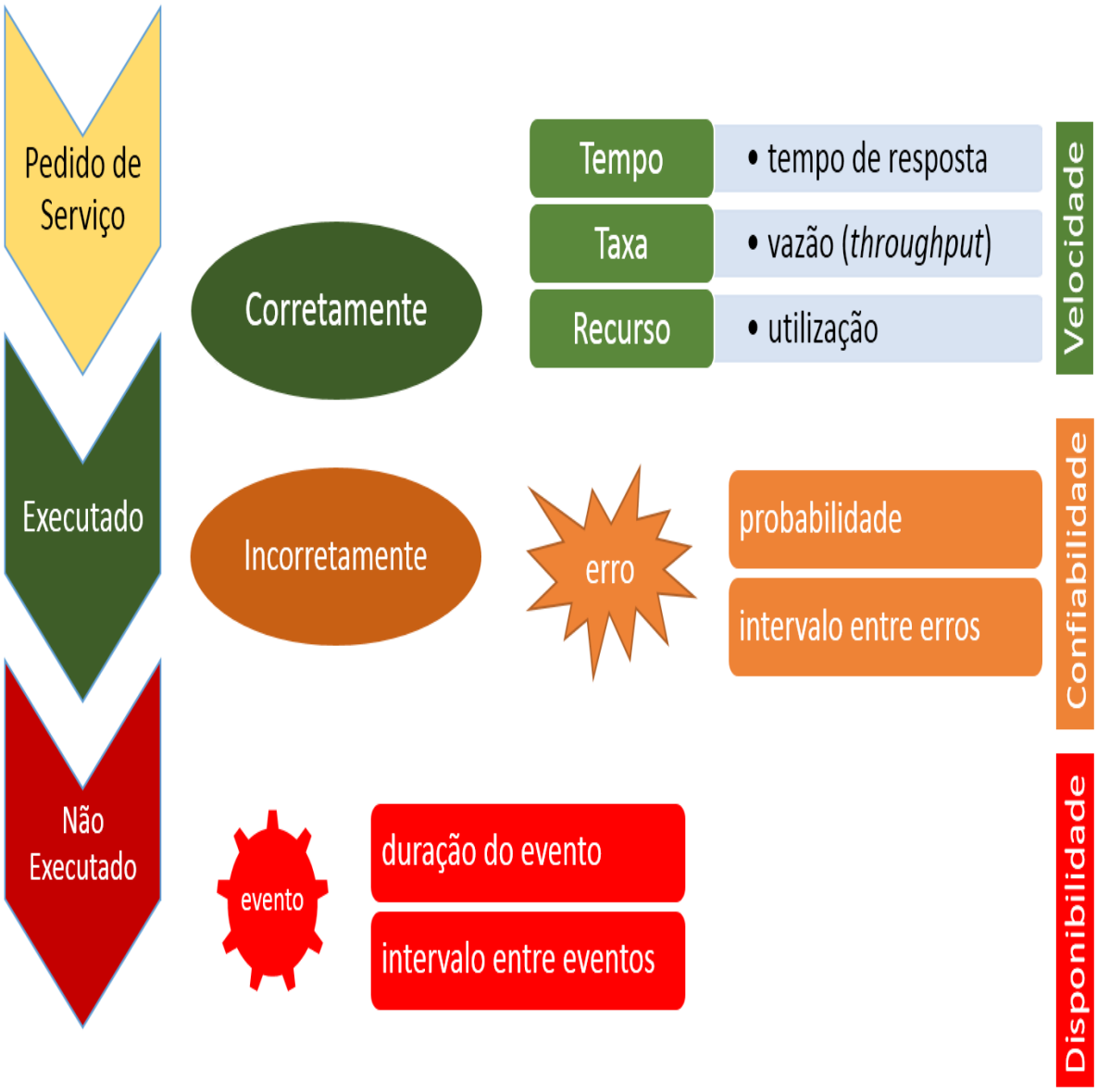
Avaliação de Desempenho de Sistemas



Brauliro Gonçalves Leal

Sobre o autor:
Professor do Colegiado de Engenharia da Computação Universidade Federal do Vale do São Francisco Avenida Antônio Carlos Magalhães, 510 Santo Antônio 48.902-300 Juazeiro - BA - Brasil
e-mail: brauliro.leal@univasf.edu.br
site: www.univasf.edu.br/~brauliro.leal
telefone: 74 2102 7636
Primeira Edição Eletrônica: Junho de 2016
ISBN: a ser feito#

Avaliação de Desempenho de Sistemas

Prefácio

Este livro tem como principal objetivo o de servir de texto para a disciplina de Avaliação de Desempenho de Sistemas do curso de Engenharia da Computação do Colegiado de Engenharia da Computação da Universidade Federal do Vale do São Francisco.

Este texto busca responder as questões:

- O que se entende por Avaliação de Desempenho de Sistemas?
- Que parâmetros são utilizados para quantificá-la e qualificá-la?
- Quais são seus conceitos fundamentais?
- Como a análise de casos particulare podem levar a conclusões mais amplas?
- Quais perguntas centrais deste assunto podem ser respondidas?
- Quais são suas limitações?
- Quais respostas não se pode encontrar a partir destes conceitos?
- Como e quando usar os conceitos da Avaliação de Desempenho de Sistemas?

O autor espera que tenha contribuido com o ensino destes conteúdos de modo a torná-los mais atraentes, visuais, dinâmicos e aplicados nas diversas áreas de conhecimento, quando cabível.

1. Introdução

O desempenho é um critério fundamental na concepção, aquisição e utilização de sistemas computacionais. Por outro lado, obter o melhor desempenho para um determinado custo é um dos objetivos da Engenharia da Computação e, para alcançá-lo, é necessário ao menos um conhecimento básico da terminologia da avaliação de desempenho, seus princípios e suas técnicas.

Os Engenheiros da Computação e demais profissionais da área devem ser capazes de indicar os requisitos de desempenho dos seus sistemas e de comparar diferentes alternativas para encontrar aquela que melhor atenda as necessidades em pauta.

Os Sistemas de Computação envolve software e hardware e, infelizmente, são tão numerosos que não é possível ter uma medida padrão de desempenho, um ambiente de medição padrão (aplicação) ou uma técnica padrão para todos os casos.

A avaliação de desempenho também é uma arte, embora possua conceitos, técnicas, critérios de avaliação e metodologias bem estabelecidas. A utilização adequada destes recursos permite evitar erros comumente observados em projetos de avaliação de desempenho.

A avaliação de desempenho requer as métricas, técnicas e ambiente de medição. A seleção da técnica e da métrica de avaliação são passos essenciais em todos os projetos de avaliação de desempenho. Há muitas considerações que estão envolvidas na seleção correta destes itens, elas são apresentadas a seguir. As métricas de desempenho comumente utilizadas também são definidas abaixo. Finalmente, uma abordagem para o problema de especificar os requisitos de desempenho é apresentada.

1.1. Técnicas de Avaliação de Desempenho de Sistemas

As técnicas de avaliação de desempenho de sistema de computação são simulação, modelagem analítica e medição, Figura 1.1.

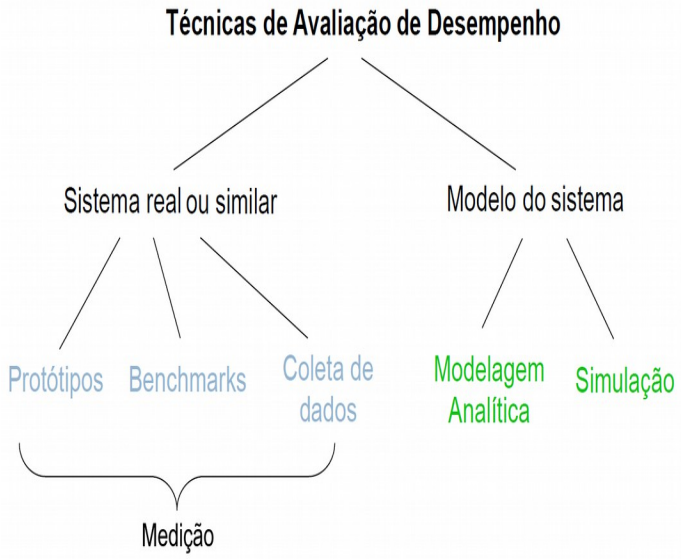


Figura 1.1 - Técnicas de Avaliação de Desempenho.

Há uma série de considerações que ajudam a decidir a técnica a ser utilizada e estão listadas na Tabela 1.1, ordenadas do mais para o menos importante.

Tabela 1.1 - Critérios para seleção de técnicas de avaliação ordenada do mais para o menos importante

Critério	Modelagem Analítica	Simulação	Medição
Fase	qualquer	qualquer	pós-prototipação
Tempo necessário	pequeno	médio	varia
Ferramenta	analistas	linguagem de programação	instrumentação
Precisão	baixo	moderado	varia
Avaliação do *trade-off*[1]	fácil	moderado	difícil
Custo	médio	pequeno	alto
Negociação	média	baixa	alta

A questão fundamental para decidir a técnica de avaliação é a fase do ciclo de vida no qual o sistema se encontra. As medições são possíveis apenas se algo semelhante ao sistema proposto já existe, como na concepção de uma versão melhorada de um produto.

Se for um novo conceito, a modelagem analítica e a simulação são as únicas técnicas possíveis. Modelagem analítica e simulação podem ser usadas nas situações em que a medição não é possível, mas, em geral, é mais convincente para as outras pessoas se a modelagem analítica ou simulação se basearem em medidas.

A próxima consideração é o tempo disponível para a avaliação. Na maioria das situações, os resultados são necessários para ontem. Se este for o caso, a modelagem analítica é provavelmente a opção indicada.

1 *trade-off* - ato de escolher uma coisa em detrimento de outra, e pode ser traduzida como perde/ganha, se relaciona com a análise de benefício/custo.

Simulações podem levar um longo tempo. O tempo necessário para as medições é o mais variável dentre as três técnicas.

A próxima consideração é a disponibilidade de ferramentas. As ferramentas incluem habilidades para modelagem, linguagens de simulação e instrumentos de medição. Muitos analistas de desempenho são hábeis na modelagem e evitam trabalhar em sistemas reais. Outros não, são tão proficientes em teoria das filas que preferem medir ou simular. A falta de conhecimento de linguagens e técnicas de programação faz com que muitos analistas evitem a simulação.

O nível de precisão desejada é outra consideração importante. Em geral, a modelagem analítica exige simplificações e suposições que podem levar a resultados pouco preciso. Simulações podem incorporar mais detalhes e exigem menos hipóteses que a modelagem analítica e, portanto, com mais frequência estão mais próximos da realidade. As medições, embora soem como algo real, também podem dar resultados pouco preciso, simplesmente porque muitos dos parâmetros do ambiente tais como a configuração do sistema, a carga de trabalho e o momento da medição, podem ser exclusivos para o experimento. Além disso, os parâmetros podem não representar o conjunto das variáveis encontradas no mundo real. Assim, a precisão da técnica de medição pode variar significativamente e conduzir a conclusões errôneas.

Modelos analíticos geralmente fornecem o melhor conhecimento sobre os efeitos dos vários parâmetros e suas interações. Com simulação é possível pesquisar o espaço de valores de parâmetro para a combinação ideal mas, muitas vezes, não fica claro o *trade-off* destes parâmetros. A medição é a técnica menos desejável a este respeito. Não é fácil dizer se o melhor desempenho é o resultado de algumas mudanças aleatórias no ambiente ou devido a uma configuração particular do parâmetro.

Os custos do projeto também são importantes. A medição requer um equipamento real, instrumentos e tempo sendo a mais cara das três técnicas. Custos, junto com a facilidade de poder alterar as configurações é, em muitos casos, a razão para que o desenvolvimento de simulações de sistemas sejam mais baratos.

A negociação dos resultados é, provavelmente, a justificativa essencial quando se consideram as despesas e o trabalho de medições. É muito mais fácil convencer os outros utilizando medição real. A maioria das pessoas fica cética diante de resultados analíticos, simplesmente porque não entendem a técnica ou o resultado final. Na verdade, as pessoas que desenvolvem novas técnicas de modelagem analítica muitas vezes utilizam simulações ou medições reais para validá-las.

Às vezes é útil usar duas ou mais técnicas simultaneamente. Por exemplo, pode-se usar a simulação e modelagem analítica em conjunto para verificar e validar os resultados de cada um. Isso nos leva às seguintes três regras de validação:

1. não confie nos resultados de um modelo de simulação até que eles tenham sido validados por modelos analíticos ou de medições
2. não confie nos resultados de um modelo analítico antes de terem sido validadas por um modelo de simulação ou de medições
3. não confie nos resultados de uma medição até que eles tenham sido validados por simulação ou modelagem analítica.

Em particular, a necessidade da terceira regra sobre a validação dos resultados das medições deve ser enfatizada. Este é a mais comumente ignorada das três regras. As medições são tão suscetíveis a erros experimentais e *bugs* quanto as outras duas técnicas.

O único requisito para validação é que os resultados não devem ser contra o bom senso. Este método de validação, chamado intuição do especialista ou perito, é comumente usado para modelos de simulação. Este e outros métodos de validação podem ser utilizados para a medição e análise dos resultados.

Duas ou mais técnicas também podem ser usadas sequencialmente. Por exemplo, um modelo analítico simples é usado para encontrar o intervalo adequado para os parâmetros do sistema e uma simulação é utilizada mais tarde para estudar o desempenho nesse intervalo. Isso reduz o número de simulação e pode resultar em um uso mais produtivo dos recursos.

1.2. Métricas de Avaliação de Desempenho de Sistemas

A medida de desempenho de um sistema de computação depende da capacidade, velocidade e compatibilidade de seus diferentes componentes. Para atender a combinação destes fatores e os diferentes componentes dos sistemas de computação foram desenvolvidos vários meios de medir desempenho. De modo geral, as medidas de desempenho são taxas, fluxos ou medidas temporais. Pode-se medir o desempenho do sistema como um todo ou de seus componentes isoladamente ou em conjuntos.

