DCC006 – Organização de Computadores I

Aula 2 – Desempenho

Prof. Omar Paranaiba Vilela Neto

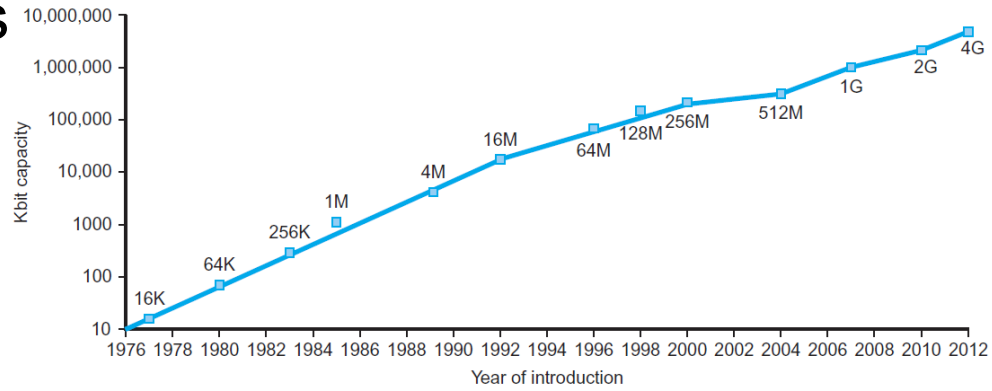
DCC

DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Evolução Tecnológica

Tecnologia de eletrônicos continua evoluindo

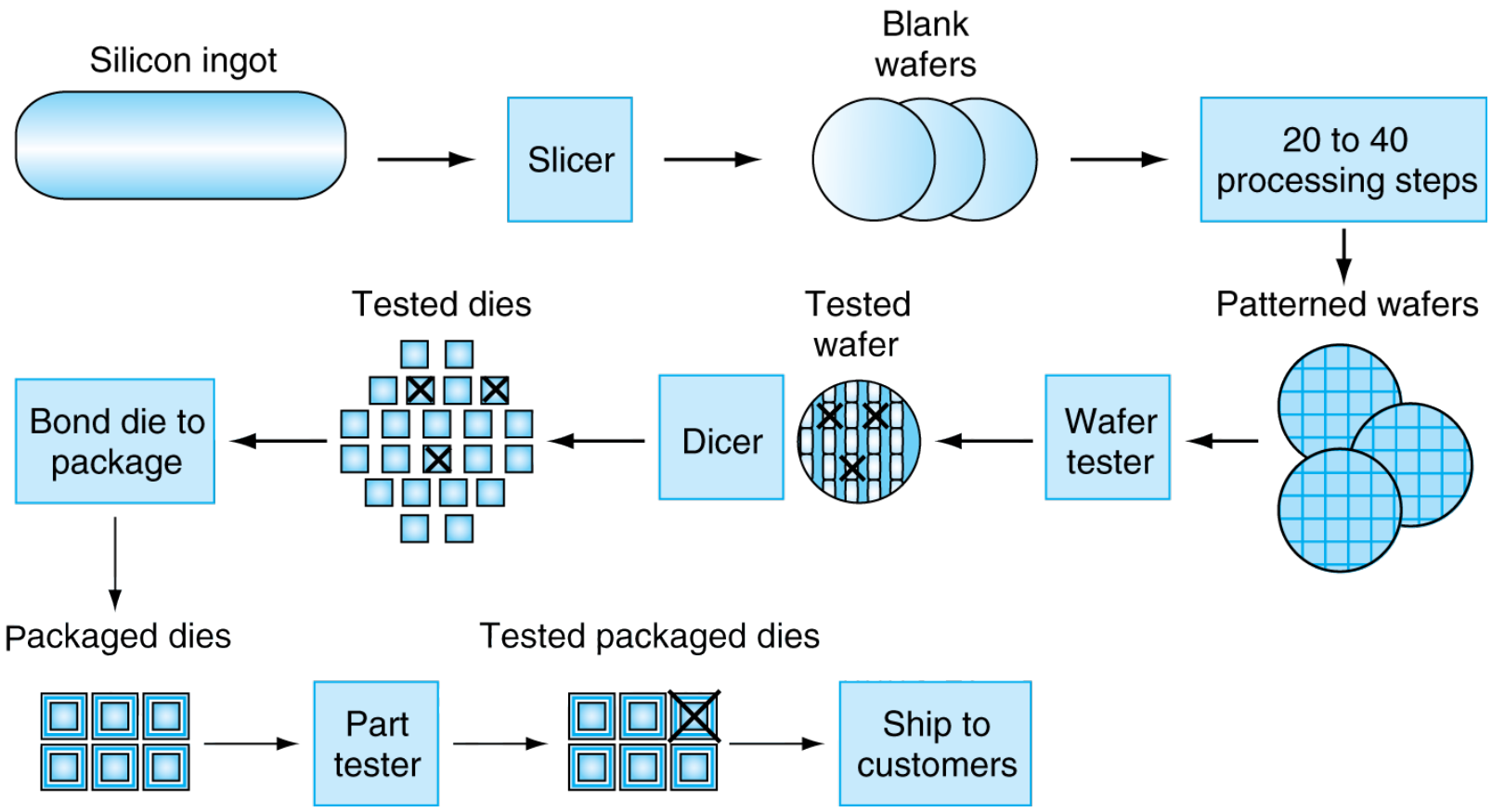
- Aumenta capacidade e desempenho
- Diminui o custo



