Avaliando e Compreendendo o Desempenho

Capítulo 4

Desempenho

- Mensurar, analisar e informar
- Fazer escolhas inteligentes

Por que um hardware é melhor que outro para programas diferentes?

Que fatores do desempenho do sistema estão relacionados ao hardware?

Como o conjunto de instruções da máquina afeta o desempenho?

Qual desses aviões possui o melhor desempenho?

Airplane	Passengers	Range (mi)	Speed (mph)
Boeing 737-100	101	630	598
Boeing 747	470	4150	610
BAC/Sud Concorde	132	4000	1350
Douglas DC-8-50	146	8720	544

- Quão mais rápido é o Concorde comparado ao 747?
- Quão maior é o 747 comparado ao Douglas DC-8-50?

Desempenho dos SCs

- Tempo de resposta (latência)
  — Quanto tempo leva para minha tarefa executar?
  — Quanto tempo devo esperar por uma consulta ao banco de dados?
- *Throughput* (vazão)
  — Quantas tarefas a máquina consegue executar simultaneamente?
  — Qual é a taxa de execução média?
  — Quanto do trabalho total já foi concluído?
- *Se a máquina é atualizada com um novo processador, o que é melhorado (latência ou throughput)?*
- *Se nós adicionamos uma nova máquina ao laboratório, o que nós aumentamos (latência ou throughput)?*

Tempo de execução

- Tempo decorrido
  - Considera tudo *(acessos ao disco e memória, I/O, etc.)*
  - Uma medida importante mas, não muito boa para propósitos de comparação

- Tempo de CPU
  - Não conta I/O ou tempo gasto executando outros programas
  - Pode ser dividido em: tempo do sistema e tempo do usuário

- Nosso foco: tempo do usuário
  - Tempo gasto executando as linhas de código do nosso programa

Definição de desempenho

- Para um programa executando na máquina X, o desempenho é dado por:

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

- Para mostrar que "X é n vezes mais rápido que Y"

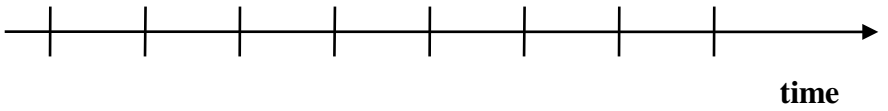
$$\text{Desempenho}_X \ / \ \text{Desempenho}_Y = n$$

- Problema:
  - Máquina A executa um programa em 20 seg.
  - Máquina B executa o mesmo programa em 25 seg.

Ciclos de clock

- Ao invés de informar o tempo de execução em segundos, utiliza-se ciclos

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



- “Barras” indicam quando a atividade pode iniciar
- Tempo de ciclo = tempo entre barras = segundos por ciclo
- Taxa de clock (freqüência) = ciclos por segundo (1 Hz = 1 cycle/sec)
  Um clock de 4 Ghz, possui tempo de ciclo igual a

$$\frac{1}{4\times10^{9}} \times 10^{12} = 250 \text{ picoseconds (ps)}$$

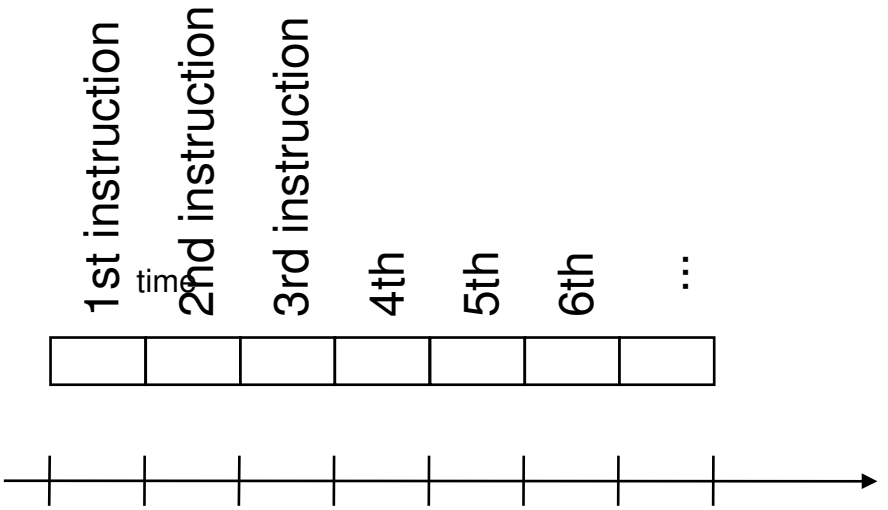
Como pode-se melhorar o desempenho?

