

Testes de Carga em Sistemas de Computação e Benchmark

Testes de carga em sistemas reais são feitos através de programas que geram carga no sistema exercitando um conjunto adequado de instruções. Submeter o sistema a tais testes de carga é o que se denomina Benchmark. Existem programas de Benchmark para exercitar determinados tipos de sistemas e de carga nestes sistemas.

1 Categorias de Benchmarks

As categorias em que se enquadram os programas de Benchmark são as seguintes:

- Instrução de soma da UCP: utilizado em testes nos sistemas antigos.
- Mix de instruções
- Kernel (núcleo)
- Programas sintéticos
- Aplicações para Benchmark

1.1 Mix de Instruções

É um conjunto de instruções organizado como um programa de computador, sendo que as instruções aparecem no programa de acordo com a ocorrência em uma carga real.

Um mix comum é o de Gibson desenvolvido por Jack C. Gibson (1959) para um IBM 704. O mix considera uma frequência de instruções observadas em programas que executavam nos computadores IBM 704. O programa de teste utilizando o mix fornece um tempo médio para a execução de instruções.

Mix de Instruções de Gibson

| Instrução | Frequência |
|---|------------|
| Load e Store | 31.2 |
| Soma e subtração em ponto fixo | 6.1 |
| Comparações | 3.8 |
| Desvios | 16.6 |
| Soma e subtração em ponto flutuante | 6.9 |
| Multiplicação em ponto flutuante | 3.8 |
| Divisão em ponto flutuante | 1.5 |
| Multiplicação em ponto-fixo | 0.6 |
| Divisão em ponto-fixo | 0.2 |
| Shift | 4.4 |
| Operações lógicas e, ou | 1.6 |
| Instruções que não utilizam registradores | 5.3 |
| Indexação | 18.0 |

O mix de Gibson tem a desvantagem de não representar as instruções de computadores atuais com diferentes modos de endereçamento tais como mecanismos de

endereçamento virtual e outras características de processador como pipeline e multiprocessamento. também não contempla testes dependentes de periféricos.

Contudo é um modo simples e efetivo de se comparar computadores com arquiteturas semelhantes e para se estimar o tempo de execução de algoritmos e programas de sistema. Estas medidas medem apenas o desempenho do processador e não o desempenho total do sistema que depende de outros fatores diversos.

Outras medidas, inversas ao de tempo médio para execução de instruções são:

MIPS - milhões de instruções por segundo

MFLOPS - milhões de operações de ponto flutuante por segundo.

1.2 Kernel

É uma generalização de mix de instruções para permitir o exercício de outras características do processador. Considera, em lugar de um mix, um conjunto de instruções que constituem uma função de alto nível, isto é um serviço provido pelos processadores.

Programas de benchmark nesta categoria: crivo de Erastóstenes, quebra-cabeças, busca em árvores, função de Ackermann, Inversão de matrizes, ordenação.

1.3 Programas sintéticos

São programas que incluem também uso de entrada e saída e exercitam as chamadas de sistema. O primeiro destes programas foi proposto por Buchholz (1969) que usou a denominação de programa sintético. São programas portáteis, escritos em linguagens de alto-nível tais como Fortran e Pascal.

A desvantagem é que eram muito pequenos e não exercitavam os mecanismos de memória e de referência a disco.

1.4 Aplicações de Benchmark

São Programas mais complexos que exercitam quase todos os recursos dos sistemas, tais como processadores, dispositivos de E/S, rede, base de dados. Atualmente os programas são especializados para teste de determinadas características: rede, Web, transações, etc.

Muitos destes programas são produtos à venda.

2 Benchmarks mais conhecidos

2.1 Crivo de Erastóstenes

Utilizado para comparar microprocessadores, computadores pessoais e linguagens de alto-nível. É o algoritmo para gerar todos os números primos abaixo de um número n . Após a geração de todos inteiros entre 1 e n , serão retirados do conjunto todos os múltiplos de 2, 3, 5,...

2.2 Benchmark: Função de Ackermann

Foi utilizado para determinar o desempenho de processadores na execução do mecanismo de chamada de procedimento em linguagens ALGOL-like.