DRAM capacity

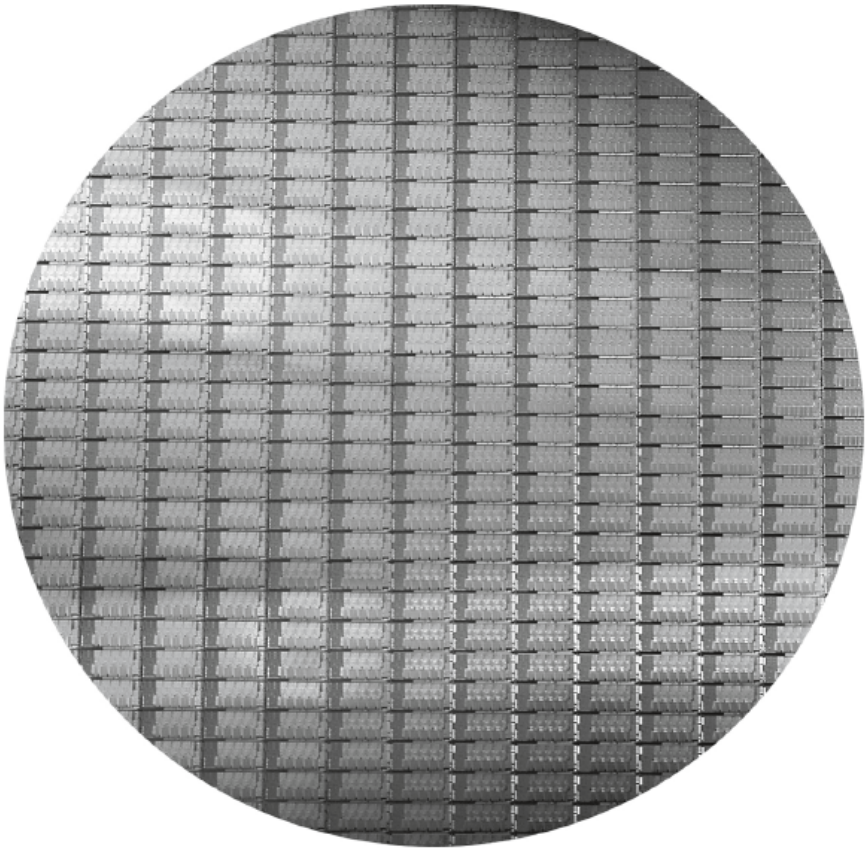
Year	Technology	Relative performance/cost
1951	Vacuum tube	1
1965	Transistor	35
1975	Integrated circuit (IC)	900
1995	Very large scale IC (VLSI)	2,400,000
2013	Ultra large scale IC	250,000,000,000

Produção de Circuitos Integrados



Produção de Circuitos Integrados

Intel
Core i7



300mm wafer, 280 chips, 32nm tecnologia
Cada chip tem tamanho 20.7 x 10.5 mm

Desempenho

Por que nos preocupamos com o desempenho dos computadores?

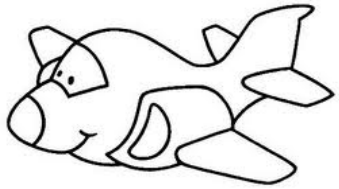
- **Medidas, Relatórios e Resumos**
- **Escolhas inteligentes**
- **Marketing**
- **Organização**

Porque um hardware é melhor que outros para diferentes programas?

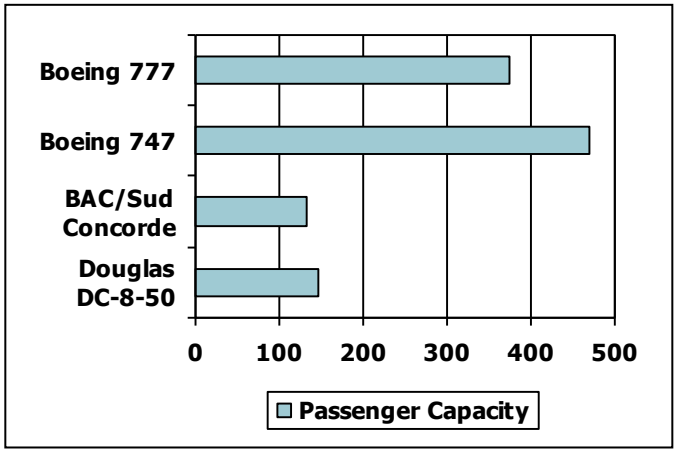
Quais fatores do desempenho são relacionados com o hardware?
(i.e., Nós necessitamos de uma nova máquina ou um novo sistema operacional?)

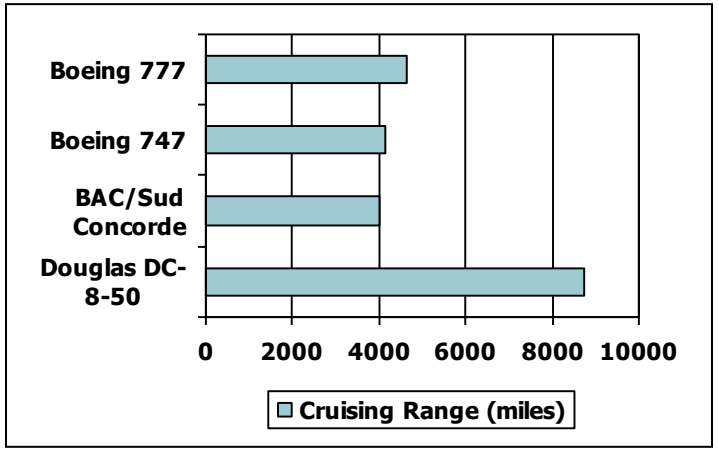
Como um conjunto de instruções afeta o desempenho?

