

Medição do Desempenho

Benchmarks

Medição do Desempenho

IDEAL: obter um número que caracterize completamente o desempenho da máquina

IMPOSSÍVEL: complexidade dos sistemas
grande variedade de aplicações
grande variedade de condições de utilização

CUIDADO: os números fornecidos pelos vendedores podem ser enganadores.

- em que condições foram conseguidos?
- reflectem a carga real que o utilizador vai impor ao sistema?

Medição do Desempenho

Porquê medir o desempenho?

- escolha do *hardware* mais apropriado
- escolha do *software* (ex. compilador) mais apropriado
- escolha do algoritmo mais eficiente
- identificação de ineficiências num sistema (*hw*, *sw*)
- selecção de alternativas de desenho
 - qual o *instruction set* mais apropriado
 - selecção da organização do processador
 - selecção da organização da memória
 - ...

Definição de Desempenho

Avião	Capacidade	Alcance (milhas)	Velocidade (m.p.h)
Boeing 777	375	4630	610
Boeing 747	470	4150	610
Concorde	132	4000	1350

Qual o melhor avião?

Definição de Desempenho

Utilizador – minimização do **tempo de execução** (ou resposta)

Centro de Computação – maximização do **débito** (*throughput*)

$$Desempenho = \frac{1}{T_{exec}}$$

$$Desempenho\ relativo\ (ou\ ganho) = \frac{Desempenho_A}{Desempenho_B}$$

Medição do Desempenho

PROBLEMA: Como garantir que a carga (*workload*) a que a máquina é sujeita durante a medição é a mesma carga a que o utilizador normalmente a sujeita?

São utilizados conjuntos de programas, designados por *benchmarks*, que se espera que reflectam a *workload* da maior parte dos utilizadores.

PROBLEMA: Como garantir que vendedores e fabricantes não manipulam os testes no sentido de melhorarem os resultados?

Os conjuntos de testes com maior credibilidade são definidos por comissões independentes, que especificam rigorosamente as condições em que estes devem ser realizados e a forma como os resultados devem ser documentados.

Tipos de *Benchmarks*

Synthetic Benchmarks – pequenos programas desenvolvidos especificamente para medir alguma característica específica da máquina. Normalmente não realizam nenhuma tarefa específica.

Desvantagens

1. Não reflectem a carga que um utilizador aplica à sua máquina;
2. Programas pequenos que utilizam apenas a *cache*;
3. Alguns compiladores geram código otimizado para estes testes. Estas optimizações não podem depois ser usadas em aplicações reais.

Vantagens

1. Na fase inicial de desenho de um sistema estes testes são muito úteis, pois são fáceis de compilar e mesmo de simular.

Exemplos: Dhrystone (int) e Whetstone (FP)

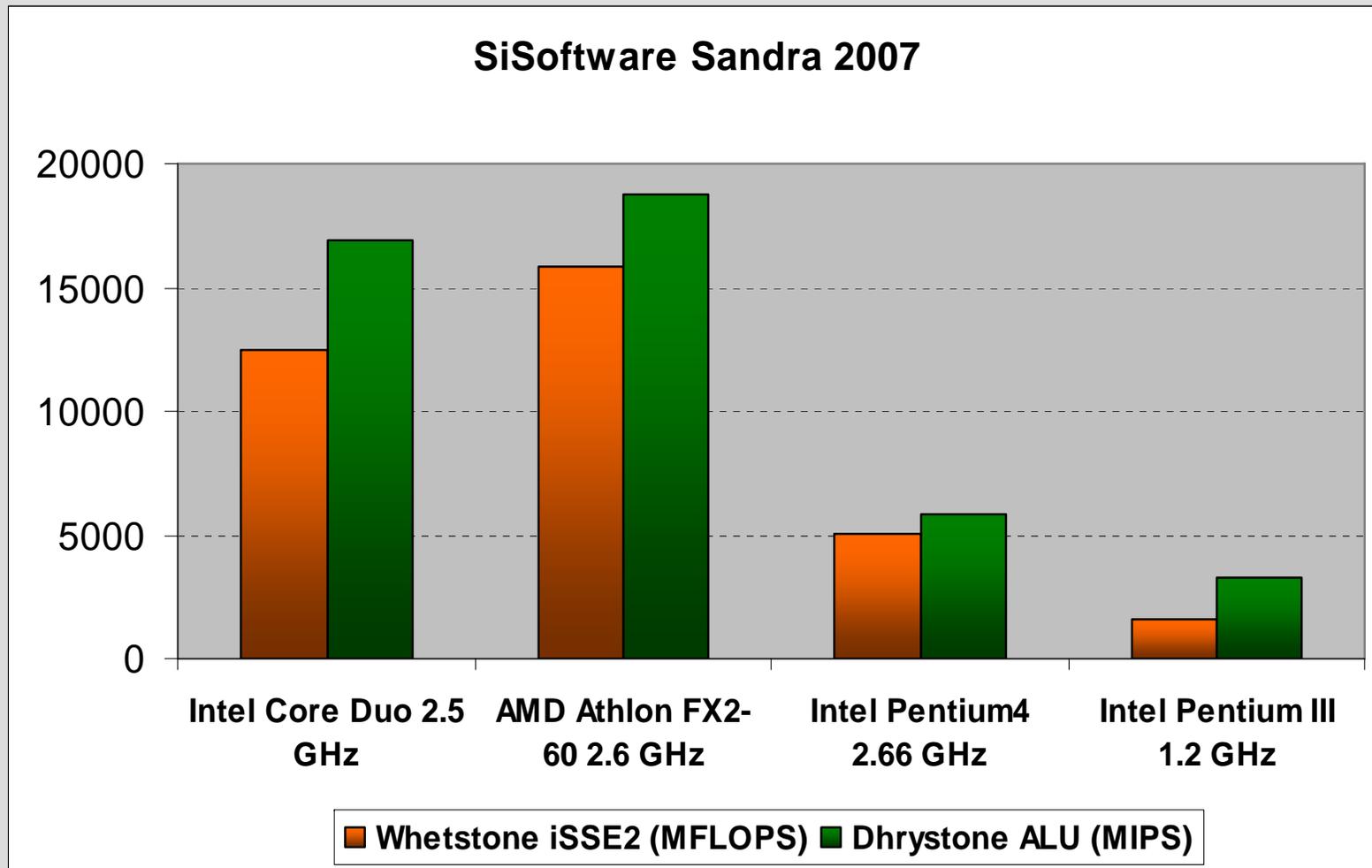
Dhrystone (inteiros)