A função de Ackermann é definida recursivamente como:

```
Ackermann (m,n) =  
    se m=0  
    então n+1  
    senão se n=0  
        então Ackermann(m-1,1)  
        senão Ackermann(m-1, Ackermann(m,n-1))
```

2.3 Whetstone

Utilizado pelo British Central Computer Agency, consiste de 11 módulos projetados para exercitar a frequência dinâmica de operações observadas em 949 programas Algol (1975).

Exercita operações em ponto-fixa e flutuantes, endereçamento de matrizes, chamadas de procedimentos, passagem de parâmetros. É representativo de programas científicos e de engenharia. O resultado é medido em KWIPS (Kilo Whetstones Instruction per Second). Foi traduzido para outras linguagens: Fortran, PL/I.

2.4 LINPACK

Desenvolvido por Jack Dongarra (1983) do Argonne National Laboratory. Consiste de programas para resolver sistemas de equações lineares densas (não esparsas), utilizando o pacote de sub-rotinas LINPACK. Possui alta porcentagem de somas e multiplicações em ponto flutuante. Destinava-se a avaliar sistemas voltados para engenharia mecânica.

2.5 Dhrystone

Desenvolvido em 1984 por Reinhold Weicker da Siemens e disponível em C, Pascal e ADA.

O resultado é medido em DIPS (Dhrystone Instructions per Second). Contem muitas chamadas de procedimento e é considerado representativo de ambientes de sistemas de programação.

É uma medida adequada para medir desempenho de operações com inteiros, não exercitando ponto flutuante ou E/S.

2.6 Lawrence Livemore Loops

Consiste de 24 testes separados que exercitam predominantemente cálculos científicos vetoriais, implementados no Lawrence Livemore National Laboratories (1986).

O resultado medido em MFLOPS é apresentado nos casos mínimos, máximos e médios utilizando diferentes definições de média.

Consiste de código extraído de aplicações reais do Laboratório e se tornou um padrão para a determinação da potência de sistemas computacionais.

2.7 SPEC

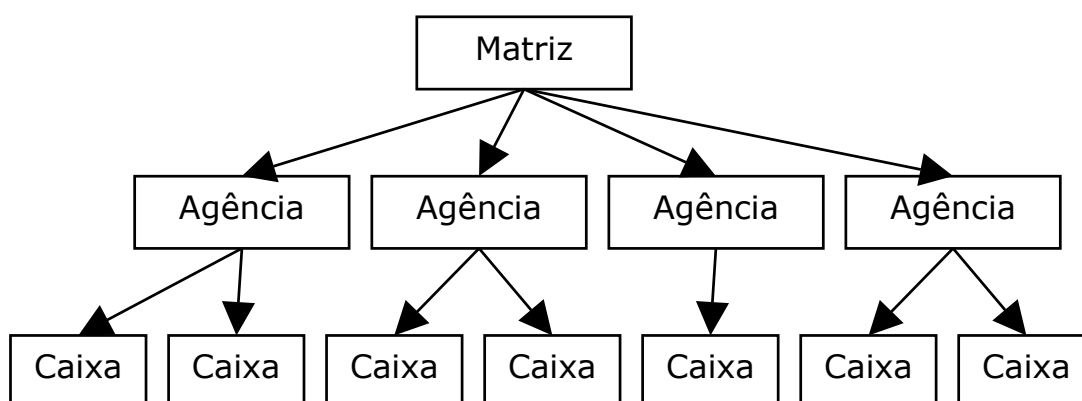
O **System Performance Evaluation Corporation** (SPEC) é uma entidade sem fins lucrativos, formados por vendedores/fabricantes de computadores, que desenvolveu um conjunto padronizado de benchmarks.