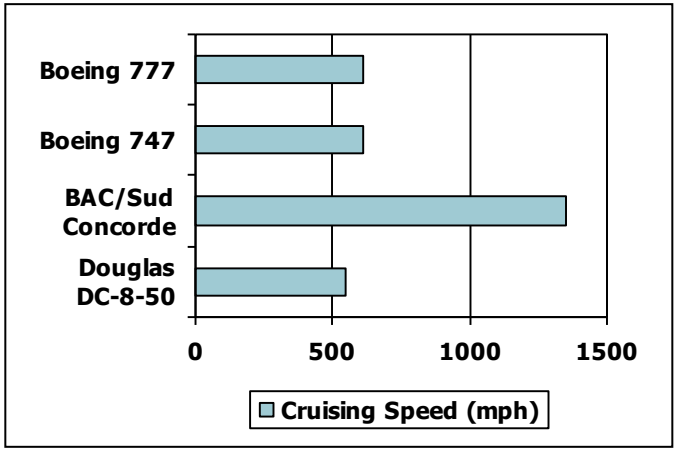
Definindo Desempenho

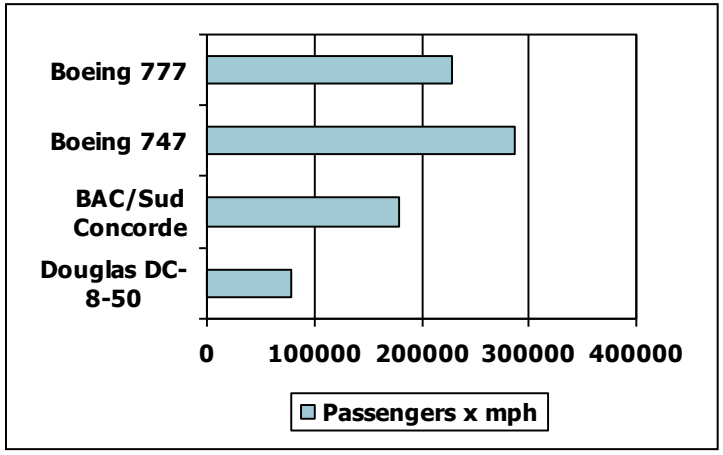


Qual avião tem melhor desempenho?









Desempenho dos computadores: tempo, tempo, tempo ...

- **Tempo de Resposta (latency)**
  - **Demora para uma tarefa rodar**
  - **Demora para executar uma tarefa**
  - **Espera de uma consulta a base de dados**

- **Throughput (vazão)**
  - **Quantos jobs a máquina pode rodar cada vez?**
  - **Qual a taxa média de execução?**
  - **Quantos trabalhos são concluídos?**

Se um novo processador é colocado na máquina o que nós melhoramos?
Se uma nova máquina é colocada no laboratório o que melhoramos?

Tempo de Execução

- **Tempo decorrido**
  - **Tempo total para uma tarefa incluindo acesso a disco, memória, I/O, ...**
  - **Freqüentemente não é bom para comparações**

- **Tempo de CPU**
  - **Não considera I/O ou tempo gasto por outros programas**
  - **pode ser dividido em system time, e user time**

- **Nosso foco: Tempo de CPU do Usuário**
  - **tempo despendido para executar a linha de código em nosso programa**

Definições de Desempenho

- Para vários programas rodando em uma máquina X,

- $\text{Desempenho}_X = 1 / \text{Tempo de execução}_X$

- "X é n vezes mais rápido que Y"

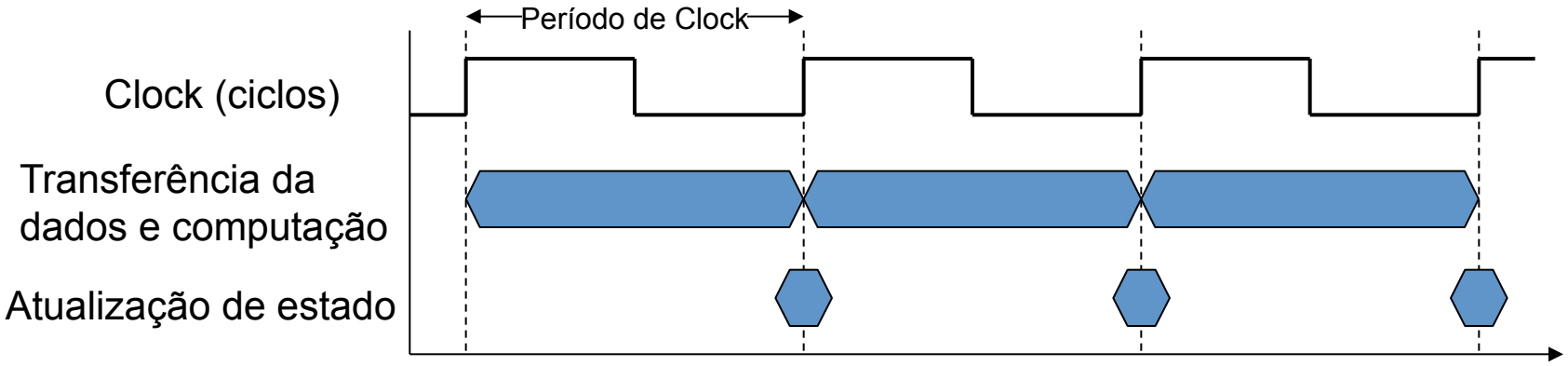
- $\text{Desempenho}_X \ / \ \text{Desempenho}_Y = n$