- Teste sintético que contém uma mistura representativa de operações inteiras:
 - invocação de procedimentos
 - utilização de apontadores, inteiros e caracteres
 - atribuições e cálculo de expressões
- Desenvolvida em 1984 por Reinhold Weicker em Ada. Melhorada e convertida para C em 1989 por Weicker e Richardson
- Resultados em *Dhrystone loops per second*
Resultados em **MIPS** (Milhões de instruções por segundo) não podem ser usados para comparar diferentes arquiteturas (ex. CISC vs. RISC)
- Muito divulgada, mas muitos compiladores incluem otimizações específicas para este teste

Whetstone (vírgula flutuante)

- Teste sintético que contém uma mistura representativa de operações em vírgula flutuante:
 - `abs`, `sqrt`, `exp`, `alog`, `sin`, `cos`, `atan`, ...
- Desenvolvida em Algol60, em 1972
- Resultados em MWIPS (Milhões de Whetstone instruções por segundo) ou em MFLOPS (Milhões de operações em vírgula flutuante por segundo)
- Muito divulgada, mas muitos compiladores incluem otimizações específicas para este teste

Medição do Desempenho - Resultados



Tipos de *Benchmarks*

Conjuntos de aplicações reais – aplicações seleccionadas por representarem cargas típicas para um grande número de utilizadores

Desvantagens

1. Difícil seleccionar conjuntos de aplicações que garantidamente representem uma grande maioria de utilizadores;
2. Estes testes levam muito tempo a executar e as condições de teste e relatório de resultados são geralmente muito exigentes;

Vantagens

1. O utilizador pode geralmente confiar que os resultados reflectem com algum grau de precisão o desempenho a esperar da sua máquina;
2. Tratando-se de aplicações reais torna-se difícil aos fabricantes introduzirem características especiais no *hardware* ou nos compiladores para inflaccionar os resultados.

Exemplos: SPEC Benchmarks

SPEC - Standard Performance Evaluation Corporation

A SPEC (<http://www.spec.org/>) é uma associação criada em 1989 por um grupo de companhias para normalizar:

- o conjunto de testes a que as máquinas devem ser submetidas;
- as condições em que estes testes devem ser realizados;
- a forma como os resultados devem ser documentados

Benchmark	Descrição
SPEC WEB'2005	Servidores WEB
SPEC HPC'2002	High Performance Computing
SPEC JBB'2005	Java Virtual Machine
SPEC SFS'97_R1	Sistem File Server
SPEC MAIL'2001	Servidores de eMail
SPEC CPU'2006	Processador – memória - compilador

SPEC CPU'2006

Conjunto de programas cuidadosamente seleccionados para representarem a carga que um utilizador “regular” impõem à sua máquina.

Inclui vários testes dos quais se destacam:

Benchmark	Descrição
SPECint_base2006	Programas com operações em inteiros, compilados sem optimizações.
SPECint2006	Programas com operações em inteiros, compilados com optimizações.
SPECfp_base2006	Programas com operações em vírgula flutuante, compilados sem optimizações.
SPECfp2006	Programas com operações em vírgula flutuante, compilados com optimizações.

SPECint2006

Teste	HLL	Descrição
445.gobmk	C	Artificial Intelligence
464.h264ref	C	Vídeo Compression
403.gcc	C	C Programming Language Compiler
429.mcf	C	Combinatorial Optimization
458.sjeng	C	Game Playing: Chess
471.omnetpp	C++	Discrete Event Simulation
456.hmmer	C	Search Gene Sequence
400.perlbench	C	PERL Programming Language
462.libquantum	C	Physics: Quantum Computing
473.astar	C++	Path finding algorithms
401.bzip2	C	Compression
483.xalancbmk	C++	XML processing

12 programas

SPECfp2006

Conjunto de 17 aplicações incluindo:

- dinâmica de fluidos;
- química quântica;
- dinâmica molecular;
- simulação da relatividade geral;
- análise de elementos finitos;
- *ray tracing*;
- meteorologia;
- reconhecimento de discursos.