Para cada estudo de desempenho de um sistema, um conjunto de critérios de desempenho e métricas deve ser escolhido. Uma maneira de identificá-los é fazer uma lista dos serviços oferecidos pelo sistema. Há vários resultados possíveis para cada solicitação de serviço feitas ao sistema. Geralmente, estes resultados podem ser classificados em três categorias, conforme mostrado na Figura 1.2.

Figura 1.2 - Três possíveis resultados de uma solicitação de serviço.

O sistema pode executar o serviço corretamente, incorretamente, ou não realizar o serviço. Por exemplo, considerando um *gateway* em uma rede de computadores que oferece o serviço de encaminhamento de pacotes para os destinos especificados em redes heterogêneas. Quando ele recebe um pacote para ser enviado, o serviço pode ser executado:

1. corretamente, sucesso
2. incorretamente, fracasso parcial
3. não ser executado, fracasso total

Quando ele recebe um pacote, ele pode recebê-lo corretamente, ele pode recebê-lo de forma incorreta ou pode não recebê-lo. As métricas associadas aos três resultados, ou seja, serviço bem-sucedido, com erro e não executado, são também chamadas métricas de velocidade, confiabilidade e disponibilidade.

A maioria dos sistemas oferece mais de um serviço e, portanto, o número de métricas cresce proporcionalmente uma vez que cada um destes serviços possui uma série de métricas de velocidade, uma série de métricas de confiabilidade e uma série de métricas de disponibilidade.

O recurso com a maior utilização é chamado de gargalo. Melhorar o desempenho deste recurso oferece o maior retorno. Conhecer a utilização dos vários recursos do sistema é uma parte importante da avaliação de desempenho.

Uma única falha pode comprometer todo o funcionamento do sistema. Para evitar essa situação, há os sistemas tolerantes a faltas, mas isso pode afetar seu tempo de serviço e dificultar seu projeto, desenvolvimento e manutenção. Uma outra maneira de contornar as falhas é adicionar redundâncias com o inconveniente de aumentar o consumo de energia e a complexidade de gerenciamento. Ao final, os custos associados com a confiabilidade são, em última instância, transferidos para os usuários finais.

1.2.1. Métricas de Velocidade

Se o sistema executa o serviço corretamente, seu desempenho é medido pelo tempo necessário para executar o serviço, a taxa na qual o serviço é realizada e os recursos consumidos durante a execução do serviço. Estas três medidas relacionadas ao desempenho bem-sucedido do serviço, o tempo, a taxa e os recursos utilizados, são também chamadas tempo de resposta, vazão e utilização, respectivamente.

Por exemplo, a resposta de um *gateway* de rede é medida por seu tempo de resposta, o intervalo de tempo entre a chegada de um pacote e a sua entrega bem-sucedida. A vazão do sistema é medida pelo número de transferências, pacotes transmitidos ou recebidos, por unidade de tempo. Este valor dá uma indicação da percentagem de tempo que os recursos do *gateway* estão sendo usados para o nível de carga, que é a utilização.

1.2.1.1. Tempo de Resposta

O tempo de resposta (*turn-around time*) é definido como o intervalo entre o instante em que a tarefa é submetida ao sistema e o momento em que produz a saída completa e a tarefa se encerra. É uma métrica interessante por representar a performance do ponto de vista do usuário, que submete sua tarefa e espera algum tempo até que ela seja executada.

Os valores médios do tempo de resposta podem ser influenciados pelos seus valores extremos. Tarefas com tempo de respostas pequenos e outras com tempo de resposta muito longos, podem ter seus valores médios afetados por estes valores ou muito grandes ou muito pequenos.

O tempo de resposta médio deve ser calculado por meio da média geométrica que tem a característica de ser menos influenciada por valores extremos. O uso da média geométrica, ao invés da média aritmética, tem o objetivo de reduzir o efeito dos valores extremos sobre a média geral do tempo médios da tarefa.

O tempo de resposta é definido como o intervalo entre o pedido do usuário e a resposta do sistema, como mostrado na Figura 1.3a.

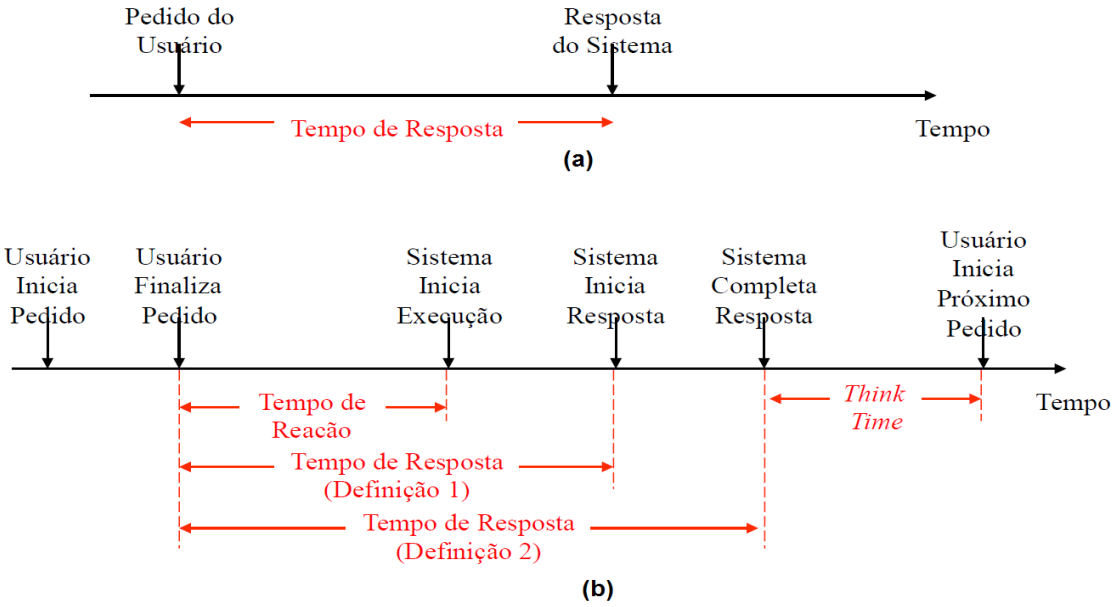


Figura 1.3 - Definições de tempo de resposta: a) pedidos e respostas instantâneos; e b) pedidos e respostas realistas.

A definição apresentada na Figura 1.3a, entretanto, é simplista, uma vez que os pedidos, bem como as respostas, não são instantâneos. Os usuários passam tempo digitando o pedido e o sistema leva tempo para exibir a resposta, como mostrado na Figura 1.3b.

Há duas possíveis definições do tempo de resposta neste caso:

- **Definição 1** - intervalo entre o fim de uma apresentação do pedido e o início da resposta correspondente do sistema Figura 1.3a
- **Definição 2** - intervalo entre o fim de uma apresentação do pedido e o final da correspondente resposta do sistema Figura 1.3b

A segunda definição é preferível se o tempo entre o início e o final da resposta é longo, neste caso, o tempo de resposta para usuários interativos em um sistema de tempo compartilhado seria o intervalo entre o pedido (ao teclar enter) e último caractere da resposta do sistema.

Para tarefas em lote, a resposta é medida pelo tempo de execução (*turn-around time*), que é o intervalo de tempo entre a apresentação do lote de tarefas e a sua conclusão. Observe que o tempo para ler a entrada está incluído no tempo de execução.

O tempo entre a apresentação de uma tarefa e o início de sua execução pelo sistema é chamado de tempo de reação (*reaction time*). Para medir o tempo de reação, é necessário controlar as ações internas do sistema desde a solicitação do pedido até o início da execução, pois pode não haver nenhum evento externo visível para indicá-lo. Por exemplo, em sistemas de tempo compartilhado, o intervalo entre o último toque do usuário no teclado e a entrada do primeiro byte do processo na CPU seria

chamado tempo de reação.

O tempo de resposta de um sistema geralmente aumenta à medida que a carga do sistema aumenta. A proporção de tempo de resposta para uma determinada carga e a carga mínima do sistema é chamada fator de carga (*stretch factor*). Para um sistema de tempo compartilhado, por exemplo, o fator de carga é definido como a razão entre o tempo de resposta com e sem a multiprogramação.

Os conceitos discutidos acima se aplicam igualmente a sistemas de hardware, especificamente entre comunicação entre máquinas (M2M - *Machine to Machine*).

1.2.1.2. *Vazão ou Throughput*

Vazão ou *throughput* é definida como a taxa temporal em que as tarefas são atendidas pelo sistema. Para o processamento em lotes, a vazão é medida em tarefas (*jobs*) por segundo. Para sistemas interativos, a vazão é medida em solicitações por segundo. Para CPUs, a vazão ou rendimento, é medida em milhões de instruções por segundo (MIPS), ou milhões de operações de ponto flutuante por segundo (MFLOPS). Para as redes, a vazão ou rendimento é medida em pacotes por segundo (pps) ou bits por segundo (bps). Para sistemas de processamento de transações a vazão ou rendimento é medida em transações por segundo (TPS). Em geral, cada sistema de computação tem sua unidade de vazão.

No início, a vazão de um sistema geralmente aumenta à medida que a carga do sistema aumenta. Depois de uma determinada carga, a taxa de transferência para de aumentar, na maioria dos casos, pode até começar a diminuir, como mostrado na Figura 1.4. A taxa de transferência máxima possível sob condições ideais de trabalho é chamada de capacidade nominal (*nominal capacity*) do sistema. Para as redes de computadores, a capacidade nominal é chamada de largura de banda (*bandwidth*) e é normalmente expressa em bits por segundo.

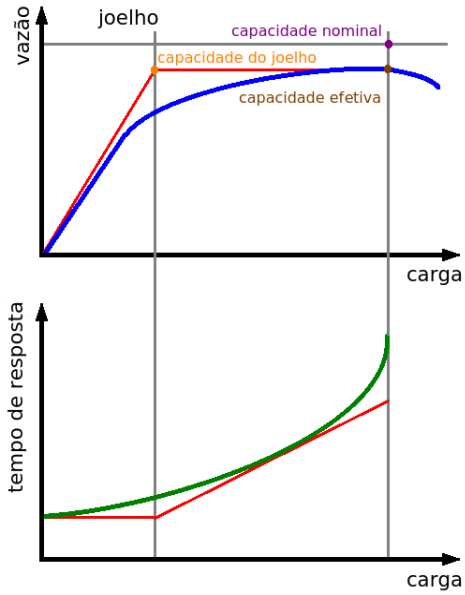


Figura 1.4 - Capacidade de um sistema de computação.

Muitas vezes o tempo de resposta na taxa de transferência máxima é demasiado elevado para ser aceitável. Nesses casos, é mais interessante saber a taxa de transferência máxima possível, sem exceder o limite de tempo de resposta pré-especificado. Isso pode ser chamado de capacidade útil do sistema (*usable capacity*). Em muitas aplicações, o ponto de inflexão da curva de tempo de resposta é considerado o ponto de funcionamento ótimo, este é o ponto além do qual o tempo de resposta aumenta rapidamente em função da carga, mas o ganho em vazão é pequeno.

Antes do ponto de inflexão, o tempo de resposta não aumenta significativamente, mas a vazão aumenta com o aumento da carga. A taxa correspondente ao ponto de inflexão é chamada capacidade de inflexão do sistema (*knee capacity*).

Também é comum medir a capacidade em termos de carga, por exemplo, o número de usuários, ao invés da taxa de transferência. Mais uma vez, é importante definir com precisão as métricas e suas unidades antes de usá-las em um projeto de avaliação de desempenho.

A relação entre taxa máxima alcançável (capacidade utilizável) e a capacidade nominal é chamada eficiência (*efficiency*). Por exemplo, se a taxa de transferência máxima é igual a 100 Mbps e a capacidade utilizável de uma LAN (*Local Area Network*) é de apenas 85 Mbps, sua eficiência é de 85%. O termo eficiência também é usado para sistemas com múltiplos processadores. A razão entre o desempenho de um processador em relação aos n-processadores do sistema é a sua eficiência, como mostrado na Figura 1.5. O desempenho é geralmente medido em termos de MIPS ou MFLOPS.

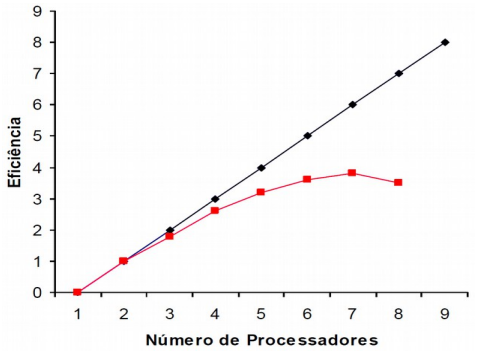


Figura 1.5 - Eficiência de um sistema multiprocessador (cor vermelha) e a capacidade utilizável (cor preta) versus número de processadores.

1.2.1.3.Utilização

A utilização (*utilization*) de um recurso é medida como a fração do tempo utilizado pelo recurso em relação às demais solicitações de serviço. Por exemplo, a razão entre o tempo ocupado e tempo total durante um determinado período. O período durante o qual o recurso não está sendo utilizado é chamado de tempo ocioso (*idle time*).

Os gestores de sistemas devem buscar equilibrar a carga de modo que não se utilize um recurso mais que outros. Naturalmente, nem sempre isso é possível. Alguns recursos, como processadores, estão sempre ocupados ou inativos, por isso a sua utilização em termos de percentagem de tempo ocupado e o tempo total faz sentido. Para outros recursos, como memória, apenas uma fração dos recursos podem ser utilizados em um determinado momento, a sua utilização é medido como a fração média utilizada durante um intervalo.