- Problema:
  - máquina A roda um programa em 20 segundos
  - máquina B roda o mesmo programa em 30 segundos

  - O quanto A é mais rápido que B

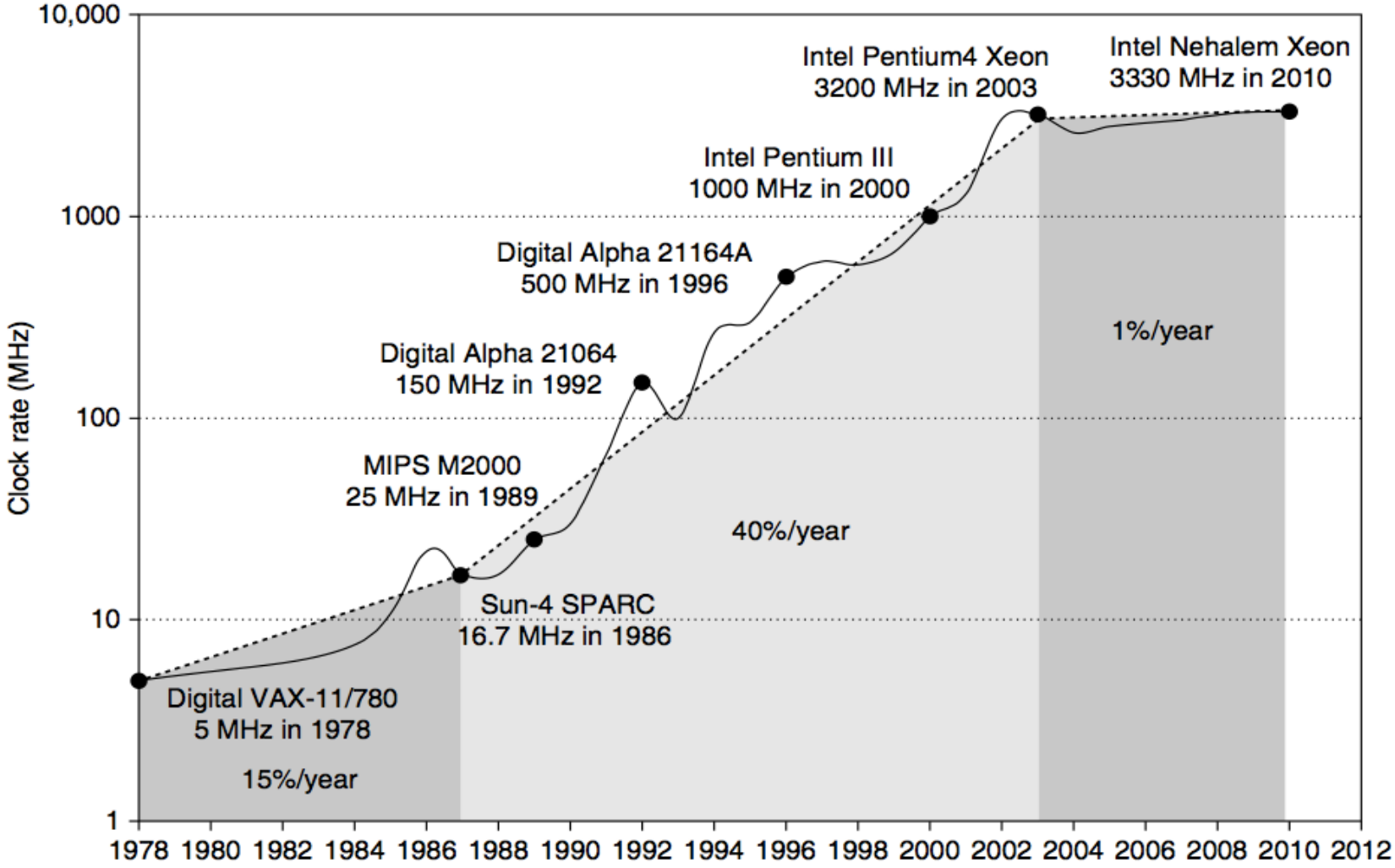
CPU Clock

Operação do hardware digital é governado pela taxa de clock constante



- Período de Clock: duração de um ciclo de clock
  - ex., 250ps = 0.25ns = 250×10^{-12}s
- Frequência de Clock (taxa: ciclos por segundo)
  - ex., 4.0GHz = 4000MHz = 4.0×10^{9}Hz

