Capítulo 3

Avaliação de Desempenho

Este capítulo aborda como medir, informar e documentar aspectos relativos ao **desempenho** de um computador. Além disso, descreve os principais fatores que influenciam no seu desempenho. A razão para este estudo é que o desempenho do *hardware* é responsável direto pela eficácia do sistema, que inclui não só o *hardware*, mas também o *software*.

A avaliação de desempenho de um sistema constitui um desafio devido à diversidade e complexidade dos sistemas modernos.

3.1 Definição de Desempenho

Para melhorar o desempenho de um *software* executado num determinado *hardware* é necessário conhecer quais fatores do *hardware* influenciam no desempenho global do sistema. Além disso, devemos determinar a importância relativa de cada um destes fatores. Entre os fatores do *hardware* que têm maior influência no desempenho podemos citar:

- como o compilador utiliza as instruções da máquina na geração do código de um programa;
- como o *hardware* implementa as instruções e
- como a memória e os dispositivos de entrada e saída se comportam durante o processamento do programa.

A compreensão de como determinar o impacto no desempenho de cada um destes fatores é importante para se entender a motivação dos projetos de aspectos específicos da máquina.

Nos sistemas de computadores o desempenho pode ser definido de maneiras diferentes. Podemos considerar como o sistema de melhor desempenho aquele que executa em menor tempo um determinado programa, ou ainda, aquele sistema que executa o maior número de programas num determinado intervalo de tempo. Dessa forma, estamos considerando dois aspectos diferentes para avaliar o desempenho. O primeiro é o **tempo de resposta** ou **tempo de execução da aplicação**. O segundo é denominado ***throughput***, e considera a quantidade de trabalho executado na unidade de tempo. Estes aspectos refletem a visão do usuário e a visão do gerente do sistema.

Podemos relacionar o desempenho e o tempo de execução em uma máquina como:

$$\textbf{Desempenho} = \frac{1}{\textbf{Tempo de execução}}.$$

Isto significa que, se considerarmos duas máquinas *X* e *Y*, se o desempenho de *X* for melhor do que *Y*, teremos:

$$\textbf{Desempenho } X > \textbf{Desempenho } Y$$

ou

$$\frac{1}{\textbf{Tempo de execução de } X} > \frac{1}{\textbf{Tempo de execução de } Y}$$

$$\textbf{Tempo de execução de } Y > \textbf{Tempo de execução de } X$$

Assim, o tempo de execução em *Y* é maior do que o tempo de execução em *X*. A máquina *X* é mais rápida do que a máquina *Y*. Além disso, podemos relacionar o desempenho de duas máquinas de modo quantitativo, *X* é *n* vezes mais rápida do que *Y*. Dessa forma, o **desempenho relativo** entre as máquinas *X* e *Y* é dado pela seguinte fórmula:

$$\frac{\textbf{Desempenho } X}{\textbf{Desempenho } Y} = n \text{ ou}$$

$$\frac{\textbf{Tempo de execução } Y}{\textbf{Tempo de execução } X} = n.$$

Através destas fórmulas podemos dizer que para melhorar o desempenho é necessário diminuir o tempo de execução.

3.2 Medidas de Desempenho

O tempo é a medida de desempenho de um sistema de computador. Ele, em geral, é medido em segundos e pode ser definido de maneiras diferentes.

O tempo de execução ou tempo de resposta define o tempo total para se completar uma tarefa computacional, incluindo os acessos à memória e ao disco, as atividades de entrada e saída e o *overhead* do sistema operacional.

Quando o sistema é compartilhado o processador pode trabalhar em vários programas simultaneamente. Nestes casos o desempenho melhora quando aumentamos o *throughput*. Dessa forma, muitas vezes é interessante distinguir entre o **tempo total de execução** de um programa e o tempo gasto pelo processador num programa em particular. Este último tempo é denominado **tempo do processador**.

O tempo do processador também pode ser dividido em **tempo do usuário** e **tempo do sistema**. O tempo do usuário é o tempo gasto na execução das instruções do programa do usuário. Já o tempo do sistema é o tempo gasto pelo sistema operacional para executar tarefas em benefício do programa do usuário.