1.2.2. Confiabilidade

Se o sistema executa o serviço incorretamente, é dito ter ocorrido um erro (neste contexto erro, falha e defeito são considerados equivalentes). Neste caso, a confiabilidade é uma medida do número de interrupções críticas durante o tempo em que um sistema está em funcionamento. É útil classificar os erros e determinar as probabilidades para cada classe de erros.

Confiabilidade é a probabilidade de um sistema desempenhar sua funções em dado período de tempo sob determinadas condições.

Desta forma, define-se confiabilidade, R(t), como a probabilidade de que o sistema opere sem falhas por um período de tempo t, ou seja, $R(t) = P(X > t)$, Equação 1.1.

$$R(t) = 1 - F(t) = 1 - \int_{t}^{\infty} f(u)\,du \qquad 1.1$$

Em que F(t) é a probabilidade de que o sistema falhe antes do tempo t e f(x) é a fdp dos tempos de falha.

Os modelos que fornecem a estimativa de taxa de falha de um sistema função do tempo são baseados em distribuições estatísticas e as mais utilizadas são as funções Exponencial, Weibull e Lognormal.

Por exemplo, no caso do *gateway*, que pode-se buscar encontrar a probabilidade de erros de um único bit, os erros de dois bits, e assim por diante. Pode-se também querer encontrar a probabilidade de um pacote ser entregue parcialmente (fragmento).

A confiabilidade (*reliability*) de um sistema geralmente é medida pela probabilidade de erros ou o tempo médio entre erros. Este último é frequentemente especificado como segundos sem erros (*error-free seconds*).

1.2.3. Disponibilidade

A disponibilidade (*availability*) de um sistema é definida como a fração do tempo que o sistema está disponível para atender às solicitações dos usuários. O tempo durante o qual o sistema não está disponível é o chamado tempo de inatividade (*downtime*), o tempo durante o qual o sistema está disponível é chamado de *uptime*. Muitas vezes, o *uptime* médio, mais conhecido como o tempo médio até falhar (MTTF – *Mean Time To Failure*) é o melhor indicador uma vez que a combinação de pequenos *downtime* e *uptime* pode resultar em uma medida de alta disponibilidade, mas os usuários podem não ser capazes de obter qualquer serviço se o tempo de funcionamento é inferior ao tempo necessário para completar o serviço.

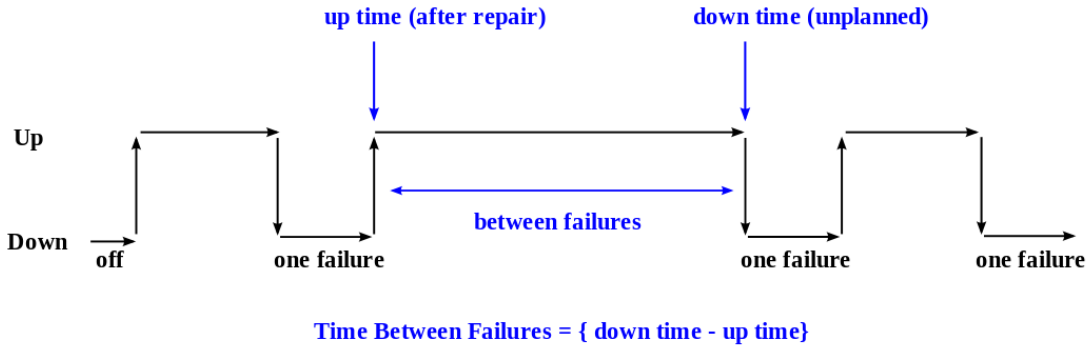


Figura 1.6 - Representação da disponibilidade de um sistema destacando a aorrencia de falha, sua reparação e o tempo entre falhas (fonte: https://en.wikipedia.org/wiki/Mean_time_between_failures)

$$disponibilidade = \frac{\sum uptime}{tempo\,total} \quad (1.2)$$

$$MTBF = \frac{\sum (início\,do\,downtime - início\,do\,uptime)}{número\,de\,falhas} \quad (1.3)$$

Se o sistema não executa o serviço, ele pode estar falhando ou indisponível. Mais uma vez, é útil classificar os modos de falha e determinar as probabilidades destas ocorrências. Por exemplo, o *gateway* pode não estar disponível em 0,01% do tempo devido a uma falha do processador e em 0,03% devido a uma falha de software.

1.2.4. Métricas de Confiabilidade e Disponibilidade

As métricas de confiabilidade e disponibilidade estão relacionadas e fazem parte de um um conceito mais amplo que é a confiança de sistemas computacionais (assunto que está além deste texto).

Dado um período de funcionamento de um sistema computacional T, seja em T_{up} o tempo total de operação sem falhas, T_{down} o tempo total do sistema parado devido a falhas, t_i o tempo de ocorrência da i-ésima falha ($i \in [1,NF]$) e NF o número total de falhas no período T, NF > 0. Os valores de MTBF, MTTR e da disponibilidade são dados pelas Equações 1.5, 1.6, 1.7, respectivamente.

$$MTTF = \frac{\sum t_i}{NF} \quad (1.4)$$

$$MTBF = \frac{T_{up}}{NF} \quad (1.5)$$

$$MTTR = \frac{T - T_{up}}{NF} = \frac{T_{down}}{NF} \quad (1.6)$$

$$disponibilidade = \frac{MTBF}{MTBF + MTTR} \quad (1.7)$$

$$R(MTTF) = 1 - \int_{MTTF}^{\infty} f(x)\,dx \quad (1.8)$$

Se não há falha (NF = 0) então estes indicadores são indefinidos e o sistema é tolerante a falhas no período.

Exemplo 1.1 - Um sistema foi projetado para operar corretamente durante 100 horas. Durante esse período foram observadas 9 falhas que somadas duraram 300 minutos (5 horas).

Calculando o valor de MTBF, tem-se: $MTBF = \frac{100-5}{9} = 10{,}56\,h$

Este valor indica que a cada 10 horas e 34 minutos poderá ocorrer uma falha no sistema, deixando-o indisponível.

Calculando o valor de MTTR, tem-se: $MTTR = \frac{5}{9} = 0{,}56\,h$

Este valor indica que a duração média de uma falha é 34 minutos.

Como consequência, a cada 10 horas e 34 minutos o sistema poderá ficar indisponível por 34 minutos.

Calculando a disponibilidade, tem-se: $disponibilidade = \frac{10{,}56}{10{,}56+0{,}56} = 94{,}96 = 94{,}96\,\%$

Este valor indica que a disponibilidade do sistema é igual 94,96%.

Os MTBF e MTTR são métricas HB e LB, respectivamente. São indicadores de desempenho usados há mais de 60 anos e podem ser utilizados para avaliar processos, a produtividade e a qualidade dos sistemas computacionais, também servem para estimar e dimensionar o esforco de manutenção corretiva ou preventiva.

Exemplo 1.2 - Um sistema apresentou falhas nos tempos 12, 19, 28, 42 e 72 h apoś iniciar seus trabalhos. Calcule a confiabilidade deste sistema neste período. Considere a distribuição exponencial como modelo para a estimativa da taxa de falha deste sistema.

Calculando o valor de MTTF, tem-se: $MTTF = \frac{12+19+28+42+72}{5} = 34{,}6\,h$

Como $f(x) = \lambda e^{-\lambda x}$, com $E(x) = 1/\lambda$ e $\lambda = 1/MTTF = 0{,}028902$.

A confiabilidade do sistema é dado por: $R = 1 - \int_{34{,}6}^{\infty} 0{,}028902\, e^{-0{,}028902\,x}\, dx = 0{,}6321$

Este valor indica que a confiabilidade do sistema é de 63,21% no período de tempo considerado e seu médio para falhar é igual a 34 horas 36 minutos.

1.2.5. Outras Métricas

Nos estudos de aquisição do sistema, a relação custo/desempenho (*cost/performance ratio*) é comumente usada como uma métrica para comparar dois ou mais sistemas. O custo inclui o custo de hardware e licenciamento de software, instalação e manutenção ao longo de um determinado número de anos. O desempenho é medido em termos de *throughput* em uma restrição de tempo de resposta dada. Por exemplo, dois sistemas de processamento de transações podem ser comparados em termos de dólares por TPS.

Em muitos sistemas, serviços requerem o envio de mensagens através de

uma rede de interconexão. A latência é o tempo necessário para enviar mensagem através de uma rede de interconexão, inclui o tempo de empacotar e desempacotar dados além do tempo de envio propriamente dito. A latência aumenta a medida que a quantidade de dados a serem enviados aumenta. Este aumento não é linear uma vez que pode ocorrer aumento e, ou, redução do tamanho dos pacotes, forçando o desempacotamento e reempacotamento, perdas de pacotes, reenvio de pacotes, confirmações de recebimento, enfim, tráfego de dados de controle. A componente do tempo referente ao custo de empacotamento e desempacotamento não varia tanto em relação ao tamanho da mensagem como a componente de custo de envio pela rede, conforme já discutido.

As mensagens enviadas através de uma rede de interconexão utilizam protocolos de comunicação que incluem dados de controle para o fluxo da mensagem através da rede. A eficiência de um protocolos de comunicação é a razão entre o tamanho da mensagem e a quantidade total de dados (mensagem+controle) transmitida.

1.3. Caracterização das Métricas de Desempenho

Para muitas métricas, o valor médio é o mais importante. No entanto, não se deve negligenciar o efeito da variabilidade. Por exemplo, um tempo de resposta alto, de um sistema de tempo compartilhado, assim como uma alta variabilidade do tempo de resposta, pode degradar significativamente a produtividade. Se este for o caso, é necessário estudar essas duas métricas.

Nos sistemas de computação compartilhada por muitos usuários, dois tipos de métricas de desempenho devem ser considerados: individual e global. As métricas individuais refletem o uso de cada usuário, enquanto a métrica global reflete o uso de todo o sistema. A utilização dos recursos, confiabilidade e disponibilidade são as métricas globais, enquanto o tempo de resposta e *throughput* pode ser medida para cada indivíduo, bem como a nível global para o sistema. Há casos em que a decisão que otimiza as métricas do indivíduo é diferente da que otimiza o sistema global.

Por exemplo, em redes de computadores, o desempenho é medido pela taxa de transferência (pacotes por segundo). Em um sistema onde o número total de pacotes permitidos na rede é mantido constante, aumentando o número de pacotes de uma fonte pode levar ao aumento do seu rendimento, mas também pode diminuir a taxa de transferência de outra pessoa. Assim, tanto o rendimento de todo o sistema e sua distribuição entre usuários individuais deve ser estudado. Utilizando apenas o *throughput* do sistema ou a transferência individual pode conduzir a situações injustas.

Um subconjunto de métricas de desempenho, elas devem se completar com baixa variabilidade e evitando redundância. Na Tabela 1.2 estão as

considerações usadas para selecionar um subconjunto de métricas de desempenho.

Tabela 1.2 - Critérios para selecionar métricas de desempenho de sistemas computacionais

baixa variabilidade	baixa variabilidade ajuda a reduzir o número de repetições necessárias para obter um determinado nível de confiança estatística. Métricas que utilizam duas variáveis geralmente possuem uma maior variabilidade do que qualquer das duas tomadas individualmente e devem ser evitadas, se possível
evitar redundância	se duas métricas estimam essencialmente a mesma informação, é menos confuso para o estudo utilizar apenas uma. Isso nem sempre é evidente, no entanto. Por exemplo, em redes de computadores, o tempo médio de espera em uma fila é igual ao quociente entre o comprimento médio da fila e a taxa de chegada. Estudando o comprimento médio da fila e o tempo de espera médio pode não prover quaisquer *insights* adicionais
completude	o conjunto de métricas incluídas no estudo deve ser completo. Todos os resultados possíveis devem ser refletidos no conjunto de métricas de desempenho. Por exemplo, em um estudo comparando diferentes protocolos em uma rede de computadores, um protocolo foi escolhido como o melhor até que foi descoberto que o melhor protocolo levou a um maior número de desligamentos de circuito prematuramente. A probabilidade de desconexão foi então adicionada ao conjunto de métricas de desempenho

A classe de utilidade de uma métrica é útil para apresentação de dados. Dependendo da função de utilidade de uma métrica de desempenho, ela pode ser classificada em três classes, descritos na Tabela 1.3 e a Figura 1.7 mostra gráficos hipotéticos de utilidade de três classes de métricas.

Tabela 1.3 - Classes de métricas de desempenho de sistemas computacionais

menor é melhor ou LB (*lower is better*)	os usuários do sistema e administradores de sistemas preferem os menores valores de tais métricas. O tempo de resposta é um exemplo de uma métrica LB
nominal é melhor ou NB (*nominal is best*)	ambos os valores altos e baixos são indesejáveis. Um valor especial, intermediário entre os extremos, é considerado o melhor. A taxa de utilização é um exemplo de uma característica NB
maior é melhor ou HB (*higher is better*)	os usuários do sistema e administradores de sistemas preferem os maiores valores de tais métricas. *Throughput* do sistema é um exemplo de uma métrica HB

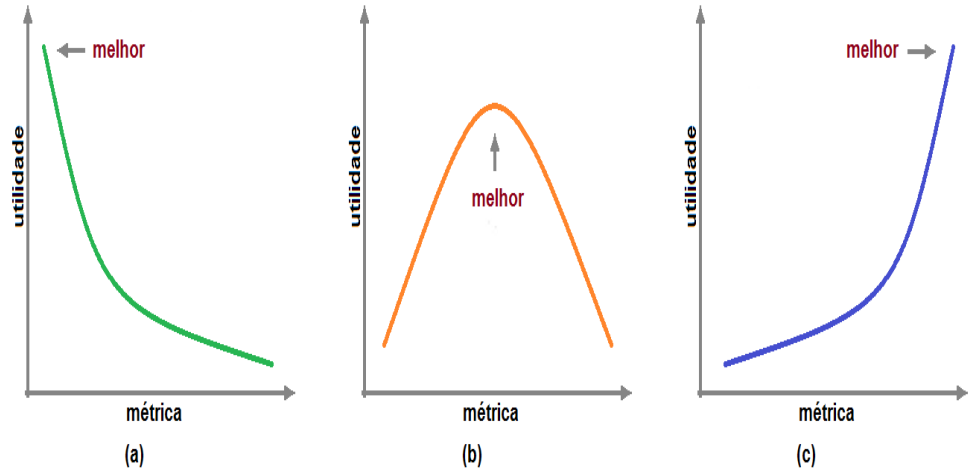


Figura 1.7 - Tipos de métricas: a) menor é melhor (LB); b) nominal é melhor (NB); e c) maior é melhor (HB).