CPU Clock


10,000
1000
100
10
1
Clock rate (MHz)
1978 1980 1982 1984 1986 1988 1990 1992 1994 1996 1998 2000 2002 2004 2006 2008 2010 2012
Intel Pentium4 Xeon
3200 MHz in 2003
Intel Nehalem Xeon
3330 MHz in 2010
Intel Pentium III
1000 MHz in 2000
Digital Alpha 21164A
500 MHz in 1996
Digital Alpha 21064
150 MHz in 1992
MIPS M2000
25 MHz in 1989
Sun-4 SPARC
16.7 MHz in 1986
Digital VAX-11/780
5 MHz in 1978
15%/year
40%/year
1%/year

Tempo de CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa			
Compilador			
Conj. Instrs.			
Organização			
Tecnologia			

Tempo de CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador			
Conj. Instrs.			
Organização			
Tecnologia			

Tempo de CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.			
Organização			
Tecnologia			

Tempo de CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.	X	X	
Organização			
Tecnologia			

Tempo de CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.	X	X	
Organização	X	X	
Tecnologia			

Tempo de CPU

$$\text{CPU time} = \frac{\text{Seconds}}{\text{Program}} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

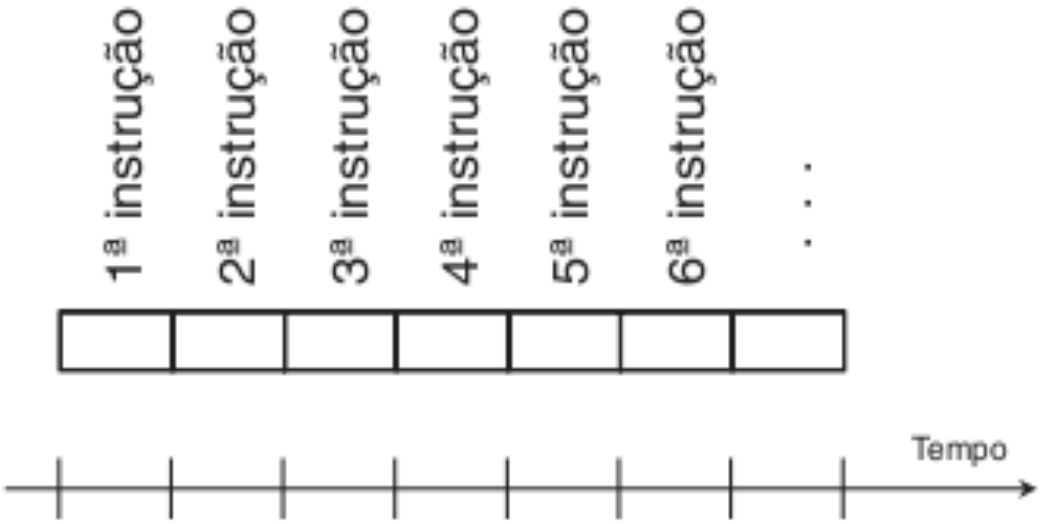
	Instr. Cnt	CPI	Clock Rate
Programa	X		
Compilador	X	X	
Conj. Instrs.	X	X	
Organização	X	X	
Tecnologia			X

Vocabulário

- **Um dado programa requer**
  - **várias instruções (instruções de máquina)**
  - **vários ciclos**
  - **vários segundos**
- **Nós temos um vocabulário que relaciona estas grandezas:**
  - **tempo de ciclo (segundos por ciclo)**
  - **taxa de clock (ciclos por segundo)**
  - **CPI (# médio de ciclos por instrução)**
  - ***uma aplicação intensiva em ponto flutuante poderia ter um alto CPI***
  - **MIPS (milhões de instruções por segundo)**
  - ***poderia ser bom para um programa usando instruções simples***

Ciclos por Instrução

- **Poderíamos assumir que # de ciclos = # de instruções**



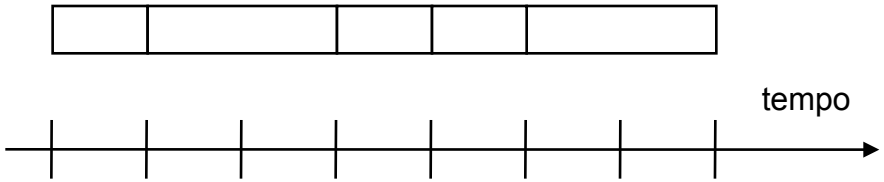
Esta afirmação é incorreta,

instruções gastam tempos diferentes em máquinas diferentes.

Como? lembrem -se que são instruções de máquina e não linhas de código C

Ciclos por Instrução

Diferentes números de ciclos para diferentes instruções



Multiplicação gasta mais tempo que adição

Operações de ponto flutuante são mais longas que inteiros

Acesso a memória gasta mais tempo que acesso a registro

Ponto importante*: *Mudanças no tempo de ciclo freqüentemente muda o número de ciclos requeridos por várias instruções (mais tarde)

Exemplo

•Nosso programa roda em 10 segundos em um computador A, que tem um clock de 2 Ghz. Nós estamos ajudando alguém projetar uma nova máquina B, que deverá rodar este programa em 6 segundos. O projetista poderá usar uma nova (ou mais cara) tecnologia para melhorar substancialmente a taxa de clock, mas ele tem a informação de que este aumento afetará o projeto do resto da CPU e por essa razão a máquina B requer 1,2 vezes mais ciclos de clock que a máquina A para um mesmo programa. Qual a taxa de clock a nova máquina deverá ser projetada?