Para melhorar o desempenho (todas outras coisas são iguais). Voce pode calcular:

_____ o # de ciclos requeridos para um programa, ou
_____ o tempo de ciclo de clock ou ainda,
_____ a taxa de clock.

Quantos ciclos são requeridos para um programa?

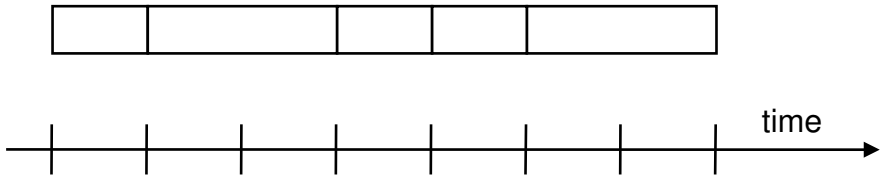
- Pode assumir que número de ciclos é igual ao número de instruções



Assumir isso está incorreto!

Instruções diferentes tomam diferentes quantidades de tempo em diferentes números de instruções

Diferentes números de ciclos para instruções diferentes



- Multiplicação leva mais tempo que adição
- Operações de ponto-flutuante tomam mais tempo que as de inteiros
- Acessar memória toma mais tempo do que acessar registradores

- *Ponto Importante: modificar o tempo de ciclo resulta na modificação do número de ciclos para várias instruções*

Exemplo

- Um programa P executa em 10 segundos no computador A que possui 4GHz.
- A idéia é construir uma máquina B que executa P em 6 segundos.
- O projetista pode utilizar uma tecnologia mais avançada (e talvez mais cara) para aumentar a taxa de *clock*, mas isso afetará o projeto da CPU B de maneira que irá requerer 1.2x mais ciclos que A para o programa P.
- Qual é a taxa de clock que o projetista deve considerar para B?

Compreendendo o desempenho

- Um programa requer:
  - Uma quantidade de instruções (instruções de máquina)
  - Uma quantidade de ciclos
  - Uma quantidade de segundos

- Para relacionar essas quantidades temos:
  - Tempo de ciclo (quantidade de segundos por ciclo)
  - Taxa de clock (quantidade de ciclos por segundo)
  - CPI (quantidade de Ciclos por Instrução)
  - MIPS (Millions of Instructions per Second)

Desempenho

- Desempenho é determinado pelo tempo de execução

- Tomar uma das variáveis como indicadora de desempenho?
  - # de ciclos para executar o programa?
  - # de instruções em um programa?
  - # de ciclos por segundo?
  - # de ciclos médio por instrução?
  - # de instruções médio por segundo?

- Erro comum: Tomar uma das variáveis como indicativo de desempenho quando, na realidade, não é!

Exemplo de CPI

- Suponha duas implementações do mesmo conjunto de instruções
- Para um programa P,

  Máquina A possui tempo de clock de 250 ps e CPI=2.0
  Máquina B possui tempo de clock de 500 ps CPI=1.2

  Qual máquina é mais rápida para esse programa e por quanto?

- *Se duas máquinas possuem o mesmo ISA qual das seguintes medidas será igual para as duas máquinas?*
- *clock rate*
- *CPI*
- *execution time*
- *# of instructions*
- *MIPS*

Exemplo - # de Instruções

- O projetista de compilador está tentando decidir entre duas seqüências de código para uma máquina particular. A partir da implementação em Hw, existem 3 diferentes classes de instruções: classe A, classe B e classe C, requerendo um, dois e três ciclos, respectivamente.

  A primeira seqüência de código possui 5 instruções:
  - 2 de A, 1 de B, 2 de C

  A segunda seqüência possui 6 instruções: 4 de A, 1 de B, e 1 de C.

  Que seqüência executará mais rápido? Quão mais rápido será? Qual é a CPI para cada seqüência?

Exemplo MIPS

- Dois compiladores diferentes estão sendo testados com uma máquina com 4GHz com três diferentes classes de instruções: Classe A, Classe B, Classe C, que requerem um, dois e três ciclos, respectivamente. Ambos os compiladores são usados para gerar código para uma aplicação

- O primeiro código usa 5 milhões da classe de instruções A, 1 milhão de classes de instruções B, e 1 milhão da classe de instruções C.

- O segundo compilador utiliza: 10 milhões da classe A, 1 milhão da classe B e 1 milhão da classe C

- Qual seqüência será mais rapida de acordo com a medida MIPS?
- Qual seqüência será mais rápida de acordo com o tempo de execução?

