \text{speedup}(E) = \frac{1}{(1-F) + F/S}$$

$F = 0,3$

$S = 2$

$\text{SpeedUp}(A) = 1,18$

$S = 2$

$\text{SpeedUp}(B) = 1,43$

Exemplo 2

- Se um computador executa um programa P em 100 segundos, onde 80% do tempo é gasto em operações de memória. Quanto seria preciso melhorar o desempenho da memória para que o sistema tive um ganho de desempenho de cinco vezes?

$$\text{speedup}(E) = \frac{1}{(1-F) + F/S}$$

$$F = 0,8$$

$$S = ?$$

$$\text{SpeedUp}(A) = 5$$

$$S \rightarrow \infty$$

Desempenho

- Todos os componentes influem mas...
- O desempenho da CPU é **crítico** para o desempenho do Computador.
 - desempenhoCPU(X) = 1/TempoCPU(X)

$$\text{cpu_time} = \frac{\text{seconds}}{\text{program}} = \frac{\text{instructions}}{\text{program}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{seconds}}{\text{cycle}}$$

O que influi no desempenho de CPU

Design level	Instr. count	CPI	Clock period
program	X		
compiler	X	X	
instr. set	X	X	
organization		X	X
technology			X

Como medir o desempenho de um computador para as tarefas de interesse do usuário?

- Desempenho é medido com base em tempos de execução de programas/tarefas. Então, como selecionar?
- Situação ideal: usuário sabe exatamente quais programas utiliza com quais dados de entrada e testa nos computadores seus desempenhos. Factível?

Candidatos a programas de avaliação (benchmark)

- Programas reais
 - Dados de entrada
 - portabilidade
- Aplicações modificadas
 - Uso de scripts para fornecer a interatividade
- Kernels
 - Implementam trechos de código comuns em vários programas. Por exemplo: laços com operação de vetores (Linpac)

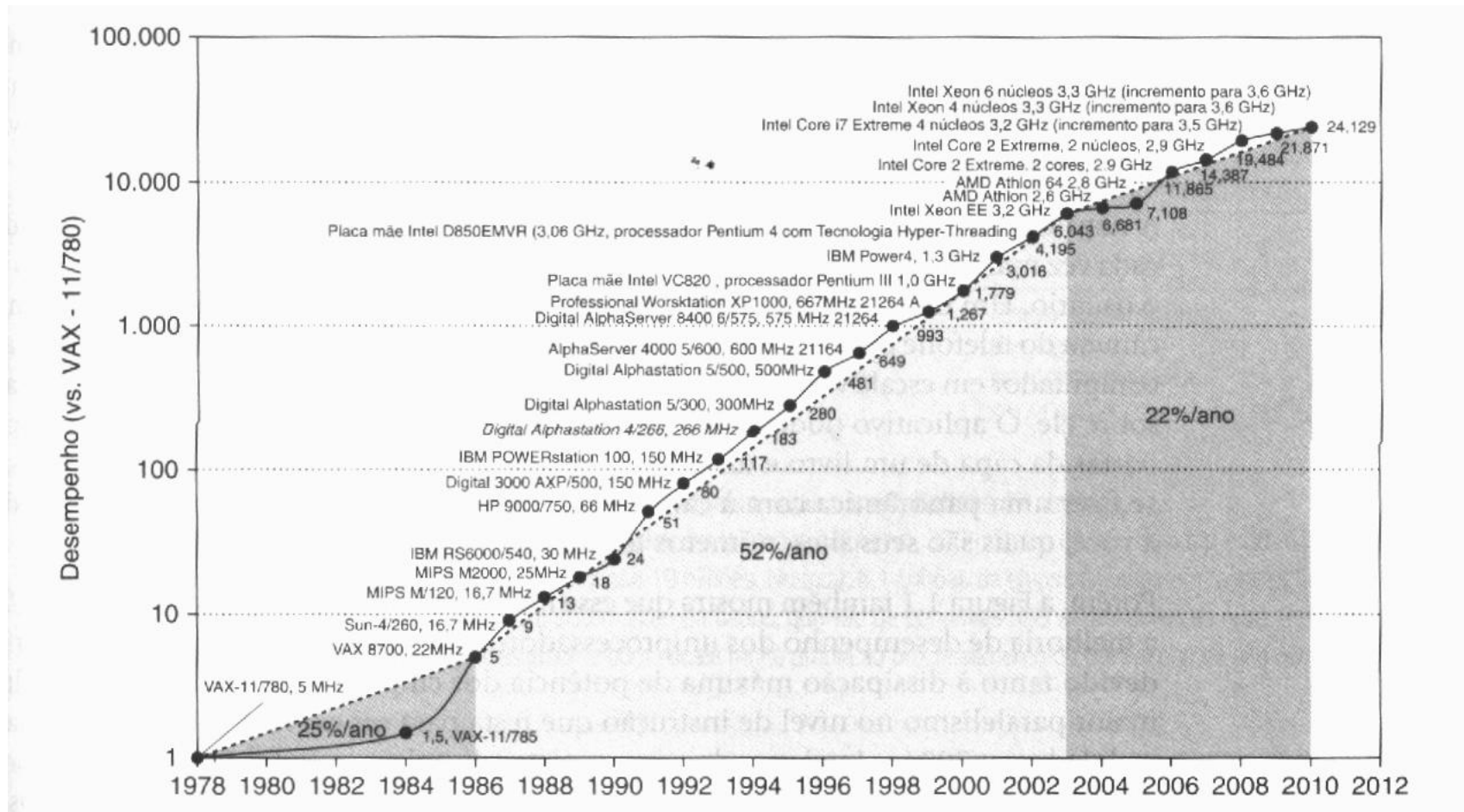
Candidatos a programas de avaliação (benchmark) 2