As métricas de desempenho também devem ser específicas, mensuráveis, viáveis e possuir eficácia, conforme é descrito na Tabela 1.4. Em conjunto, as métricas de desempenho devem possuir tanto características quantitativas quanto qualitativas.

Tabela 1.4 - Características desejáveis das métricas de desempenho de sistemas computacionais

especificidade	impede o uso de palavras como baixa probabilidade
mensurabilidade	exige a verificação de que um determinado sistema atenda aos requisitos
factibilidade ou aceitabilidade	demandam limites de exigências de configuração ou decisões de arquitetura alto o suficiente para ser aceitável e baixo o suficiente para ser viável
eficácia	estabelece que as exigências devem ser definidas para todos os resultados possíveis incluindo seus modos de falha

Considere a afirmação de exigência típica:

O sistema deve ser eficiente tanto em termos de processamento quanto de memória. Não deve criar sobrecarga excessiva. Com probabilidade extremamente baixas de que a rede duplicará um pacote, entregar um pacote para o destino errado ou alterar os dados em um pacote.

Estas declarações são exigências inaceitáveis uma vez que sofrem de um ou mais dos seguintes problemas:

Tabela 1.5 - Características indesejáveis para requisitos de métricas de desempenho de sistemas computacionais

inespecíficos	não são especificados em números claros. Palavras qualitativas tais como baixo, alto, raros e extremamente pequenos são usadas
não mensurável	não há maneira de estimar medidas e verificar se elas atendem ao requisito
inaceitável	os valores numéricos de requisitos, se especificados, são definidas com base no que pode ser alcançado ou o que parece ser bom. Se for

	feita uma tentativa para definir os requisitos de forma realista, eles acabam por ser tão baixos que se tornam inaceitáveis
não realizável	muitas vezes, os requisitos são definidos tão alto que parece bom. No entanto, tais requisitos não podem ser reais
incompletas	nenhuma tentativa é feita para especificar um dos resultados possíveis

Estes problemas podem ser resumido em uma palavra: SMART, ou seja, os requisitos devem ser específicos, mensuráveis, aceitáveis, realizávéis e eficazes (Specific, Measurable, Acceptable, Realizable and Thorough Specificity).

Exemplo 1.3 - Considere o problema de especificar os requisitos de desempenho para um sistema de rede de alta velocidade, LAN (*Local Area Network*), descrito por Jain (1991). A LAN, basicamente, presta o serviço de transporte de pacotes para a estação de destino especificada. Atendendo uma solicitação do usuário para enviar um pacote para o destino da estação D, existem três categorias de resultados:

a) o pacote é entregue corretamente para D
b) entregue incorretamente (entregue a um destino errado ou com indicação de erro para D)
c) não entregue

Os requisitos de desempenho para estas três categorias de resultados foram os seguintes:

a) Velocidade: se o pacote é entregue corretamente, o tempo necessário para entregar e a taxa em que é entregue são importantes. Isso leva às seguintes exigências:

a) o atraso no acesso em qualquer estação deve ser inferior a 1 s
b) a produção sustentada deve ser pelo menos 80 Mb/s

b) Confiabilidade: cinco modos diferentes de erro foram considerados importantes. Cada um destes modos de erro são expressos por quantidades diferente de erros e, portanto, tem diferentes níveis de aceitabilidade. Os requisitos de probabilidade para cada um desses modos de erro e seu efeito combinado, são os seguintes:

a) a probabilidade de um bit estar errado deve ser inferior a 1.0×10^{-7}
b) a probabilidade de entrega de pacote com erro com indicação de erro definida deve ser inferior a 1%
c) a probabilidade de entrega de pacote com erro sem indicação de erro deve ser inferior a 1.0×10^{-15}
d) a probabilidade de entrega de pacote com erro detectado no endereço de destino deve ser inferior a 1.0×10^{-18}
e) a probabilidade de entrega de pacote ser entregue mais de uma vez (duplicado) deve ser inferior a 1.0×10^{-5}
f) a probabilidade de perda de pacote na LAN, devido a todos os tipos de erros, deve ser inferior a 1%

c) Disponibilidade: dois modos de falha foram considerados significativos. O primeiro

foi o tempo perdido devido à re-inicializações da rede, e o segundo foi o tempo perdido devido às falhas permanentes que exigem chamadas de serviço de manutenção. Os requisitos para a frequência e duração destes modos de falha foram especificadas da seguinte forma:

a) o tempo médio para inicializar a LAN deve ser inferior a 15 ms
b) o tempo médio entre inicializações da LAN devem ser de pelo menos um minuto
c) o tempo médio para reparar a LAN deve ser inferior à uma hora (partições LAN podem ser operacionais, durante esse período)
d) o tempo médio entre o particionamento da LAN deve ser de pelo menos metade de uma semana

Todos os valores numéricos acima referidos foram verificados para a factibilidade utilizando modelagem analítica, que mostrou que os sistemas de LAN que satisfazem estes requisitos são viáveis.

Exemplo 1.4 - Considere o problema de comparar dois algoritmos diferentes de controle de congestionamento em redes de computadores. Uma rede de computadores consiste de um conjunto de sistemas finais interligados através de uma série de sistemas intermediários. Os sistemas finais enviam pacotes para sistemas na extremidade da rede. Os sistemas intermediários encaminhar os pacotes ao longo do caminho. O problema de congestionamento ocorre quando o número de pacotes esperando por um sistema intermediário é superior a capacidade de transferência do sistema e alguns dos pacotes devem ser descartados.

O sistema, neste caso consiste na rede e o único serviço em questão é o encaminhamento de pacotes. Quando um usuário de rede envia um bloco de pacotes para outra estação terminal chamado destino, há quatro resultados possíveis:

a) alguns pacotes são entregues para o destino corretamente
b) alguns pacotes são entregues fora de ordem para o destino
c) alguns pacotes são entregues mais de uma vez para o destino (pacotes duplicados)
d) alguns pacotes são descartados no caminho (pacotes perdidos)

Para os pacotes entregues em ordem, a aplicação direta das métricas em tempo taxa de recursos produz a seguinte lista:

a) tempo de resposta: o atraso dentro da rede de pacotes individuais
b) *throughput* ou vazão: o número de pacotes por unidade de tempo
c) processamento da fonte: tempo por pacotes no sistema
d) processamento do destino final: tempo por pacote nos sistemas de destino
e) processamento intermediário: tempo por pacote nos sistemas intermediários

O tempo de resposta determina o tempo que um pacote tem que ser mantido da fonte até a estação final usando seus recursos de memória, menor tempo de resposta é considerado melhor. O *throughput* é o desempenho como visto pelo usuário, aumentar a taxa de transferência é considerado melhor.

A variabilidade do tempo de resposta também é importante uma vez que os resultados de resposta variam muito em função de retransmissões desnecessárias. Assim, a variação do tempo de resposta tornou-se uma outra métrica, a sexta.

Pacotes fora de ordem são indesejáveis, uma vez que geralmente não podem ser entregues para o usuário imediatamente. Em muitos sistemas, pacotes fora de ordem são descartados nos sistemas de destino final. Em outros, eles são armazenados no sistema de *buffers* aguardando a chegada de demais pacotes. Em ambos os casos,

chegadas de pacotes fora de ordem podem causar sobrecarga adicional. Assim, a probabilidade de chegadas fora de ordem foi a sétima métrica.

Pacotes duplicados consumem recursos de rede. A probabilidade de pacotes duplicados, por conseguinte, é a oitava métrica.

Pacotes perdidos são indesejáveis por razões óbvias. A probabilidade de perda de pacotes é a nona métrica.

Perdas excessivas de retransmissões podem causar queda nas conexões do usuário, assim a probabilidade de desconexão foi adicionada como a décima métrica.

A rede é um sistema multiusuário. É necessário que todos os usuários sejam tratados de forma justa. Portanto, a equidade foi adicionada como a décima primeira métrica. Ela é definida como uma função da variabilidade da vazão entre os usuários. Para um dado conjunto de usuário $\{x_1, x_2, ..., x_n\}$, a seguinte função pode ser usada para atribuir um índice de equidade para o conjunto:

$$f(x_1, x_2, \cdots, x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2} \qquad 1.9$$

O índice de equidade se situa entre 0 e 1. Se todos os usuários recebem rendimento igual, o índice é igual a 1. Se k dentre os n usuários receberem rendimentos iguais e os restantes n-k usuários receberem rendimento zero, o índice de equidade é k/n.

Após alguns experimentos, ficou claro que a vazão e os atrasos são métricas redundantes. Todos os esquemas que resultou em uma maior vazão também resultou em maior atraso. Assim, as duas métricas foram retiradas da lista e, substituídas por uma métrica que combina as duas, definida como a razão do throughput e o tempo de resposta.

A variação no tempo de resposta também foi descartada, uma vez que era redundante com a probabilidade de duplicação e a probabilidade de desconexão. A maior variação resultou em uma maior probabilidade de duplicação de esforços e uma maior probabilidade de desconexão prematura. Assim, neste estudo, um conjunto de nove indicadores foi usado para comparar diferentes algoritmos de controle de congestionamento.

1.4. Erros comuns

Contrariamente à crença comum, como já foi comentado, a avaliação de desempenho é uma arte. Como uma obra de arte, uma avaliação de sucesso não pode ser produzida mecanicamente. Cada avaliação exige um conhecimento adequado do sistema a ser modelado e uma cuidadosa seleção da métrica, da carga e das ferramentas. Definir o problema real e estabelecer as ferramentas e técnicas a serem utilizadas, sem perder de vista o tempo e outras restrições, é uma parte importante da arte do Engenheiro da Computação.

Exemplo 1.5 - Os resultados mostrados na Tabela 1.6 refere-se aos *throughputs* de dois sistemas A e B, medidos em transações por segundo. Qual deles é o melhor?

Tabela 1.6 - *Throughput* em transações por segundos

Sistema	Carga 1	Carga 2
A	20	10
B	10	20

Existem três maneiras de comparar o desempenho dos dois sistemas, Tabela 1.7.

f) A primeira maneira é calcular a média dos desempenhos das duas cargas. Isto leva a análise mostrada na Tabela 1.7a. A conclusão neste caso é que os dois sistemas são igualmente bons
g) A segunda forma é considerar a relação de desempenhos sendo B a referência, como mostrado na Tabela 1.7b. A conclusão neste caso é que o sistema A é melhor do que B
h) A terceira forma é considerar a relação de desempenho sendo A a referência, como mostrado na Tabela 1.7c. A conclusão neste caso é que o sistema B é melhor do que A

Tabela 1.7 - Comparando os *throughput* dos sistemas A e B: a) desempenho médio; b) desempenho relativo a A; e c) desempenho relativo a B

Sistema	Carga 1	Carga 2	Média
A	20	10	15
B	10	20	15

(a)

Sistema	Carga 1	Carga 2	Média
A	2	0,5	1,25
B	1	1	1

(b)

Sistema	Carga 1	Carga 2	Média
A	1	1	1
B	0.5	2	1,25

(c)

A Tabela 1.8 apresenta uma lista de questões relativas à análise de desempenho. Todas as perguntas devem ser respondidas afirmativamente de modo que Avaliações de Desempenho sejam bem-sucedidas.

Tabela 1.8 - *Checklist* para evitar erros comuns na Avaliação de Desempenho

1. Os objetivos da análise estão corretamente e claramente definidos?
2. As metas foram estabelecidas de forma imparcial?
3. Todas as etapas da análise foram seguidas sistematicamente?
4. O problema está claramente entendido antes de analisá-lo?
5. As métricas de desempenho são relevantes para este problema?
6. A carga de trabalho é correta para este problema?
7. A técnica de avaliação é a mais adequada?
8. Os parâmetros utilizados são aqueles que afetam o desempenho por completo?
9. Todos os parâmetros que afetam o desempenho foram escolhidos como variáveis?
10. O projeto experimental é eficiente em termos de tempo e de resultados?
11. O nível de detalhamento é apropriado?
12. Os dados medidos foram analisados e interpretados adequadamente?
13. A análise estatística está correta?
14. A análise de sensibilidade foi feita?
15. Os erros na entrada afetaram os resultados de forma significativa?
16. Os outliers na entrada e, ou, na saída foram tratados adequadamente?
17. As alterações futuras do sistema e a carga de trabalho foram modeladas?
18. A variação dos dados de entrada foi levada em consideração?
19. A variação dos resultados foi analisada?
20. A análise pode ser facilmente explicada?
21. O estilo de apresentação é o mais adequado para o público alvo?
22. Os resultados foram apresentados de forma gráfica da melhor forma possível?
23. Os pressupostos e as limitações da análise estão claramente documentados?

A maioria dos problemas de desempenho é único. As métricas, a carga de trabalho e as técnicas de avaliação usadas para um problema geralmente não podem ser usadas para o próximo problema. No entanto, existem alguns passos comuns a todos os projetos de avaliação de desempenho que ajudam a evitar os erros comuns, Tabela 1.9.