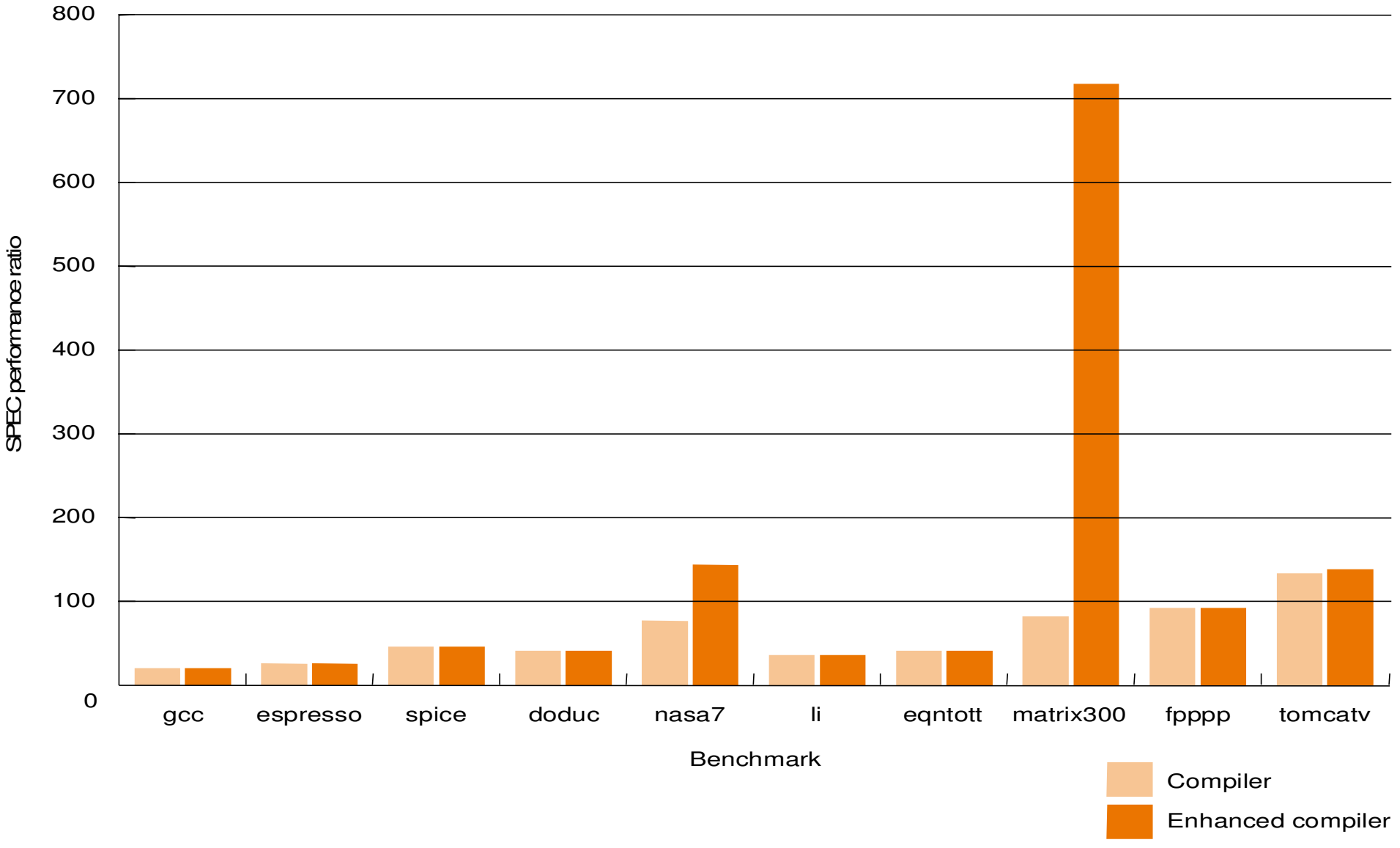
Benchmarks

- O desempenho é melhor determinado ao executar aplicações reais
  - Usar programas com carga de trabalho determinada
  - Ou de classes de aplicações conhecidas
    e.g., compilers/editors, multimídia, etc.