Consiste de 10 aplicações:

- GCC: Tempo do compilador GNU para compilar 19 programas .
- Espresso: Ferramenta de Eletronic Design Automation(EDA) que realiza minimização de funções booleanas utilizando lógica heurística para programação de PLA.
- Spice 2g6: Outra ferramenta de EDA para simulação de circuitos analógicos.
- Doduc: Realiza simulação de Monte Carlo de processos de Reatores Nucleares.
- NASA7: conjunto de funções que realizam operações de ponto flutuante em dupla precisão e operações matriciais.
- LI: tempo necessário para resolver o problema das 9-rainhas escrito em LISP.
- Eqntott: Converte uma equação booleana em uma tabela verdadeiro-falso.
- Matrix300: Realiza operações com matrizes 300x300 utilizando o pacote LINPACK em lógica de ponto flutuante de dupla precisão.
- Fpppp: Benchmark de química quântica utilizando lógica de ponto flutuante de dupla precisão para integração/derivação.
- Tomcatv: Programa que opera com reticulados vetoriais em lógica de ponto flutuante de dupla precisão. Adequado para testes de memória compartilhada em multiprocessadores.

2.8 Débito-Crédito

Representa uma rede bancária distribuída e está sendo utilizado desde 1973 e se tornou um padrão para a avaliação de sistemas transacionais.



Foi desenvolvido por um banco que queria que seu sistema suportasse transações on-line que atingiam pico de 100 Transações por Segundo (TPS) em suas 1.000 agências, 10.000 caixas e 10.000.000 de contas-correntes.

Begin-Transaction

Ler mensagem do terminal (100 bytes)
Atualizar saldo
Escrever extrato
Atualizar caixa (Teller)
Atualizar agência
Escrever mensagem no terminal

Commit-Transaction

2.9 TPS - Transaction Processing System

A organização sem fins lucrativos Transaction Processing Performance Council, criada em 1988, definiu benchmarks para sistemas de processamento transacional e de bases de dados.

Foram definidos:

- TPC-C Mede o desempenho máximo mantido pelo sistema em transações por minuto de transações comerciais.
- TPC-H Mede o desempenho de ambientes de suporte a decisões.
- TPC-R é similar ao TPC-H mas executa um conjunto de consultas a bases de dados.

3 Bibliografia

- [1] Jain, R., "The Art of Computer Systems Performance Analysis", John Wiley & Sons Inc, ISBN: 0-471-50336-3, 1991, 685 p.
- [2] Law, A. M., Kelton, W. D., "Simulation Modeling and Analysis", 3rd ed., McGraw-Hill Companies Inc, 2000, ISBN 0-07-059292-6, 760p.

4 Links

BAPCo: The Business Applications Performance Council

<http://www.bapco.com/>

EEMBC: The EDN Embedded Microprocessor Benchmark Consortium

<http://www.eembc.org/>

IOZone

<http://www.iozone.org/>

PDS: The Performance Database Server

<http://netlib2.cs.utk.edu/performance/html/PDStop.html>

SPEC

<http://www.specbench.org/>

Transaction Processing Performance Council

<http://www.tpc.org>

ZDNET

<http://www.zdnet.com>

5 Exercícios

- 1) Utilize uma ferramenta de benchmark de domínio público para analisar o seu computador pessoal em comparação com os computadores de outros dois colegas. Faça um relatório com os resultados da análise. Apresente gráficos comparativos de desempenho dos computadores utilizando métricas tais como MIPS, MFLOPS ou SPECS. Verifique como os parâmetros do computador (memória, disco, placas de vídeo, CPU) afetam o desempenho. Apresente suas conclusões sobre o desempenho dos computadores analisados. Escolha computadores de gerações próximas (não dá para concluir muita coisa quando se compara um Pentium I com um Pentium III, por exemplo pois é óbvio que o último é mais rápido) e mesmo sistema operacional.
- 2) Utilize uma ferramenta básica tal como o Ping ou Traceroute para determinar o tempo médio de resposta, em um determinado horário, em diferentes segmentos de rede para acesso a um site. Por exemplo, verifique os tempos médios de acesso ao site Intranet da Poli ou a algum site comercial de vendas on-line. Faça um certo número de medidas e determine os tempos médios em cada segmento. Acesse o mesmo site a partir de computadores diferentes, por exemplo, de dentro da Escola e de sua casa. Apresente tabelas indicando a contribuição porcentual de cada segmento no tempo total de acesso.