

2 – O papel da Performance

Como avaliar a performance de um processador?

O que devemos avaliar?

Frequência do processador?

Tempo de resposta à execução de um determinado software?

A performance de um determinado sistema não pode ser avaliada somente considerando-se a velocidade que um determinado software vai ser executado em uma determinada máquina.

Para realmente compreendermos como e por que um determinado software se comporta em uma determinada máquina, ou como o conjunto de instruções pode ser implementado para funcionar melhor que outro, ou para saber como determinada característica do hardware afeta a velocidade de processamento, precisamos conhecer os fatores determinantes no desempenho de uma máquina.

A eficiência de um software sobre determinado hardware depende de alguns fatores:

- Compilador utilizando as instruções da máquina na geração do código de um programa.
- A maneira como a memória e os dispositivos de entrada/saída (I/O) se comportam durante o processamento do programa.

2.1 - Definição de Performance

Para a definição da performance de uma determinada máquina é importante nos referirmos a qual das estamos utilizando. Seja a tabela abaixo onde podemos definir como métricas para os aviões apresentados como: autonomia de vôo, quantidade de passageiros transportados, velocidade de navegação, quantidade de passageiros por milhas por hora.

Avião	Capacidade (passageiros)	Autonomia (milhas)	Velocidade (milhas/hora)	Throughput (passageiros x milhas/hora)
Boeing 777	375	4630	610	228.750
Boeing 747	470	4150	610	286.700
Concorde	132	4000	1350	178.200
DC – 8 -50	146	8720	544	79.424

No caso de computadores a performance pode ser avaliada de diversas formas. Por exemplo:

1. Dois computadores rodando o mesmo software → a máquina mais rápida será a que termina primeiro.
2. Dois computadores em regime de tempo compartilhado (*time-sharing*) executando tarefas submetidas por diversos usuários → a máquina mais rápida será a que consegue executar o maior número de instruções por dia.
 - Se considerarmos como métrica o ponto de vista do usuário, a máquina mais rápida será a que executa o programa primeiro, ou seja, com o menor tempo de resposta.
 - Se considerarmos o ponto de vista do CPD, a máquina de melhor performance será a que oferece maior vazão (*throughput*), ou seja, a quantidade de trabalho executado em determinado tempo.

Métricas de vazão e tempo de resposta

Para entender como estas métricas podem ser avaliadas considere o exemplo:

As mudanças a seguir são introduzidas em um sistema de computação, melhoram a vazão, diminuem o tempo de resposta, ou afetam ambos os parâmetros?

1. Substituição do processador do computador por um modelo mais rápido.
2. Alocação de processadores adicionais a um sistema que usa vários processadores para executar programas diferentes.

Solução:

A queda no tempo de resposta quase sempre melhora a vazão. A troca do processador por um modelo mais rápido melhora tanto a vazão quanto o tempo de resposta.

No segundo caso devemos considerar a demanda de processamento. Caso a demanda de processamento seja menor que a demanda por vazão (jobs submetidos), o tempo de resposta em nada seria alterado e a melhoria seria somente na capacidade de enfileirar mais jobs.

Mas, caso a demanda por processamento seja maior que a vazão, o tempo de fila de cada programa poderia ser reduzido.

Resumindo, na maioria dos sistemas de computação melhorias no tempo de resposta afetam a vazão e vice-versa.

Dessa forma, podemos relacionar a Performance de uma máquina X com o tempo de resposta como:

$$Performance_X = \frac{1}{tempo\ de\ execução}$$

Ao compararmos duas máquinas X e Y temos:

$$Performance_X > Performance_Y = \frac{1}{tempo\ de\ exec.X} > \frac{1}{tempo\ de\ exec.Y}$$

$$Tempo\ de\ exec.Y > Tempo\ de\ exec.X$$

A relação de performance será:

$$\frac{Performance_X}{Performance_Y} = N$$

Então dizemos que a máquina X tem uma performance de X vezes a da máquina Y.

Exemplo: Calcule a performance relativa de uma máquina A que roda um programa em 10 segundos com outra máquina B que roda o mesmo programa em 15 segundos. Quantas vezes a máquina A é mais rápida que a B?

Para evitar confusões futuras com os termos “aumentar” e “diminuir”, vamos nos referir ao “aumento da performance” como **melhoria de performance** e **melhora do tempo de resposta** ao invés de diminuição do tempo de resposta.

2.2 Medidas de Performance

O tempo é a medida da performance de um computador, o que realiza a mesma tarefa no menor espaço de tempo é o mais rápido.

O tempo de execução de um processador é medido em segundos, mas o tempo pode ser medido de várias maneiras depende do que estamos medindo.

Ao executar o comando **time** no Unix podemos ter a informação de quanto tempo foi gasto no processamento, o tempo gasto pelo sistema operacional e tempo total decorrido na execução desta tarefa.

```
90.7 u 12.9 s 2:39 65%
```

O tempo de processador (ou tempo de execução pelo processador) é tempo gasto por ele na execução das instruções dedicadas a um programa e não inclui o tempo gasto por dispositivos de I/O nem o tempo gasto na execução de outros programas diferentes que compartilham o mesmo processador.