Tabela 1.9 - Etapas para uma Avaliação de Desempenho que ajudam a evitar erros comuns

1. Definição do Sistema e Objetivos
2. Lista de Serviços e Resultados
3. Seleção das Métricas
4. Lista de Parâmetros
5. Fatores Selecionados para Estudo
6. Seleção da Técnica de Avaliação
7. Seleção da Carga de Trabalho
8. Design dos Experimentos
9. Analisar e Interpretação dos Dados
10. Apresentação dos Resultados

1.5. Exercícios

1. Que métricas de desempenho devem ser usadas para comparar o desempenho dos seguintes sistemas?

a) duas unidades de discos
b) dois sistemas de processamento de transações
c) dois algoritmos de retransmissão de pacote

d) dois circuitos eletrônicos

2. O número de pacotes perdidos em dois links foi medido para quatro tamanhos de arquivo, como mostrado na Tabela 1.10. Qual link é o melhor?

Tabela 1.10 - Pacotes perdidos em dois links

Tamanho do Arquivo (B)	Link A	Link B
1000	5	10
1200	7	3
1300	3	0
50	0	1

3. Que metodologia deve ser utilizada para:
a) selecionar um computador pessoa
b) selecionar 1.000 postos de trabalho para uma empresa
c) comparar dois pacotes de planilha
d) Para comparar duas arquiteturas de fluxo de dados, se a resposta fosse necessária ontem, no próximo trimestre e no ano que vem.

4. Faça uma lista completa de métricas para comparar:
a) dois computadores pessoais
b) dois sistemas de banco de dado
c) duas unidades de disco
a) dois sistemas de janela

5. A partir da Tabela 1.11 faça um estudo da confiabilidade e da disponibilidade deste sistema computacional.

Tabela 1.11 - Amostra de falhas de sistemas operacionais MS Windows 7 medidas em 735 computadores

Grupo de Computador	1	2	3	4	5	6
Período de medição (dias)	269	72	332	391	378	891
Total de computadores	5	63	268	275	41	83
Total de falhas	284	406	6844	19725	548	3008
Duração das falhas (dias)	23,67	156,49	13381,99	2720,38	1218,14	6803,68

Fonte: http://sbesc.lisha.ufsc.br/sbesc2014/dl222

2. Medições

2.1. Técnicas e Ferramentas de Medição

Sistemas de computação estão se tornando cada vez mais onipresentes na nossa vida cotidiana. As pessoas confiam cada vez mais nestes sistemas para resolver a maioria dos seus problemas como, por exemplo, saúde, educação, entretenimento e finanças.

A maioria das pessoas precisam interagir com os sistemas de apoio automatizados ou semi-automatizados e esperam respostas imediatas. O número de pessoas com acesso a serviços de comunicação está aumentando a taxas exponenciais.

Medições de desempenho de sistemas de computação envolve monitorá-lo enquanto ele está sendo submetido a uma carga de trabalho particular. A fim de realizar medições significativas, a carga de trabalho deve ser cuidadosamente selecionada e, para atingir esse objetivo, o analista de desempenho precisa entender e responder as seguintes perguntas, antes de realizar medições:

a) Quais são os diferentes tipos de cargas de trabalho?
b) Que cargas de trabalho são comumente usadas por outros analistas?
c) Os tipos de carga de trabalho selecionados são adequados?
d) Como os dados medidos da carga de trabalho serão sumarizadas?
e) Como é o desempenho do sistema monitorado?
f) Como colocar a carga de trabalho desejada no sistema de modo controlado?
g) Como os resultados da avaliação serão apresentados?

2.2. Seleção e Caracterização de Carga

A termo carga de trabalho de teste denota qualquer carga de trabalho utilizada em estudos de desempenho. A carga de trabalho de teste pode ser real ou sintética.

A carga de trabalho real é aquela observada em um sistema durante sua operação. Sua medição não pode ser repetida e, portanto, geralmente não

é adequado para uso como uma carga de trabalho de teste.

A carga de trabalho sintética é desenvolvida e usada para estudos, possui características semelhantes aos da carga de trabalho real, mas pode ser aplicadas várias vezes de maneira controlada. A principal razão para a utilização de uma carga de trabalho sintética é que ela é uma representação ou modelo da carga de trabalho real. Outras razões para a utilização de uma carga de trabalho sintética é:

- não conter dados do mundo real
- não são grandes
- não conter dados sensíveis
- pode ser facilmente modificada sem afetar a operação
- pode ser facilmente portada para sistemas diferentes devido ao seu pequeno tamanho
- podem ser incorporadas como funções internas de medição

A Tabela 2.1 discute os principais tipos de cargas de trabalho de teste que tem sido utilizadas para comparar sistemas de computação.

Tabela 2.1 - Cargas de trabalho de teste utilizadas para comparar sistemas de computação

instruções mistas	simulam a demanda de utilização dos recursos de um sistema por meio de um conjunto de instruções do seu processador
kernel	consiste no exame da parte essencial de uma aplicação, suas partes mais frequentemente usadas são determinadas e a seguir programadas. Um programa kernel é uma mistura de instruções que compõem um programa ou parte de um programa e o seu tempo de execução é determinado com base nos tempos de instrução fornecidos pelo fabricante
programas sintéticos	simulam a demanda de utilização dos recursos do sistema de maneira requerida pela carga, são usados no sistema como carga-piloto reproduzível, podendo ter suas características modificadas por meio de um conjunto de parâmetros de entrada
benchmarks	é um conjunto de programas selecionados de maneira a construir uma composição representativa de carga de uma instalação que são processados no sistema que se deseja avaliar. Permite determinar se um sistema particular pode processar adequadamente uma carga real, se as características de preço e desempenho do sistema proposto são suficientes e quais as melhores opções de configuração, tanto de hardware quanto de software

A carga de trabalho é a parte mais importante de qualquer projeto de avaliação de desempenho. É possível chegar a conclusões equivocadas se a carga de trabalho não está devidamente selecionada. A adequação da carga de trabalho é uma etapa crítica para que as conclusões de um estudo sejam aceitáveis.

Como outros aspectos da avaliação de desempenho, a seleção adequada de cargas de trabalho requer muitas considerações e julgamentos pelo analista, que é uma parte da arte da avaliação de desempenho que vem com a experiência. Vale lembrar a máxima da computação: **entra lixo sai lixo**.

Os quatro principais considerações na seleção da carga de trabalho são:

- serviços executados por ela
- seu nível de detalhe
- sua representatividade
- oportunidade

A melhor maneira de iniciar a seleção de carga de trabalho é ver o sistema como um fornecedor de serviços. Cada sistema oferece uma série de serviços e fazer uma lista destes serviços é um dos primeiros passos de um estudo sistemático de avaliação de desempenho.

Muitas vezes, o termo Sistema em Teste (SUT - *System Under Test*) é utilizado para designar o conjunto completo dos componentes que estão sendo avaliados, adquiridos ou sendo projetados. Às vezes há um componente específico no SUT cujas alternativas estão sendo consideradas. Este componente é chamado de componente em estudo (CUS - *Component Under Study*), como mostrado na Figura 2.1.

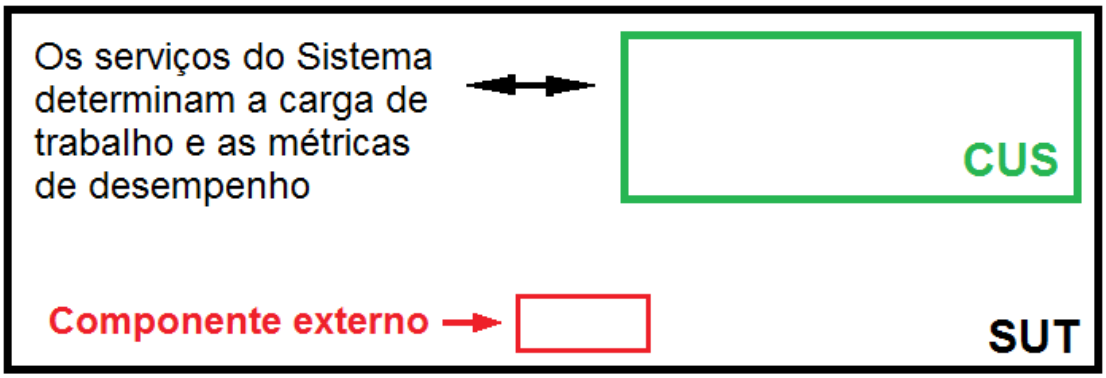


Figura 2.1 - Componentes de um sistema destacando o Sistema em Teste (SUT) e o Componente em Estudo (CUS)

Por exemplo, a equipe de design de uma CPU pode querer entender o impacto de diferentes organizações das Unidades Lógica Aritmética (ALU). Neste caso, a CPU é o SUT e o ALU é o CUS. Da mesma forma um Provedor Web, para comprar um sistema de processamento de transações pode querer comparar diferentes dispositivos de disco. Neste caso, o sistema de processamento de transações é o SUT e os dispositivos de disco são os CUS.

Claramente, a identificação do SUT e do CUS é importante pois a carga de trabalho, bem como as métricas de desempenho são determinados principalmente pelo SUT. Confundir CUS com SUT, e vice-versa, é um erro comum que leva a resultados enganosos.

A palavra sistema será usada para significar SUT e a palavra componente será usada para significar CUS.

As métricas escolhidas devem refletir o desempenho dos serviços prestados ao nível do sistema e não no nível do componente. Por exemplo,

o MIPS é uma métrica justificável para comparar duas CPUs, mas não é apropriado para comparar dois sistemas de tempo compartilhado. A CPU é apenas um componente do sistema de tempo compartilhado. Um sistema de tempo compartilhado pode prestar serviços tais como processamento de transações, caso em que o desempenho seria medido por transações (ao contrário de instruções) por segundo.

A base para seleção de carga de trabalho é o sistema e não o componente. Por exemplo, os serviços prestados pela CPUs são instruções, e os projetistas de CPU podem querer usar a frequência de instruções como uma representação possível da carga de trabalho. Os serviços prestados pelos sistemas web são geralmente chamados de transações e, assim, pode-se utilizar as frequências de transação como a carga de trabalho.

2.3. Monitores

Um monitor é uma ferramenta utilizada para observar as atividades em um sistema. Em geral, os monitores são usados para observar o desempenho dos sistemas, coletar estatísticas de desempenho, analisar os dados e exibir os resultados. Alguns também identificam áreas problemáticas e propõem soluções.

Monitores são usados não só pelos analistas de desempenho, mas também por programadores e gerentes de sistemas. Algumas das razões para monitorar um sistema são os seguintes:

- um programador de sistemas podem usar um monitor para encontrar os segmentos frequentemente usados do software e otimizar seu desempenho
- um gerente de sistemas pode usar um monitor para medir a utilização de recursos e para encontrar o gargalo de desempenho
- um gerente de sistemas também pode utilizar um monitor para ajustar o sistema. Os parâmetros do sistema podem ser ajustados para melhorar o seu desempenho
- um analista de sistemas pode usar um monitor para encontrar os parâmetros do seu modelo analítico, para validá-los e obter dados para entrada de simulação

Em resumo, a monitoramento é o primeiro passo e é chave em medições de desempenho.

2.3.1. Terminologia para Monitor

Os termos relacionados a monitoramento e que são usados com frequência estão discutidos na Tabela 2.2.

Tabela 2.2 - Terminologia para os monitores de desempenho e suas descrições

evento	a mudança de estado do sistema é chamado de evento. Exemplos de eventos são processo de mudança de contexto, início de busca em um disco

	e a chegada de um pacote
trace	um traço é um log de eventos em geral, incluindo o tempo do evento, o tipo de evento e outros parâmetros importantes associados a ele
overhead	a maioria dos monitores perturbam ligeiramente a operação do sistema. Eles podem consumir recursos do sistema, como CPU ou armazenamento. Por exemplo, os dados coletados pelo monitor podem ser gravadas no armazenamento secundário. Este consumo de recursos do sistema é chamado de *overhead* (sobrecarga). Um dos objetivos do projeto de monitorar é a de minimizar a sobrecarga
domínio	o conjunto de atividades observáveis pelo monitor é o seu domínio. Por exemplo, a contabilidade registra informações sobre registro de tempo de CPU, número de discos, terminais, redes e paginação E/S, o número de caracteres transferidos entre os discos, terminais, redes e dispositivo de paginação e o tempo de resposta para cada sessão do usuário. Estes constituem o domínio dos logs de contabilidade
taxa de entrada	a frequência máxima de eventos que um monitor pode observar corretamente é chamada de taxa de entrada. Geralmente, duas taxas de entrada são especificados: modo *burst* e sustentada. A taxa de modo *burst* especifica a taxa em que um evento pode ocorrer por um período curto. É maior do que a taxa sustentada, que o monitor pode tolerar por longos períodos
resolução	a granulação da informação observada é chamado de resolução. Por exemplo, um monitor pode ser capaz de registrar o tempo apenas em unidades de 16 milissegundos. Da mesma forma, o tamanho das classes utilizadas em um histograma pode determinar a resolução do histograma
largura de entrada	o número de bits de informações gravadas em um evento é chamado de largura de entrada. Isto, junto com a taxa de entrada, determina o armazenamento necessário para registrar os eventos

2.3.2. Classificação de Monitores

Os monitores são classificados com base em uma série de características como o nível de implementação, mecanismo de disparo e habilidade em exibir resultados.

Dependendo do nível em que um monitor é implementado, ele é classificado como um monitor em software, em hardware, em firmware ou monitor híbrido. O monitor híbrido é uma combinação de hardware, firmware ou software. Esta é a classificação mais comum.

Dependendo do mecanismo que desencadeia a ação do monitor, um monitor pode ser classificado como orientado a evento ou temporizado (monitor para amostragem). Um monitor orientado a eventos é ativado somente pela ocorrência de determinados eventos. Assim, não há sobrecarga de monitoramento se o evento é raro. Mas se o evento é frequente, pode causar muita sobrecarga. O monitor de amostragem é ativada em intervalos de tempo fixo por interrupções do relógio.