Exemplo

•Nosso programa roda em 10 segundos em um computador A, que tem um clock de 2 Ghz. Nós estamos ajudando alguém projetar uma nova máquina B, que deverá rodar este programa em 6 segundos. O projetista poderá usar uma nova (ou mais cara) tecnologia para melhorar substancialmente a taxa de clock, mas ele tem a informação de que este aumento afetará o projeto do resto da CPU e por essa razão a máquina B requer 1,2 vezes mais ciclos de clock que a máquina A para um mesmo programa. Qual a taxa de clock a nova máquina deverá ser projetada?

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6\text{s}}$$

Exemplo

•Nosso programa roda em 10 segundos em um computador A, que tem um clock de 2 Ghz. Nós estamos ajudando alguém projetar uma nova máquina B, que deverá rodar este programa em 6 segundos. O projetista poderá usar uma nova (ou mais cara) tecnologia para melhorar substancialmente a taxa de clock, mas ele tem a informação de que este aumento afetará o projeto do resto da CPU e por essa razão a máquina B requer 1,2 vezes mais ciclos de clock que a máquina A para um mesmo programa. Qual a taxa de clock a nova máquina deverá ser projetada?

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6\text{s}}$$

$$\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

Exemplo

•Nosso programa roda em 10 segundos em um computador A, que tem um clock de 2 Ghz. Nós estamos ajudando alguém projetar uma nova máquina B, que deverá rodar este programa em 6 segundos. O projetista poderá usar uma nova (ou mais cara) tecnologia para melhorar substancialmente a taxa de clock, mas ele tem a informação de que este aumento afetará o projeto do resto da CPU e por essa razão a máquina B requer 1,2 vezes mais ciclos de clock que a máquina A para um mesmo programa. Qual a taxa de clock a nova máquina deverá ser projetada?

$$\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6\text{s}}$$

$$\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A$$

$$= 10\text{s} \times 2\text{GHz} = 20 \times 10^9$$

$$\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6\text{s}} = \frac{24 \times 10^9}{6\text{s}} = 4\text{GHz}$$

Definições de Desempenho

- **Desempenho é determinado pelo tempo de execução**
- **Como podemos quantificar o desempenho?**
  - **# de ciclos para executar um programa**
  - **# de instruções no programa**
  - **# de ciclos por segundo**
  - **média # de ciclos por instrução**
  - **Média # de instruções por segundo**

- **Erro comum: indicar uma das variáveis como desempenho quando ela realmente não é.**

CPI Exemplo

- **Suponha que você tem duas implementações de um mesmo conjunto de instruções.**

- **Máquina A tem um tempo de ciclo de clock de 250 ps. e CPI de 2.0**
- **Máquina B tem um tempo de ciclo de clock de 500 ps. e CPI de 1.2**

- **Qual a máquina mais rápida para este programa e quanto?**

CPI Exemplo

- **Suponha que você tem duas implementações de um mesmo conjunto de instruções.**

- **Máquina A tem um tempo de ciclo de clock de 250 ps. e CPI de 2.0**
- **Máquina B tem um tempo de ciclo de clock de 500 ps. e CPI de 1.2**

- **Qual a máquina mais rápida para este programa e quanto?**

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

CPI Exemplo

- **Suponha que você tem duas implementações de um mesmo conjunto de instruções.**

- **Máquina A tem um tempo de ciclo de clock de 250 ps. e CPI de 2.0**
- **Máquina B tem um tempo de ciclo de clock de 500 ps. e CPI de 1.2**

- **Qual a máquina mais rápida para este programa e quanto?**

$$\begin{aligned}\text{CPU Time}_A &= \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A \\ &= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps} \\ \text{CPU Time}_B &= \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B \\ &= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}\end{aligned}$$

CPI Exemplo

- **Suponha que você tem duas implementações de um mesmo conjunto de instruções.**

- **Máquina A tem um tempo de ciclo de clock de 250 ps. e CPI de 2.0**
- **Máquina B tem um tempo de ciclo de clock de 500 ps. e CPI de 1.2**

- **Qual a máquina mais rápida para este programa e quanto?**

$$\begin{aligned}
\text{CPU Time}_A &= \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A \\
&= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps} \\
\text{CPU Time}_B &= \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B \\
&= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps} \\
\frac{\text{CPU Time}_B}{\text{CPU Time}_A} &= \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2
\end{aligned}$$