- SPEC (*System Performance Evaluation Cooperative*)
  - Fabricantes acordaram em um conjunto de programas reais e e entradas padronizadas
  - É um ótimo indicador de desempenho e (tecnologia de compilação)

SPEC ‘89

- Melhorias no compilador e desempenho



SPEC CPU2000

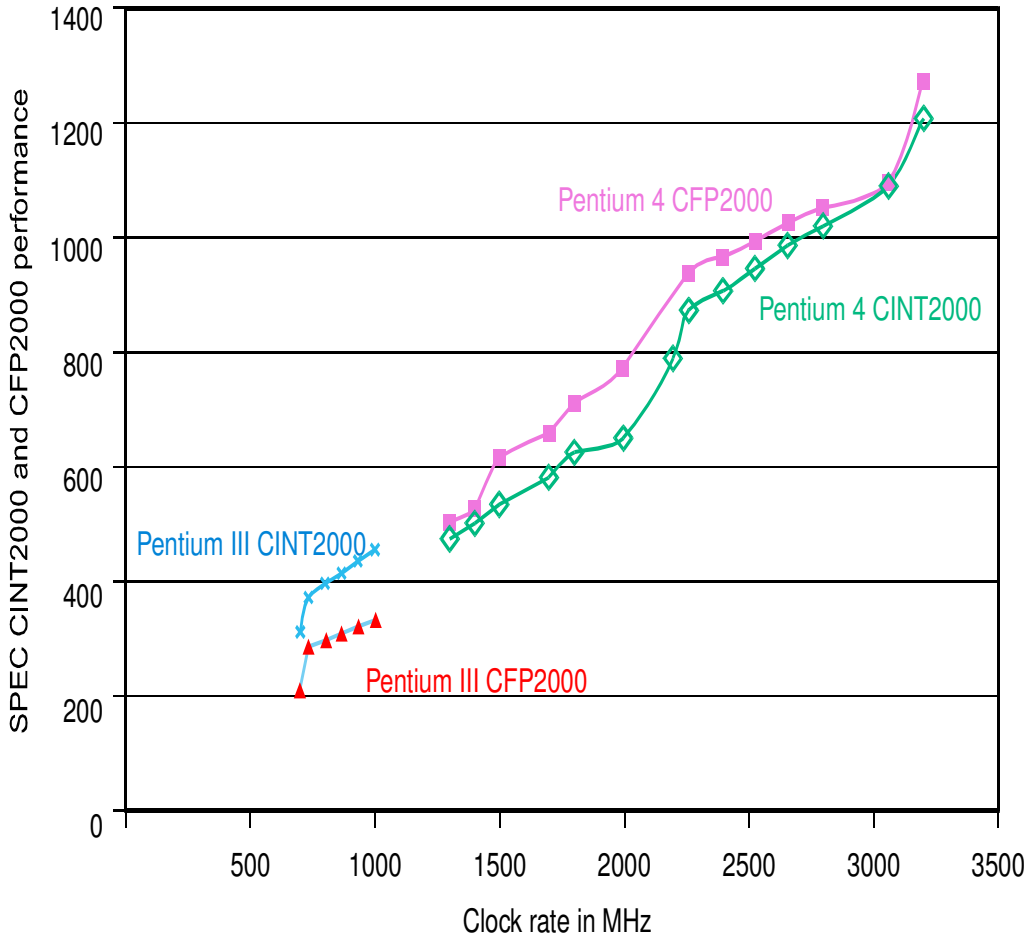
Integer benchmarks		FP benchmarks	
Name	Description	Name	Type
gzip	Compression	wupwise	Quantum chromodynamics
vpr	FPGA circuit placement and routing	swim	Shallow water model
gcc	The Gnu C compiler	mgrid	Multigrid solver in 3-D potential field
mcf	Combinatorial optimization	applu	Parabolic/elliptic partial differential equation
crafty	Chess program	mesa	Three-dimensional graphics library
parser	Word processing program	galgel	Computational fluid dynamics
eon	Computer visualization	art	Image recognition using neural networks
perlbmk	perl application	equake	Seismic wave propagation simulation
gap	Group theory, interpreter	facerec	Image recognition of faces
vortex	Object-oriented database	ammp	Computational chemistry
bzip2	Compression	lucas	Primality testing
twolf	Place and rote simulator	fma3d	Crash simulation using finite-element method
		sixtrack	High-energy nuclear physics accelerator design
		apsi	Meteorology: pollutant distribution