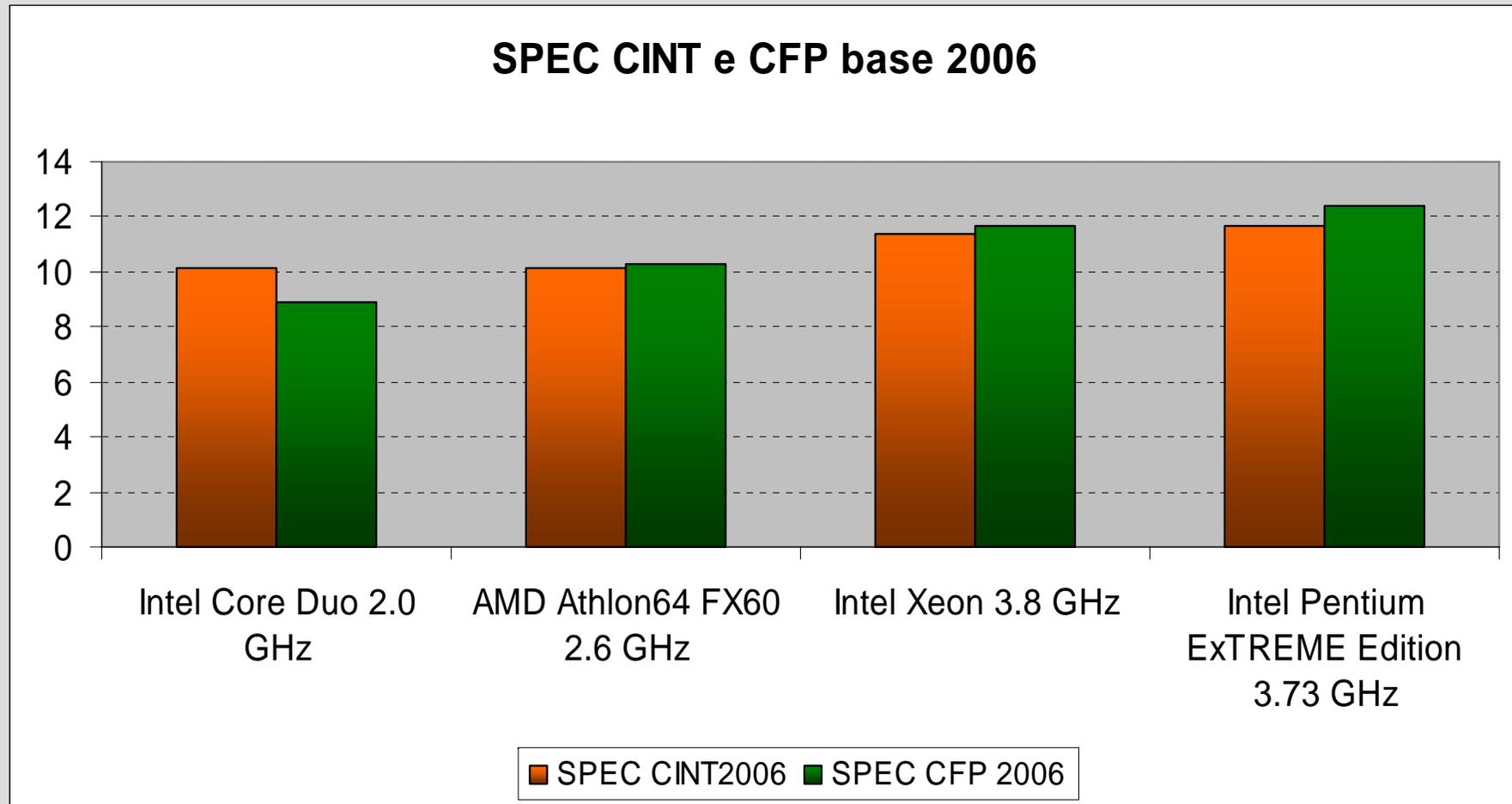
SPEC CPU2006

Como calcular os índices SPECint2006 e SPECfp2006?

1. O tempo de execução de cada teste numa máquina de referência (SUN Ultra Enterprise 2) é dividido pelo tempo de execução na máquina a testar. Chama-se a esta razão o **SPEC ratio**.
2. É calculada a média geométrica dos vários SPEC ratios.

$$SPEC = \sqrt[n]{\prod_i^n SPECratio_i}$$

SPEC CPU 2006



Sumário

Tema	H & P
Benchmarks	Sec. 2.4
SPEC	Sec 2.6
Média Geométrica	Sec 2.7
Synthetic Benchmarks	Sec 2.7

Para mais informação ver: <http://www.spec.org/>