Monitores de amostragem são ideais para a observação de eventos frequentes. Na ativação, o monitor registra o estado do dispositivo e

contadores. A frequência de amostragem é determinada pela frequência do evento e a resolução desejada.

Outra forma de classificar monitores está de acordo com sua capacidade de exibir resultados. Monitores online exibem o estado do sistema de forma contínua ou em intervalos. Monitores de lote, por outro lado, recolhem dados para que possam ser analisados posteriormente, utilizando um programa de análise em separado.

Todas as três classificações anteriores podem ser usadas juntas para caracterizar um monitor. Por exemplo, um monitor específico pode ser classificado como um monitor híbrido de amostragem de lote.

2.3.3. Monitores em Software

Monitores de software são usadas para monitorar os sistemas operacionais e software como redes e bancos de dados. A cada ativação, várias instruções são executadas, e, portanto, eles são adequados apenas se a taxa de entrada é baixa. Por exemplo, se o monitor executa 100 instruções por evento, cada ativação levaria 0,1 milissegundo em uma máquina de 1 MIPS. Assim, para limitar a sobrecarga para 1%, deve ser ativada em intervalos de 10 milissegundos ou mais. Ou seja, a taxa de entrada deve ser inferior a 100 eventos por segundo.

Monitores de software geralmente têm menores taxas de entrada, resoluções mais baixas e maior *overhead* do que os monitores de hardware. No entanto, eles têm larguras de entrada superiores e maior capacidade de gravação do que os monitores de hardware. Eles são mais fáceis de desenvolver e mais fácil de modificar, se necessário.

2.3.4. Monitores em Hardware

Um monitor de hardware consiste em peças separadas de equipamentos que estão ligados ao sistema a ser monitorado por meio de sondas. Sem consumir os recursos do sistema em monitoramento. Assim, monitores de hardware geralmente têm menor sobrecarga do que monitor de software. Sua taxa de entrada também é maior. Além disso, a probabilidade de introduzir erros no funcionamento do sistema é geralmente mais baixa do que a dos monitores de software.

Os monitores de hardware de propósito geral disponíveis no mercado são constituídos dos elementos listados na Tabela 2.3.

Tabela 2.3 - Elementos dos monitores hardware de propósito geral disponíveis no mercado

sondas	sondas de alta impedância são utilizadas para observar os sinais nos pontos desejados no hardware do sistema

contadores	estes são incrementados sempre que um determinado evento ocorre
elementos da lógica	os sinais de muitas sondagens podem ser combinados usando AND, OR e outras portas lógicas. As combinações são usados para indicar eventos que podem incrementar contadores
comparadores	estes podem ser usados para comparar valores de contadores ou sinal com valores pré-definidos
mapeamento de hardware	permite que histogramas de quantidades observadas possam ser calculados. É composto de vários comparadores e contadores
timer	usado para marcar tempo ou para desencadear uma operação de amostragem
unidade de memória	a maioria dos monitores têm unidades de memória para armazenar os dados

2.3.5. Monitores de Software versus de Hardware

Dado um problema de monitoramento, deve-se usar um monitor de hardware ou um monitor de software? A escolha não é tão difícil como pode parecer. Para a maioria das aplicações, apenas um dos dois tipos de monitores será satisfatório.

O primeiro passo na escolha do monitor é considerar o que precisa ser medido. Monitores de hardware podem medir os sinais elétricos do sistema e pode gravá-los com precisão, mesmo em alta velocidade. No entanto, é difícil para eles determinar as informações de nível superior, tais como comprimentos de fila ou usuário atual, a menos que a informação fique facilmente disponível num registo de hardware. Monitores de software, ao contrário, pode facilmente determinar a informação de nível superior, mas não pode observar facilmente eventos de nível inferior, como o tempo para buscar um código de operação para uma instrução. Exemplos de variáveis que podem ser observados por ambos os tipos de monitores são a utilizações de dispositivos e sobreposição CPU-E/S.

A segunda consideração é a taxa de entrada, a taxa na qual os eventos devem ser observados. Monitores de hardware podem registrar eventos muito rapidamente. Um monitor de software, por outro lado, pode exigir algumas centenas de instruções por observação e por isso não pode ser usado se o tempo de intervenção é pequeno.

O resolução de tempo necessária é a consideração seguinte. Um monitor de hardware tem um relógio de hardware separado e pode fornecer a resolução de tempo em alguns nanossegundos. Os monitores de software usam o relógio do sistema, que normalmente tem uma resolução de alguns milissegundos.

A perícia do analista de desempenho também devem ser levados em consideração na escolha do monitor. Apenas um analista com um bom

conhecimento do hardware do sistema pode utilizar corretamente um monitor de hardware. Um monitor de software, por outro lado, requer um conhecimento profundo do software do sistema a ser instrumentado.

A quantidade de dados gravados diretamente afeta a sobrecarga de um monitor de software. Se o montante deve ser muito grande, um monitor de hardware com armazenamento interno secundário deve ser utilizado.

Um monitor de software, por sua natureza, é sequencial, uma vez que não pode gravar vários eventos simultâneos (a menos que o software seja distribuído). Por exemplo, se vários dispositivos solicitarem serviços a partir do processador utilizando interrupções, as interrupções serão atendidas sequencialmente e serão observados pelo software do monitor sequencialmente. Monitores de hardware têm várias sondas que pode gravar eventos simultâneos.

Monitores de software consomem recursos do sistema que seriam disponíveis para os usuários. Um monitor de hardware, por outro lado, consome pouco, se houver, dos recursos do sistema. Sua presença pode ou não ser visível para o sistema.

A maioria dos monitores de hardware são projetados para serem anexados a uma variedade de sistemas. Assim, o mesmo monitor pode ser usado para monitorar os sistemas de diferentes fornecedores ou os sistemas que usam diferentes sistemas operacionais. Monitores de software são desenvolvidos especificamente para um determinado hardware e software de base e não podem ser facilmente vendidos separadamente.

Um monitor de hardware continua observando o sistema mesmo quando ele não está funcionando corretamente, e assim, ele pode ser usado para depurar o sistema. Um monitor de software podem não ser capaz de observar corretamente durante avarias, e não pode ser executado quando o sistema trava.

Monitores de hardware e de software podem ter *bugs* e introduzir erros nos dados medidos. No entanto, com monitores de software, uma vez que o software foi completamente depurado, os erros são raros. Com um monitor de hardware, totalmente depurado, é possível erros de sondagem.

Finalmente, e mais importante, os monitores de hardware são mais caros que os monitores de software. Este fato sozinho pode ser suficiente para polarizar a escolha em muitos casos.

Uma comparação entre monitores de hardware versus monitores de software estão resumidos na Tabela 2.4.

Tabela 2.4 - Comparação entre monitores de hardware e de software

Critério	Monitor de Hardware	Monitor de Software
Domínio	difícil de monitorar eventos do sistema operacional	difícil monitorar eventos de hardware a não ser reconhecível por uma instrução

Taxas de entrada	taxa de amostragem de 10^5 por segundo é possível	taxa de amostragem limitada pelo processador e é necessário *overhead*
Resolução de tempo	10 nanosegundos é possível	geralmente 10 a 16 milissegundos
Perícia	requer profundo conhecimento do hardware	requer o conhecimento do software
Capacidade de gravação	limitada pela memória e armazenamento secundário, não é um problema atualmente	limitada pela sobrecarga desejada
Largura de entrada	pode gravar simultaneamente vários eventos	não é possível gravar vários eventos simultâneos a menos que haja múltiplos processadores
Overhead do Monitor	Nenhum	A sobrecarga depende das taxa de entrada e largura de entrada, menos de 5% é o adequado
Portabilidade	geralmente portátil	específicas para um sistema operacional
Disponibilidade	monitoramento continua mesmo durante mau funcionamento do sistema ou falha	não pode monitor durante o travamento do sistema
Erros	possíveis ao conectar as pontas de prova em pontos errados	uma vez depurado, os erros são raros
Custo	alto	médio

2.3.6. *Firmware* e Monitores Híbridos

Monitores de *firmware* são implementados através da modificação do microcódigo do processador. Estes são úteis para aplicações que se situam entre o software e os limites de monitoramento de *hardware*. Na maioria dos aspectos, monitores *firmware* são semelhantes aos monitores de software. No entanto, uma vez que o espaço para o microcódigo é limitado e existem limitações de tempo, monitores de *firmware* geralmente são muito limitados. Eles são úteis em aplicações onde as considerações de tempo impede o uso de monitores de software e a inacessibilidade de pontos de sondagem impede o uso de monitores de hardware.

Monitores de *firmware* têm sido usados para monitoramento de rede, onde as interfaces de rede existentes podem ser facilmente microprogramadas para monitorar o tráfego na rede. Outra aplicação adequada para monitores *firmware* é para gerar perfis de endereço do microcódigo. Esses perfis são usados para otimizar o código. Um monitor usando uma combinação de hardware, software ou firmware é um monitor híbrido.

Monitores de software têm boas capacidades de redução de dados, enquanto os monitores de hardware têm alta resolução. Assim, um monitor híbrido, composto de um componente de hardware para coleta de dados junto com um componente de software para a redução de dados, oferece o melhor dos dois mundos.

2.4. *Benchmarking*

Um dos problemas importantes para os gestores de instalações de sistemas de computação é o **planejamento de capacidade**, que consiste em assegurar que os recursos adequados de computador estarão disponíveis para atender a demanda futura de carga de trabalho de forma rentável, satisfazendo simultaneamente os objetivos de desempenho.

O termo **gerenciamento da capacidade** é usado para denotar o problema de garantir que os recursos computacionais disponíveis estão sendo usados para fornecer o mais alto desempenho. Assim, o gerenciamento de capacidade se preocupa com o presente, enquanto o planejamento de capacidade está preocupado com o futuro.

As alternativas para o planejamento de capacidade geralmente consistem em obter mais recursos de computação, enquanto as alternativas para o gerenciamento de capacidade consistem em ajustar os padrões de uso, reorganizando a configuração e mudando os parâmetros do sistema para maximizar o desempenho. O processo de ajuste dos parâmetros do sistema para otimizar o desempenho também é chamado de ajuste de desempenho.

Para comparar o desempenho de dois sistemas concorrentes de uma forma objetiva, *benchmarks* são executados nesses sistemas usando carga direcionada e de modo automático. Os erros e a comparação de benchmarking, bem como questões relacionadas com o direcionamento de carga são discutidos neste capítulo.

Benchmarking é o processo de comparar dois sistemas que utilizam o padrão bem conhecido *benchmarks*. O processo nem sempre é realizado de forma justa. Algumas das maneiras que os resultados de um estudo de *benchmarking* podem ser enganosos ou tendenciosos são discutidos a seguir.

a) Configurações diferentes podem ser usadas para executar a mesma carga de trabalho nos dois sistemas. As configurações podem ter uma quantidade diferente de memória, discos diferentes ou número diferente de discos.
b) Os compiladores podem otimizar a carga de trabalho. Há casos em o compilador, ao otimizar o código executável, elimina partes de programas sintéticos, dando assim o desempenho melhor do que outros sistemas não otimizados
c) Especificações de teste podem ser escritos de forma que eles privilegiam uma máquina. Isso pode acontecer se as especificações são escritas com base em um ambiente existente, sem consideração para generalizar os requisitos para diferentes fornecedores
d) Sequência de tarefas sincronizados podem ser usadas. É possível manipular uma sequência de trabalho para que sincronize a CPU o sistema de E/S produzindo um melhor desempenho global
e) A carga de trabalho pode ser arbitrariamente escolhida. Muitos dos programas não foram verificados para serem representativos das aplicações do mundo real
f) *Benchmarks* muito pequenos pode ser usado. *Benchmarks* com poucas instruções podem explorar pouco os circuitos internos de hardware, aumentando acertos de cache, não mostrar o efeito de sobrecarga de E/S e alternância de contexto. Uma escolha pouco

criteriosa de instruções de loop pode levar a resultados distorcidos, por qualquer quantidade desejada. A maioria dos sistemas reais fazem uso de uma ampla variedade de cargas de trabalho. Para comparar dois sistemas, deve-se, portanto, utilizar o maior número possível de cargas de trabalho. Usando apenas alguns *benchmarks* selecionados, os resultados podem ser tendenciosos, conforme desejado

g) *Benchmarks* podem ser traduzidos manualmente para otimizar o desempenho. Muitas vezes, *benchmarks* precisam ser traduzidos manualmente para torná-los executáveis em sistemas diferentes. O desempenho pode, então, depender mais da habilidadedo do tradutor do que no sistema em teste

2.4.1. Etapas do Planejamento e da Gestão de Capacidade

A Figura 2.2 mostra as etapas do processo de planejamento de capacidade. Para o planejamento, bem como a gestão, as etapas são basicamente iguais:

a) Instrumentalizar o sistema
b) Monitorar o uso do sistema
c) Caracterizar a carga de trabalho
d) Prever o desempenho sob diferentes alternativas
e) estabelecer o modelo do sistema
f) Selecione a alternativa com menor custo e o maior desempenho

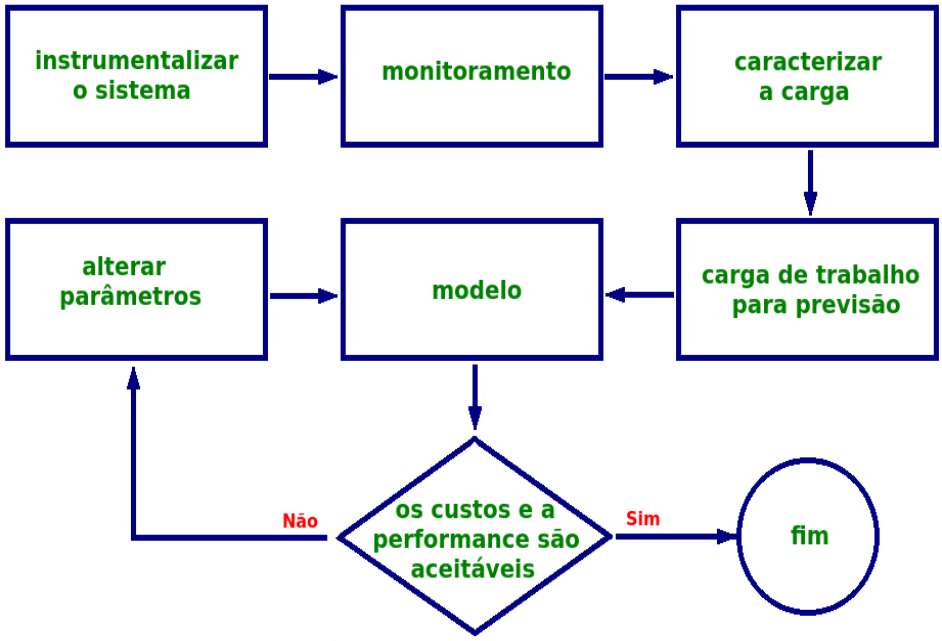


Figura 2.2 - Esquema do processo de planejamento e de gestão de capacidade.