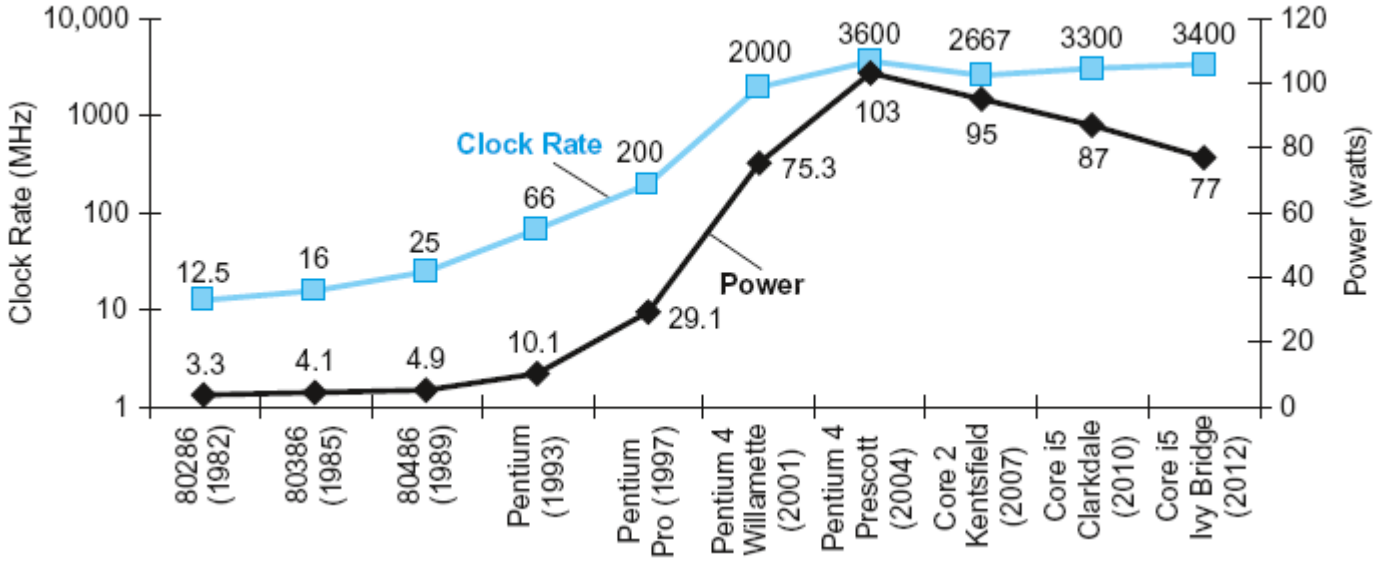
CPI Exemplo 2

2 Alternativas compiladas usando instruções em classes A, B, C

Class	A	B	C
CPI for class	1	2	3
IC in sequence 1	2	1	2
IC in sequence 2	4	1	1

- Sequencia 1: IC = 5
  - Clock Cycles
    = 2×1 + 1×2 + 2×3
    = 10
  - Avg. CPI = 10/5 = 2.0
- Sequencia 2: IC = 6
  - Clock Cycles
    = 4×1 + 1×2 + 1×3
    = 9
  - Avg. CPI = 9/6 = 1.5

Tendências de Potência



In CMOS IC technology

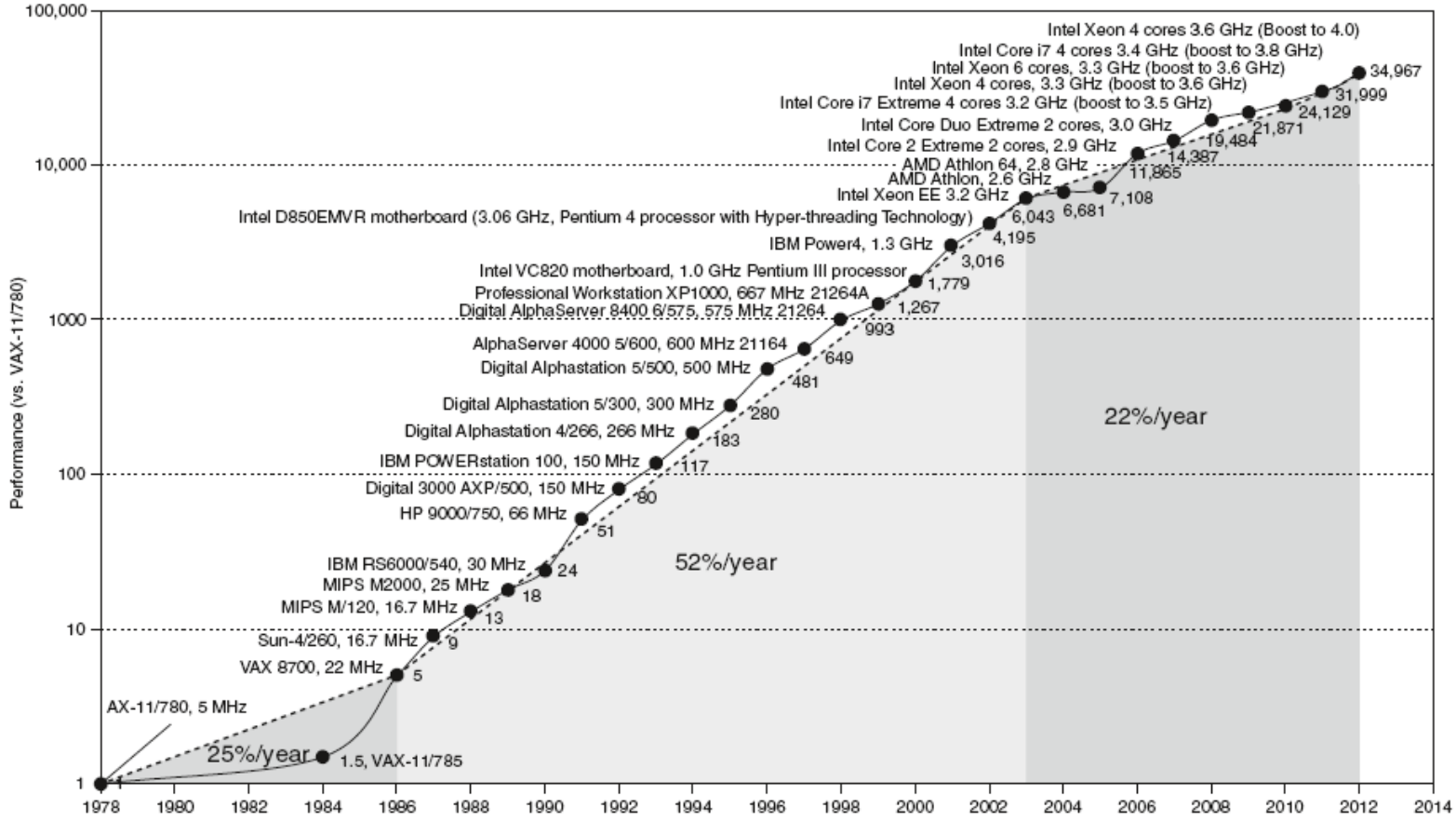
$$\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