O tempo total de execução desta tarefa inclui o tempo de acesso aos dispositivos de I/O e de outros programas diferentes.

O tempo de processador pode ainda ser dividido em **tempo de usuário** gasto na execução das instruções do próprio programa e **tempo de sistema** que é o tempo do sistema operacional.

A porcentagem do tempo gasto pelo processador é: $\frac{90,7+12,9}{159} = 0,65$

Para diferenciar a performance, vamos nos referir como **performance de sistema** e **performance de processador**.

Os computadores são conhecidos pela frequência de seus clocks, que é a taxa constante em que o computador determina a ocorrência de seus eventos de hardware. Esta frequência determina os intervalos de tempo, ou ciclos de clock do processador, sendo também conhecido como ticks de clock.

2.3 Relação entre as métricas

Até agora estivemos nos referindo ao tempo de execução gasto pelo processador, mas se quisermos avaliar a performance em função de mudanças em um projeto temos que relacionar algumas métricas.

$$\begin{aligned} \text{Tempo de execução} & \quad \text{Número de ciclos do} \\ \text{no processador para} & \quad \text{clock do processador X} \quad \text{Ciclo do clock} \\ \text{o programa} & \quad \text{para o programa} \\ \text{ou} & \\ \text{Tempo de execução} & \quad \text{Número de ciclos do clock do} \\ \text{no processador para} & \quad \text{processador para o programa} \\ \text{o programa} & \quad \text{Freqüência de clock} \end{aligned}$$

Observamos que ao reduzir o número de ciclos de clock na execução do programa, ou o tamanho do ciclo de clock melhoramos a performance.

Exemplo:

Um programa é executado em 10 segundos em um computador A, que tem um clock de 400 MHz. Um projetista deseja construir uma máquina B que execute o mesmo programa em 6 segundos. Sabe-se que podemos contar com a tecnologia para aumentar de modo significativo a freqüência de clock da máquina, mas que esse aumento pode provocar reflexos em outros parâmetros de performance, fazendo com que B precise de 1,2 vezes mais ciclos que A para executar o programa. Qual a freqüência de clock que o projetista deve implementar para a máquina B?

Até agora não incluímos qualquer referência ao **número de instruções** necessárias à execução de um determinado programa. Mas, o compilador gera instruções que devem ser executadas pelo processador, sendo que o tempo de execução de um dado programa também é função do número de instruções executadas. O **tempo de execução de instruções varia** de instrução para instrução, por exemplo, uma instrução de multiplicação é realizada durante 7 ciclos de clock, enquanto que uma instrução de soma é realizada apenas em um ciclo de clock.

Uma forma de relacionarmos o tempo de execução ao número de instruções executadas é utilizar o **tempo médio execução de cada instrução**. Desta forma, o número de ciclos de clock necessários à execução de um programa é:

$$\text{Número de ciclos de clock do processador} = \frac{\text{Número de instruções de um programa} \times \text{Número médio de ciclos por instrução}}{1}$$

CPI – Ciclos de clock Por Instrução, significa o número médio de ciclos de clock que cada instrução gasta ao ser executada. Como as instruções gastam tempos diferentes para serem executadas, a CPI na verdade é uma média dos tempos gastos por todas as instruções executadas por um programa.

Usando a CPI podemos comparar duas implementações diferentes da mesma arquitetura de instruções.

Exemplo: Suponha duas implementações diferentes da mesma arquitetura do conjunto de instruções. A máquina A tem um ciclo de clock de 1 ns e uma CPI de 2,0 considerando um programa qualquer. A máquina B tem um ciclo de clock de 2 ns e uma CPI de 1,2 para o mesmo programa. Qual das duas máquinas executa esse programa mais rapidamente? Calcule quanto uma é mais rápida que a outra.

Agora podemos então escrever a equação básica da performance em número de instruções executadas pelo processador, em função da CPI e do tempo de ciclo de clock:

$$\text{Tempo de processador} = \frac{\text{Número de instruções} \times \text{CPI} \times \text{Ciclos de clock}}{1}$$

ou

$$\text{Tempo de processador} = \frac{\text{Número de instruções} \times \text{CPI}}{\text{Frequência de clock}}$$

Resumindo, as medidas básicas de performance podem ser em diferentes níveis do computador:

Componentes da performance	Unidades de medida
Tempo de execução de um programa no processador	Segundos para o programa
Número de instruções	Instruções executadas por um programa
Ciclos de clock por instrução (CPI)	Número médio de ciclos de clock por instrução
Ciclo de clock	Segundos por ciclo de clock

$$\text{Tempo} = \frac{\text{Instruções}}{\text{Programa}} \times \frac{\text{Ciclos de clock}}{\text{Instrução}} \times \frac{\text{Segundos}}{\text{ciclo de clock}}$$