O primeiro passo é garantir que existem contadores apropriados e *links* no sistema para registrar o uso corrente. Na maioria dos casos, os contadores são construídos nos sistemas operacionais, *software* aplicativos e dispositivos de E/S são usados. Uso de dados de registro contábil é provavelmente o método mais popular.

O segundo passo consiste em monitorar o uso e caracterizar a carga de trabalho. Isto requer a coleta de dados por um período de tempo que

permita estimar a entrada para o modelo do sistema para previsão de desempenho.

Para a gestão de capacidade, a configuração atual e carga de trabalho podem ser dados de entrada de um modelo de ajuste para informar alterações na configuração de parâmetros do sistema. Este modelo pode simular em detalhes o sistema e deve conter um conjunto de regras desenvolvidas especificamente para o sistema. Por exemplo, uma das regras pode ser para sugerir a colocação equilibrada de arquivos se um uso altamente enviesada dos dispositivos de disco é visto.

Para planejamento de capacidade, em primeiro lugar a carga de trabalho é prevista com base no monitoramento de longo prazo do sistema. Então as alternativas de configuração diferentes e cargas de trabalho futuro são dados de entrada para um modelo que prevê o desempenho. Esta etapa posterior de seleção de equipamentos também é chamado de dimensionamento. Os modelos para o dimensionamento são geralmente menos detalhados do que os modelos para ajuste de desempenho. Muitas vezes, as técnicas de modelagem analítica, tais como modelos de filas, modelos usados para o dimensionamento uma vez que a lista de alternativas inclui uma ampla variedade de novos hardware e componentes de software para que modelos detalhados podem ainda não existir. Assim, enquanto os modelos de ajuste são detalhados e específicas do sistema, os modelos de planejamento são grosseiros e independem do sistema ou, pelo menos, são menos específicos para o sistema. Em muitos locais, regras muito simples, tais como demanda crescente por um fator x a cada ano y, são usadas para planejamento de longo prazo. Em muitas instalações, a carga de trabalho futura é tão desconhecida que as técnicas de previsão mais sofisticadas não podem ser de grande ajuda.

2.4.2. Problemas no Planejamento de Capacidade

A maior parte da capacidade de planejamento tratada na literatura são seleção de métricas de desempenho, técnicas de monitoramento, caracterização de carga de trabalho, previsão e técnicas de modelagem. Uma vez que cada uma dessas questões é discutida em diferentes partes deste livro, a discussão aqui se limita à lista de problemas enfrentados pelos planejadores de capacidade. Esses problemas são discutidos na Tabela 2.5.

Tabela 2.5 - Principais problemas do planejamento de capacidade

Não há terminologia padrão
Todos os fornecedores de ferramentas de planejamento de capacidade tem uma definição diferente de gestão da capacidade, planejamento de capacidade, dimensionamento, ajuste de desempenho e assim por diante. Muitas vezes se compra uma ferramenta de planejamento de capacidade apenas para descobrir mais tarde que ele faz apenas ajustes ou dimensionamento e não tem nenhuma medida de carga de trabalho ou caracterização das instalações. Alguns fornecedores usam o termo

gerenciamento de capacidade para incluir tanto o planejamento de capacidade e ajuste. Outros usam para denotar ajustes apenas
Não existe uma definição padrão de capacidade
Existem várias possibilidades. Uma definição de capacidade, em termos de rendimento máximo, já foi apresentado. Rendimento é medido em pedidos por unidade de tempo, por exemplo, tarefas por segundo, transações por segundo (TPS), instruções por segundo (MIPS), ou bits por segundo (para *links* de rede). Outra possibilidade é definir a capacidade como o número máximo de usuários que o sistema pode suportar enquanto atende um objetivo de desempenho especificados. Nesta definição, os usuários são apenas um exemplo do que tem sido chamado de unidade de carga de trabalho. Outras unidades de carga de trabalho são pessoas, sessões, tarefas, atividades, programas, contas, projetos e assim por diante, e a capacidade é expressa nestas unidades. Unidades de carga de trabalho também são chamados de componentes carga de trabalho
Há uma série de capacidades diferentes para o mesmo sistema
Já foram descritas três capacidades, capacidade nominal, capacidade utilizável e joelho de capacidade. Os termos capacidade e outros que têm sido utilizados na literatura são a capacidade prática (capacidade utilizável) e capacidade teórica (capacidade nominal)
Não há unidade de carga de trabalho padrão
O problema com a medição da capacidade em unidades de carga de trabalho, como usuários ou sessões, é que ela requer uma caracterização detalhada da unidade de carga de trabalho que varia de um ambiente para outro. É por causa dessa dificuldade que a carga de trabalho independente de medidas de capacidade, tais como MIPS, ainda são populares. É comum gerentes de sistemas planejarem o futuro em termos de suas necessidades de MIPS
Previsões futuras são de aplicação difícil
A maioria das previsão é baseada na suposição de que a tendência futura será semelhante ao passado. Esta suposição é violada, se uma nova tecnologia surge de repente. Uma série de novas aplicações também se tornaram possíveis devido à capacidade de computação em diversas mídias e de modo distribuído
Não há uniformidade entre os sistemas de diferentes fornecedores
A mesma carga de trabalho utiliza diferentes quantidades de recursos de sistemas diferentes. Isso requer o desenvolvimento de uma referência independente de fornecedor e executá-lo em sistemas diferentes. Além disso, modelos separados (simulação ou analítica) têm de ser desenvolvidos para cada sistema. É possível, inadvertidamente introduzir vieses em qualquer uma destas fases
As entradas do modelo nem sempre podem ser medidas
Modelos de simulação ou analíticos são usados para prever o desempenho sob diferentes alternativas. Às vezes, os insumos necessários para o modelo não são mensuráveis com precisão. Por exemplo, "tempo para pensar" é comumente usado em modelos analíticos. Em um ambiente real, o tempo entre os comandos sucessivos do usuário pode incluir o pensamento, assim como outras interrupções, tais como *coffee breaks*. É quase impossível medir corretamente estes valores de tempo. Determinar insumos dos modelos se torna ainda mais difícil se a ferramenta de monitoramento, a ferramenta de análise de carga de trabalho e ferramentas de modelagem são de diferentes fornecedores. A saída de uma etapa pode não estar em um formato utilizável pelo próximo passo
Validar as projeções de modelos é difícil
Existem dois tipos de validações do modelo. O primeiro tipo, validação de base, exige a utilização de carga de trabalho real e da configuração do modelo e verificar se a saída do modelo corresponde ao desempenho do sistema observado. A segunda validação, chamada de validação de projeção, exige a mudança da carga de trabalho e de configuração e verificar se a saída do modelo corresponde o desempenho do sistema

real alterado. Embora seja fácil de mudar as entradas para um modelo, é difícil controlar a carga de trabalho e configurações em um sistema real. Por esta razão, validações de projeção raramente são executadas. Sem validações de projeção, a utilização do modelo de planejamento de capacidade torna-se suspeito
Ambientes distribuídos são complexos demais para serem modelados
Os primeiros sistemas de computador consistiam de apenas alguns componentes. Cada componente era caro o suficiente para justificar o custo de modelagem com precisão do seu comportamento. Além disso, o número de usuários do sistema era grande. Assim, embora o comportamento de cada usuário seja altamente variável, o desempenho agregado de todos os usuários não varia muito e pode ser modelado com precisão. Nos ambientes distribuídos atuais, o sistema consiste de um grande número de clientes semiautônomos, servidores, *links* de rede e E/S de dispositivos. As interações são também bastante complexa. Além disso, o custo de componentes individuais não é alto o suficiente para justificar sua modelagem precisa
Desempenho é apenas uma pequena parte do problema de planejamento de capacidade
A questão-chave no planejamento de capacidade é a de custo, que inclui o custo do equipamento, software, instalação, manutenção, pessoal, espaço, climatização. Modelagem de desempenho ajuda somente no dimensionamento do equipamento. No entanto, como o custo do hardware do computador está em declínio, estes outros custos estão se tornando dominantes e tornaram-se uma consideração importante no planejamento de capacidade

Apesar dos problemas listados acima, o planejamento de capacidade continua a ser um importante problema a ser enfrentado por um gerente de instalação de computador. Felizmente, uma série de ferramentas de planejamento comercial da capacidade estão disponíveis no mercado. Muitas dessas ferramentas têm bibliotecas de modelos de sistemas específicos e incluem analisadores de carga de trabalho que compreende a contabilidade de logs dos sistemas. Alguns também têm monitores integrados.

2.4.3. Erros Comuns no *Benchmarking*

Benchmarking, que é o processo de execução de benchmarks em sistemas diferentes, é o método mais comum para comparar o desempenho de dois sistemas de planejamento de capacidade e aquisições. A Tabela 2.6 discute os erros que têm sido observados repetidamente neste processo. Como erros discutidos em outras seções, esses erros são um resultado da inexperiência ou desconhecimento por parte dos analistas.

Tabela 2.6 - Erros comuns do processo de *benchmarking*

Apenas o comportamento médio é representado no *Workload Test*
Um teste de carga de trabalho é projetado para ser representativo da carga de trabalho real. A carga de trabalho projetada assegura que as solicitações de recursos relativos são semelhantes aos observados no ambiente real. No entanto, apenas o comportamento médio é representado, a variância é ignorada. Por exemplo, o número médio de E/S, ou tempo médio de CPU na carga de trabalho de teste podem ser semelhantes aos da carga de trabalho real. A distribuição, tais como uniforme, exponencial, ou constante, pode ser escolhidas arbitrariamente, sem qualquer validação. Valores constantes são indesejáveis, uma vez que

pode causar sincronizações levando a conclusões erradas. Em alguns casos, a variância, ou mesmo uma representação mais detalhada das demandas de recursos é necessária
Assimetria de dispositivo é ignorada
Os pedidos de E/S são comumente assumidos ser igualmente distribuídos entre todos os dispositivos de E/S. Na prática, os pedidos de I/0 pode seguir um comportamento direcionado de tal forma que sucessivos E/S podem acessar o mesmo dispositivo que permite tempos maiores nas filas. Ignorar esta assimetria para pedidos de E/S e para solicitações de rede leva a previsões de menor tempo de resposta e pode não mostrar os gargalos de dispositivo que podem ocorrer em ambientes reais
Nível de carga controlado inadequadamente
As cargas de trabalho de teste têm vários parâmetros que podem ser alteradas para aumentar o nível de carga no sistema. Por exemplo, o número de usuários pode ser aumentado, o tempo de pensar dos usuários pode ser diminuído, ou a demanda de recursos por usuário pode ser aumentado. Estas três opções não têm os mesmos resultados. A maneira mais realista é aumentar o número de usuários, mas nas medições isso requer mais recursos ou mais espaço de armazenamento em uma simulação, e assim as outras duas alternativas podem ser usadas. Alterar o tempo de pensar é a alternativa mais fácil mas não é equivalente a primeira alternativa já que a ordem dos pedidos de vários dispositivos não é alterado, portanto, o erros de cache podem ser muito menores do que aqueles em um sistema com mais usuários. A terceira alternativa muda significativamente a carga de trabalho, e isso pode não ser representativa do ambiente real
Efeitos de cache são ignorados
Caches são muito sensíveis à ordem das solicitações. Na maioria dos estudos de caracterização de carga de trabalho, as informações do pedido é perdido. Mesmo igualando a média e variância de demandas de recursos não garante que o desempenho de cache será semelhante ao que em um ambiente real. Nos sistemas modernos, cache é usado para acessos à memória, discos, bem como redes. A ordem de chegada tornou-se mais necessária para os modelos
Tamanhos de *buffer* não apropriado
O tamanho do *buffer* ter um impacto significativo sobre o desempenho de E/S e dispositivos de rede. Por exemplo, em alguns casos, uma mudança de 1 byte no tamanho do *buffer* pode dobrar o número de mensagens de rede. O tamanho e o número de *buffers* são geralmente parâmetros do sistema e seus valores em sistemas experimentais deve representar os valores usados em sistemas de produção
Imprecisões na amostragem
Alguns dos dados coletados para a caracterização de carga de trabalho é recolhida através de amostragem, onde, em uma série de indicadores do estado de contadores são lidos periodicamente. Às vezes, isso pode levar a erros significativos nas medidas
Ignorar o *Overhead* no Monitoramento
O mecanismo de coleta de dados ou o monitor software utilizado na medição pode ter sobrecarga e apresentar apresenta algum erro nos valores medidos. Ele pode consumir processador, armazenamento e E/S recursos para manter e registrar as medidas. Se a inexatidão, por exemplo, é de 10%, o resultado do modelo pode ser mais ou menos imprecisos, mesmo se o modelo for preciso
Não Validar Medidas