×30

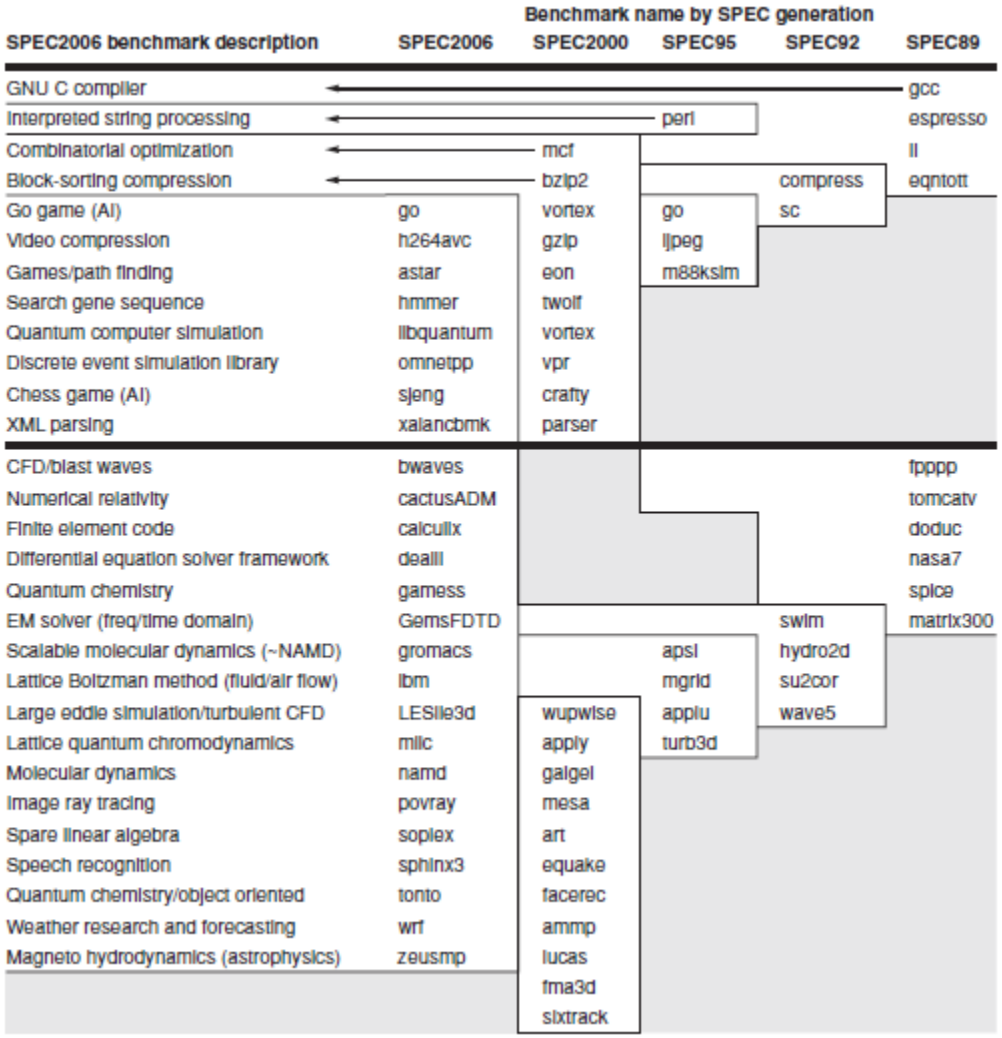
5V → 1V

×1000

Como melhoraram o desempenho?



SPEC



www.spec.org

SPEC

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

Metodologia de Projeto

Como melhorar o desempenho de computadores?

O que devemos priorizar?

Metodologia de Projeto

Princípio Básico

Torne **rápido** o mais **comum** !!!

Favoreça o mais frequente em relação ao caso pouco frequente !!!

Metodologia de Projeto

Lei de Amdahl

Speedup devido à melhoria E:

$$\text{Speedup}(E) = \frac{\text{ExTime sem E}}{\text{ExTime com E}} = \frac{\text{Desempenho com E}}{\text{Desempenho sem E}}$$

Suponha que melhoria E acelere porção F da tarefa por fator S, e que restante da tarefa permanece sem alteração, então

$$\text{ExTime}(E) =$$

$$\text{Speedup}(E) =$$

Metodologia de Projeto

Lei de Amdahl

$$ExTime_{new} = ExTime_{old} \times \left[(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}} \right]$$

$$Speedup_{overall} = \frac{ExTime_{old}}{ExTime_{new}} = \frac{1}{(1 - Fraction_{enhanced}) + \frac{Fraction_{enhanced}}{Speedup_{enhanced}}}$$

Em última análise, desempenho de qualquer sistema será limitada por porção que não é melhorada...