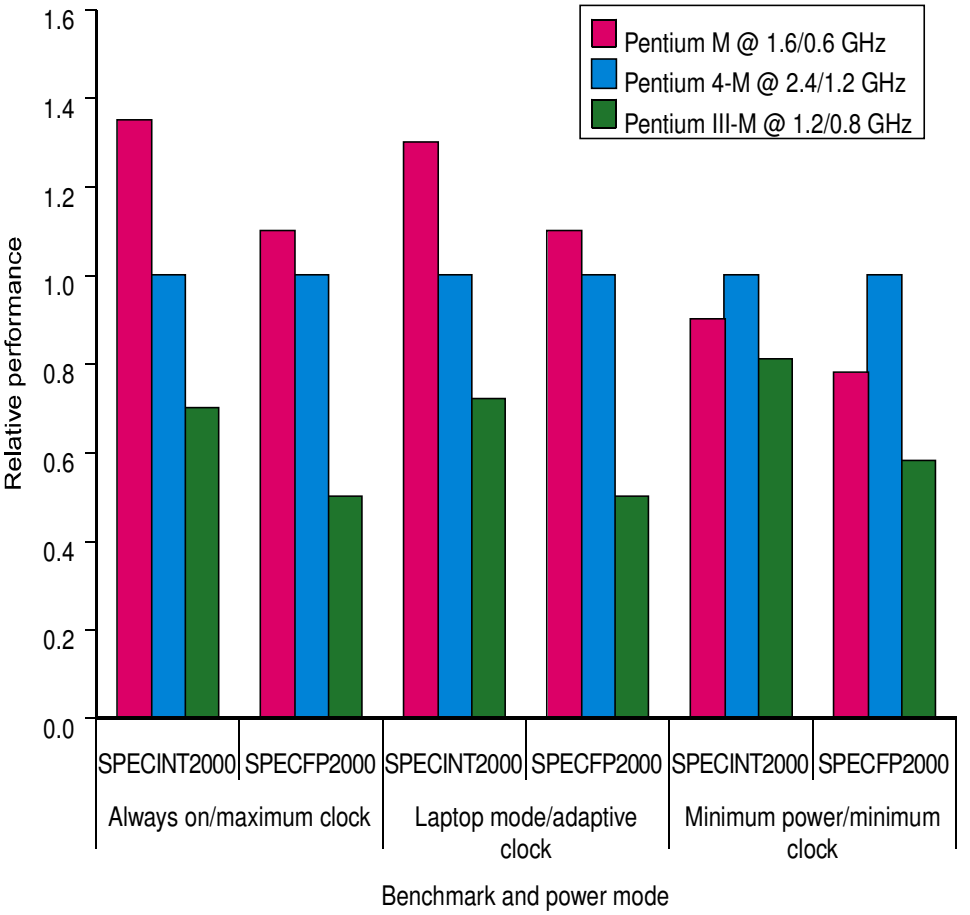
FIGURE 4.5 The SPEC CPU2000 benchmarks. The 12 integer benchmarks in the left half of the table are written in C and C++, while the floating-point benchmarks in the right half are written in Fortran (77 or 90) and C. For more information on SPEC and on the SPEC benchmarks, see www.spec.org. The SPEC CPU benchmarks use wall clock time as the metric, but because there is little I/O, they measure CPU performance.

SPEC 2000

Duplicar a taxa de clock duplica o desempenho?

Uma máquina com menor taxa de clock pode ter o melhor desempenho?





Lei de Amdahl

Tempo de Execução Após Melhoria =

Tempo de execução não afetado+(Tempo de execução afetado/Quantidade de melhoria)

- Exemplo:

  Supor o programa que executa em 100 segundos em uma máquina, e a multiplicação é responsável por 70% do tempo de execução.Quanto deve ser melhorado da velocidade de multiplicação se quisermos que o programa execute 4x mais rápido?

  E se fosse 5x mais rápido?

Exemplo

- Foi realizada uma melhoria nas instruções de ponto-flutuante de uma máquina para executarem 5x mais rápido. Se o tempo de execução de um programa P, antes da melhoria, era de 10 segundos, qual será o speedup se ½ dos 10 seg. são dedicados às operações de ponto-flutuante?

- Procura-se por um benchmark para testar a característica descrita acima. Além disso, queremos que o benchmark alcance um speedup de 3. Há um benchmark que executa em 100 seg. sem as melhorias realizadas. Quanto, do tempo de execução total, as instruções de ponto-flutuante teriam que representar nesse programa para alcançar o speedup?

Lembre-se que...

- Desempenho possui especificidades de um programa para outro
  - Tempo total de execução é um resumo consistente de desempenho
- Para uma das arquitetura, o aumento de desempenho vem de:
  - Aumento na taxa de clock
  - Melhorias na organização do processador de modo a diminuir a CPI
  - Melhorias no compilador que reduzem a CPI e/ou número de instruções
  - Escolhas da linguagem/algoritmo que afetam o número de instruções
- Erro comum: esperar que a melhoria de um aspecto (uma medida) do desempenho possa afetar o desempenho final de todo o sistema