- Toy Benchmarks:
 - Crivo de Eratóstenes, QuickSort, etc.
- Benchmarks Sintéticos
 - Semelhante aos kernels, porém utilizam frequência média de instruções em grandes programas para selecionar as instruções.
 - Ex. Whetstone e Dhrystone.
- Pacotes de programas de benchmark
 - Reunem vários outros benchmarks individuais para formar um pacote
 - Ex. SPEC(Standard Performance Evaluation Corporation)

Programas de benchmark

- Há muitos programas de benchmark disponíveis.
 - Não há melhores ou piores,
 - há aqueles mais ou menos adequados ao seu problema
- Embora se utilize freqüentemente dados de um benchmark para avaliar de forma absoluta dois computadores, lembre-se:
 - Desempenho é relativo ao programa!!

Olhe de novo para o gráfico abaixo.



- Esse gráfico foi construído com uso do SPECint benchmark. Entretanto, esse benchmark mudou ao longo dos anos. Por tal razão o desempenho dos computadores mais novos é estimado para diferentes versões do SPEC (SPEC92, SPEC95, SPEC2000)

Exemplos de benchmarks
Família SPEC

www.spec.org

SPEC2006 benchmark description	SPEC2006	SPEC2000	SPEC95	SPEC92	SPEC89
GNU C compiler					gcc
Interpreted string processing			perl		espresso
Combinatorial optimization		mcf			li
Block-sorting compression		bzip2		compress	eqrtott
Go game (AI)	go	vortex	go	sc	
Video compression	h264avc	gzip	ijpeg		
Games/path finding	astar	eon	m88ksim		
Search gene sequence	hmmer	twolf			
Quantum computer simulation	libquantum	vortex			
Discrete event simulation library	ornetpp	vpr			
Chess game (AI)	sjeng	crafty			
XML parsing	xalancbmk	parser			
CFD/blast waves	bwaves				fpppp
Numerical relativity	cactusADM				tomcatv
Finite element code	calculix				doduc
Differential equation solver framework	deall				nasa7
Quantum chemistry	garness				spice
EM solver (freq/time domain)	GemsFDTD			swim	matrix300
Scalable molecular dynamics (~NAMD)	gromacs		apsi	hydro2d	
Lattice Boltzman method (fluid/air flow)	lbm		mgrid	su2cor	
Large eddie simulation/turbulent CFD	LESlie3d	wupwise	applu	wave5	
Lattice quantum chromodynamics	milc	aply	turb3d		
Molecular dynamics	namd	galgel			
Image ray tracing	povray	mesa			
Sparse linear algebra	soplex	art			
Speech recognition	sphinx3	equake			
Quantum chemistry/object oriented	tonto	facerec			
Weather research and forecasting	wrf	ampp			

integer programs

floating-point programs

Riscos do uso de Benchmarks

- Como muitas decisões de compra são feitas com base em benchmark, os projetistas são pressionados a otimizar seus projetos para o benchmark
- Isso é realmente bom?