Ao considerarmos os detalhes de uma máquina é conveniente utilizar outra métrica para avaliarmos o desempenho. Os projetistas medem a velocidade do *hardware* na execução de suas funções básicas com o ***clock***. O *clock* possui uma taxa constante e determina o momento da ocorrência de eventos do próprio *hardware*. Estes intervalos de tempo discretos são denominados de ciclos de *clock*, *ticks*, *clock ticks*, períodos de *clock*, *clocks* ou ciclos. O tamanho de um período de *clock* é referenciado tanto como o tempo necessário para completar um ciclo de *clock* quanto como a freqüência do *clock* (inverso do ciclo de *clock*). Por exemplo, um ciclo de *clock* igual a 2 ηs corresponde a uma freqüência de 500MHz, que é o inverso do ciclo de *clock*.

3.3 Relação entre as Métricas

Fórmulas bastante simples relacionam a medida do tempo de execução gasto no processador com a métrica básica baseada nos ciclos de *clock* e tempo do ciclo de *clock*. Estas fórmulas estão mostradas a seguir:

Tempo de execução no processador = Número de ciclos de *clock* do processador X **ciclo do *clock***

ou

Tempo de execução no processador = Número de ciclos de *clock* do processador / Freqüência do *clock*

Essas fórmulas demonstram que o desempenho do sistema pode ser melhorado se reduzirmos o tamanho do ciclo de *clock* ou o número de ciclos de *clock* necessários à execução de um programa.

Estas fórmulas não incluem qualquer referência ao número de instruções necessárias à execução de um programa. O tempo de execução também depende do número de instruções do programa. O número de ciclos de *clock* necessários à execução de uma instrução é dado por:

Número de ciclos de *clock* = Número de instruções do programa X **Número médio de ciclos por instrução**

A expressão ciclos de *clock* por instrução é abreviada por **CPI** (*clock cycles per instruction*). A CPI é a média dos tempos gastos por todas as instruções executadas pelo programa. Este parâmetro permite a comparação entre diferentes implementações de uma mesma arquitetura do conjunto de instruções, uma vez que o número de instruções para a execução do programa nas diferentes implementações é o mesmo.

Podemos escrever a equação do desempenho em termos da quantidade de instruções, da CPI e do ciclo de *clock*.

Tempo de processador = Número de instruções X **CPI** X **ciclo de *clock***

ou

$$\textbf{Tempo de processador} = \frac{\textbf{Número de instruções X CPI}}{\textbf{Freqüência do } \textit{clock}}$$

Estas fórmulas são úteis porque separam os três principais fatores que afetam o desempenho: quantidade de instruções, média dos ciclos gastos por instrução e período do *clock*.

3.4 Escolha de Programas para Avaliar o Desempenho

O conjunto de programas executados pelo usuário chama-se **carga de trabalho** (*work load*). Para avaliar o desempenho de diferentes sistemas o usuário poderia simplesmente comparar o tempo de execução da sua carga de trabalho nos sistemas. Entretanto, em geral, os usuários não possuem uma carga de trabalho típica que possa ser utilizada para a avaliação.

Existem quatro níveis de programas que podem ser usados para avaliação de desempenho, eles estão listados em ordem decrescente de precisão de previsão: programas reais, núcleos ou *kernels* (pedaços de programas reais), *toy benchmarks* (programas com 10 a 100 linhas de código que produzem um resultado conhecido a priori) e *benchmarks* sintéticos (similar em filosofia aos núcleos, tentam casar a freqüência média de operações de um grande conjunto de programas).

Os *benchmarks* são conjuntos de aplicações que representam cargas de trabalho cujo objetivo é estimar o desempenho das cargas de trabalho reais. Os *benchmarks* podem conter aplicações típicas de processamento científico, compiladores, processadores de texto entre outras. Se forem pequenos os *benchmarks* podem ser mais facilmente padronizados.

3.5 Comparação e Documentação de Desempenho

Uma vez selecionados os programas adequados para usar como *bechmarks* e decidida à métrica de avaliação, tempo de resposta ou *throughput*, é necessário decidir como documentar os dados de desempenho obtidos a partir de diferentes *benchmarks*.

As pessoas preferem determinar um número único para comparar o desempenho de diferentes máquinas. A questão, portanto, é como calcular este número.

A maneira mais simples de considerar o desempenho relativo é usar o tempo total de execução de dois programas, como por exemplo:

$$\frac{\textbf{Desempenho}_\textbf{A}}{\textbf{Desempenho}_\textbf{B}} = \frac{\textbf{Tempo de execução}_\textbf{A}}{\textbf{Tempo de execução}_\textbf{B}}$$

Suponha que tenhamos executados dois programas 1 e 2 em duas máquinas diferentes A e B, e que os tempos obtidos são os mostrados na Tabela 3.1.

	Computador A	Computador B
Programa 1 (s)	1	10
Programa 2 (s)	1000	100
Tempo total (s)	1001	110

Tabela 3.1 *Tempos de execução de dois programas.*

Observando individualmente cada tempo podemos afirmar que:

- a máquina A é 10 vezes mais rápida do que a máquina B ao executar o programa 1;
- a máquina B é 10 vezes mais rápida do que a máquina A ao executar o programa 2.

No entanto, observadas em conjunto não podemos afirmar nada sobre o desempenho relativo entre as máquinas A e B. O desempenho relativo é igual a: 1001/110 = 9.1. Isto significa que a máquina B é 9.1 vezes mais rápida que a máquina A.

Este resultado é diretamente proporcional ao tempo de execução. Caso o *work load* seja a execução dos programas A e B um número idêntico de vezes, a afirmação é válida na previsão dos tempos relativos de execução do *work load* em cada uma das máquinas.

A média dos tempos de execução é diretamente proporcional ao tempo total de execução e é dada pela média aritmética (MA):

$$MA = 1/n \sum^{n}_{i=1} Tempo_i,$$

onde $Tempo_i$ é o tempo de execução do *i*-ésimo programa do total dos *n* pertencentes ao *work load*. Quanto menor o valor do resultado melhor o tempo médio de execução, permitindo a obtenção de melhores desempenhos.

3.6 *Benchmarks*

Um dos conjuntos de aplicações mais conhecidos utilizados para testes foi desenvolvido pela cooperativa americana SPEC (*System Performance Evaluation Cooperative*). Esta cooperativa foi criada com o objetivo de melhorar as medidas e informações sobre o desempenho de sistemas computacionais. A SPEC possui *benchmarks* com características diferentes. O SPECint e o SPECfp são formados por programas que manipulam números inteiros e de ponto flutuante, respectivamente. Já o SPECWeb possui aplicações com características que buscam reproduzir o comportamento das aplicações desenvolvidas para a Web.

Existem outros *benchmarks* disponíveis para avaliação de computadores. Por exemplo, podemos citar os *benchmarks Whetstone e Dhrystone*. Para avaliar sistemas de alto desempenho, e que possuem aplicações científicas, podemos citar o *benchmark* NAS, criado pela equipe da NASA e os *benchmarks* SPLASH-1 e SPLASH-2, desenvolvidos na universidade de Stanford.

3.7 Outras Medidas de Desempenho

Algumas medidas tentam padronizar e facilitar a expressão do desempenho. Porém, apesar de simples estas métricas são válidas num contexto limitado. Além disso, ao substituírem o tempo como métrica podem gerar resultados distorcidos ou interpretações incorretas.

Uma das alternativas é a métrica MIPS (*million instruction per second*), que é dada pela seguinte expressão:

MIPS = Número de instruções / Tempo de execução X 10^6

Existem problemas com o uso da métrica MIPS. Ela especifica a taxa de execução de instruções, mas não considera a capacidade de executar mais ou menos trabalho. Portanto, não podemos comparar máquinas com conjuntos de instruções diferentes. Um outro problema é que os resultados obtidos variam entre programas no mesmo computador, o que impede que determinada máquina tenha um MIPS característico.

Uma outra alternativa é a métrica denominada MFLOPS (*million floating-point operations per second*), que é dada pela seguinte expressão:

MFLOPS = Número de operações de ponto flutuante / Tempo de execução X 10^6

Uma operação de ponto flutuante pode ser uma operação de adição, subtração, multiplicação ou divisão aplicada a operandos expressos em precisão simples ou dupla.

3.8 A Lei de Amdhal

A lei de Amdhal pode ser utilizada para demonstrar o ganho de desempenho de uma máquina. Este ganho é dito **aceleração** ou ***speedup***. Entende-se por aceleração a medida de como a máquina se comporta após a implementação de uma melhora em relação ao seu comportamento anterior. Podemos definir a aceleração como:

Aceleração = Desempenho após a melhora / Desempenho antes da melhora

Aceleração = Tempo de execução antes da melhora / Tempo de execução após a melhora

A lei de Amdhal demonstra que é errado esperar que a melhora em um dos aspectos que influenciam no desempenho da máquina resulte numa melhora no desempenho total proporcional ao tamanho do ganho inicial da fração. É melhor tornar o caso mais freqüente mais eficiente do que considerar os casos infreqüentes.