Este é um erro comum. Enquanto os modelos de simulação e analíticos são rotineiramente validados antes do uso, a validação dos dados medidos raramente é pensada. Qualquer erro na configuração do experimento pode permanecer despercebido. Por exemplo, em monitores de hardware, é fácil deslocar uma sonda ou ter uma sonda solta. Portanto, é necessário verificar as medições. Todos os resultados contraintuitivos devem ser explicados. Quaisquer valores que não podem ser explicados deve ser investigado. Verificações automáticas devem ser incluídas nas medidas, por exemplo, o número total de pacotes enviados por todos os nós em uma rede deve ser próximo do número total recebido

Não Garantir as Mesmas Condições Iniciais
Cada execução do *benchmark* muda o estado do sistema. Por exemplo, o espaço em disco disponível poderá ser reduzido. Os registros de dados podem ter diferentes conteúdos. Iniciar a próxima execução no sistema que foi alterado pode fazer o experimento não-repetível. Uma solução é garantir que todos os arquivos criados por uma carga de trabalho sejam eliminados e o estado inicial do sistema seja reestabelecido. Outra solução é estudar a sensibilidade dos resultados de tais fenômenos. Se os resultados são muito sensíveis às condições iniciais, então mais fatores devem ser adicionadas ao modelo de carga de trabalho
Não Medição de Desempenho Transiente
A maioria das medições, das simulações e dos modelos analíticos são projetados para prever o desempenho em condições estáveis. Durante as medições, o sistema é levado a alcançar um estado estável antes de que medidas são tomadas. Esta é uma abordagem válida na maioria dos casos. No entanto, se o sistema é tal que leva muito tempo para atingir o estado estacionário, em ambientes reais, muitas vezes o estado do sistema deslocam de um estado para outro. Em outras palavras, o sistema poderá estar em um estado transiente mais frequentemente do que em um estado estacionário. Neste caso, é mais realista estudar o desempenho transiente do sistema do que o desempenho em estado estacionário
Usando a Utilização de dispositivos para Comparações de Desempenho
Utilização do dispositivo é uma métricas de desempenho no sentido de que, dada a mesma carga de trabalho, uma menor utilização é a preferida. No entanto, seu uso para comparar dois sistemas podem ser, por vezes, sem sentido. Por exemplo, em um ambiente fechado pedidos são gerados a intervalos de tempo fixo. Dados dois sistemas com tal ambiente, o sistema com o tempo de resposta menor terá mais pedidos gerados por unidade de tempo e terá maior utilização de dispositivo. Menor utilização para o mesmo número de usuários não deve ser interpretada no sentido de um sistema melhor. A métrica correta para comparar dois sistemas, neste caso, é comparar o rendimento em termos de solicitações por segundo. Outro erro comum é usar utilização para validações de modelos
Muita Coleta de Dados mas Com Pouca Análise
Este é um erro comum. A coleta de dados é o primeiro passo para *benchmarking*. O próximo passo é a análise de dados e pode não receber atenção adequada por várias razões. Primeiro, a pessoa que faz a medição pode ter pouco ou nenhum treinamento em técnicas de estatística. Os gráficos usuais de demanda de recursos e utilização em função do tempo podem ser preparados, mas não interpretados adequadamente. Em segundo lugar, a medição dos dados pode tomar muito tempo do projeto e faltar tempo para análise. Uma maneira de evitar essa armadilha é formar equipes de analistas com formação de medição e análise, alocar tempo de análise durante o planejamento do projeto e para discutir as atividades de medição e análise. Muitas vezes os planos de medição pode ter que ser alterado com base nos resultados da análise

A lista acima inclui apenas os erros que um analista pode fazer inadvertidamente, devido à inexperiência.

2.5. *Benchmarks* Populares

Na área comercial, o termo *benchmark* é quase sempre usado como sinônimo de carga de trabalho. *Kernels*, programas sintéticoss e cargas de trabalho em nível de aplicativo, por exemplo, são todos chamados *benchmarks*. Embora o mix de instruções é um tipo de carga de trabalho, eles não são chamados *benchmarks*. Alguns autores têm tentado restringir o termo *benchmark* para se referir apenas ao conjunto de programas que utilizam cargas de trabalho. Esta distinção, no entanto, tem sido quase sempre ignorada na literatura. Assim, o processo de comparação de desempenho por dois ou mais sistemas meio de medições é chamado de benchmarking, e as cargas de trabalho utilizadas nas medições são chamados *benchmark*. Alguns dos *benchmarks* bem conhecidos são descritas na Tabela 2.7.

Tabela 2.7 - *Benchmarks* populares

Sieve
O *kernel sieve* foi usado para comparar microprocessadores, computadores pessoais e linguagens de alto nível. Ele é baseado no algoritmo de Crivo de Eratóstenes e é usado para localizar todos os números primos abaixo de um determinado número n.
Função de Ackermann
Este *kernel* foi utilizado para avaliar a eficiência do mecanismo de procedimento de chamada em linguagem Algol. A função tem dois parâmetros e é definida de forma recursiva. A função de Ackermann(3,n) é avaliada para valores de n de 1 a 6. O tempo de execução médio por chamada, o número de instruções executadas por chamada e a quantidade de espaço de pilha necessário para cada chamada são usados para comparar sistemas.
Whetstone
Utilizado na Agência Central de Computação Britânica, o *kernel* Whetstone consiste de um conjunto de 11 módulos projetados para combinar dinamicamente as frequências observadas das operações usadas em 949 programas. Este *kernel* avalia os recursos do processador, como matriz de endereçamento, aritmética de pontos fixo e flutuante, chamadas de sub-rotinas e passagem de parâmetros. Os resultados dos *benchmarks* Whetstone são medidos em WIPS (Instruções Whetstone por Segundo). Há muitas permutações do *benchmark* Whetstone, por isso é importante para assegurar que as comparações entre vários sistemas utilizem o mesmo código fonte e que a definição do contador de laços interno é grande o suficiente para reduzir a variabilidade de tempo. Apesar de seu mix de operações sintéticas, Whetstone é geralmente é considerado uma referência de ponto flutuante e é mais representativo de pequenas aplicações de engenharia/científicas que cabem na memória cache. Seus módulos foram projetados para minimizar o impacto das otimizações conhecidas dos compiladores. Novas técnicas de otimização do compilador podem afetar significativamente o tempo de execução em um processador. Ele sofre de outros problemas de *kernels*, nele não há E/S e os valores dos parâmetros de entrada podem afetar significativamente o desempenho medido.

Linpack
Desenvolvido por Jack Dongarra (1983) do Argonne National Laboratory, este *benchmarck* é composto de uma série de programas que resolve sistemas densos de equações lineares, caracterizadas por ter elevada percentagem de adições e multiplicações de ponto flutuante. A maior parte do tempo de processamento é consumido em um conjunto de subrotinas chamadas Subprogramas Básicos de Álgebra Linear (BLAS), que são chamados várias vezes ao longo do *benchmark*. Os *benchmarks* Linpack são comparados com base na taxa de execução, medida em MFLOPS. As variantes mais populares resolvem um sistema de equações 100×100, tanto em precisão simples quanto dupla, e se tornaram uma das referências mais utilizadas para medir performance de aplicações de engenharia/científicas.
Dhrystone
Desenvolvido em 1984 por Reinhold Weicker na Siemens, este *kernel* contém muitas chamadas de procedimento e é considerado para representar ambientes de programação de sistemas. Os resultados são normalmente apresentados em Instruções Dhrystone Por Segundo (DIPS). A documentação de referência apresenta um conjunto de regras básicas para criação e execução do Dhrystone. O valor de referência foi atualizado várias vezes e é importante para especificar o número da versão quando o *kernel* é usado. O *kernel* supostamente tem pequena dinâmica no aninhamento das chamadas de função, um baixo número de instruções por chamada de função, e uma grande porcentagem de tempo gasto na cópia de cadeia de caracteres. O *benchmark* é uma medida popular de desempenho de aritmética de inteiros, mas não avalia processamento de ponto flutuante ou E/S.

2.6.Pacote de *Benchmark* do SPEC

A Performance Systems Avaliação Cooperativa (SPEC) é uma corporação sem fins lucrativos formada pelos principais fornecedores de computador para desenvolver um conjunto padronizado de *benchmarks*. A versão 1.0 do pacote de benchmarks SPEC (ver SPEC 1990) consiste nos 10 seguintes *benchmarks* retirados de diversas aplicações de engenharia e científicos, Tabela 2.8.

Tabela 2.8 - Pacotes de *benchmarks* SPEC versão 1.0

GCC	mede o tempo para o GNU Compiler C converter 19 arquivos-fontes preprocessados em linguagem *assembly*. Esse *benchmark* é representante de um ambiente de engenharia de *software* e mede a eficiência de um sistema de compilação
Espresso	é uma ferramenta da Electronic Design Automation (EDA) que realiza a minimização da função heurística *boolean* para Arrays Lógicos Programáveis (PLAs). É medido o tempo decorrido para executar um conjunto de sete modelos de entrada
Spice 2g6	outro representante do ambiente EDA, é uma ferramenta de simulação amplamente utilizada em circuito analógico. É medido o tempo para simular um circuito bipolar
Doduc	este é um *benchmark* sintético que realiza uma simulação de Monte Carlo de certos aspectos de um reator nuclear. Devido à sua estrutura iterativa e

	abundância de ramos curtos e *loops* compactos, testa a eficácia de memória cache
NASA7	esta é uma coleção de sete *kernels* de uso intensivo de ponto flutuante realizando operações na matriz de dados de precisão dupla
LI	o tempo decorrido para resolver o problema populares das 9 rainhas pelo interpretador Lisp é medido
Eqntom	este *benchmark* traduz uma representação lógica de uma equação booleana para uma tabela verdade
Matrix300	executa operações de matriz usando várias rotinas Linpack em várias matrizes de tamanho 300×300. O código usa aritmética de ponto flutuante de dupla precisão e é altamente vetorizável
Fpppp	problema de química quântica que usa precisão dupla em ponto flutuante Fortran, é difícil de vetorizavel
Tomcatv	gera uma malha vetorizavel em ponto flutuante de precisão dupla em Fortran. Uma vez que é altamente vetorizável, *speedups* substanciais têm sido observadas em diversos sistemas de memória compartilhada com vários processadores

Estes valores de referência, que enfatizam principalmente a CPU, unidade de ponto flutuante (FPU), e até certo ponto o subsistema de memória, são destinadas para comparar as velocidades de CPU. Parâmetros de comparação para E/S e outros subsistemas podem ser incluídos em versões futuras.

3. Modelagem Analítica de Sistemas de Filas

Há inúmeros sistemas computacionais constituídos por Sistema de Filas (SF) que podem ser modelados por Redes de Sistemas de Filas (RSF). Em geral, RSF são modelos em que as tarefas que saem de um SF são encaminhadas para outro SF (ou para o mesmo SF).

Ao contrário de SF individuais, não há nenhuma notação fácil para especificar o tipo de uma RSF. A maneira mais simples de classificá-las é aberta ou fechada. Uma RSF aberta tem chegadas e partidas externas, como mostrado na Figura 3.1. O número de tarefas numa RSF aberta varia com o tempo. A análise de uma RSF aberta assume-se que a vazão é conhecida (igual à taxa de chegada) e o objetivo é caracterizar a distribuição do número de tarefas no sistema.

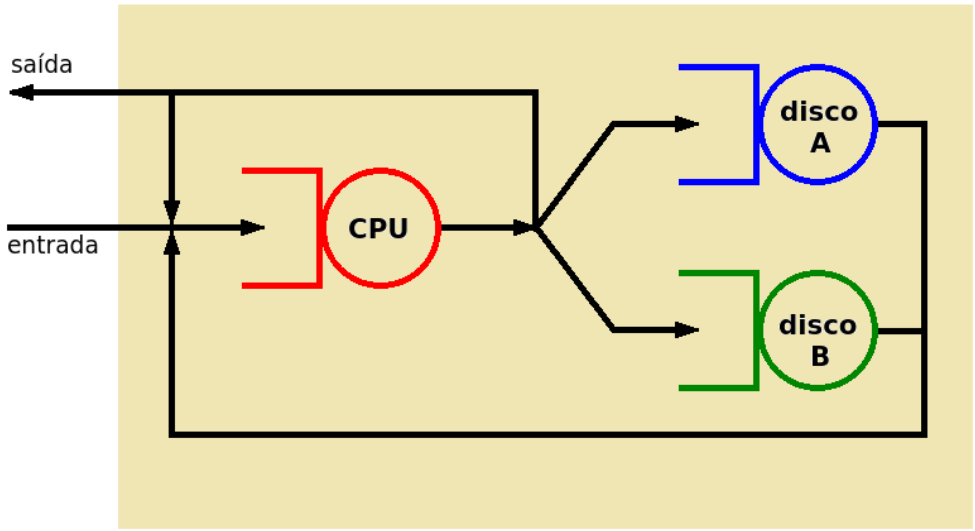


Figura 3.1 - Rede de Sistemas de Filas abertas com tarefas entrando no sistema em entrada e saindo em saída.

Uma RSF fechada não tem chegadas externas ou partidas, o número total de tarefas no sistema é constante. Como mostrado na Figura 3.2, os Servidores do sistema mantêm a circulação das tarefas nos SF da rede. É possível ver uma RSF fechada como um sistema em que a saída está ligada com a entrada e, desta forma, as tarefas que saem da rede voltam

imediatamente a ela. A análise de uma RSF fechada supõem que o número de tarefas é dado e se determinará a taxa de transferência (ou a taxa de conclusão de tarefas).

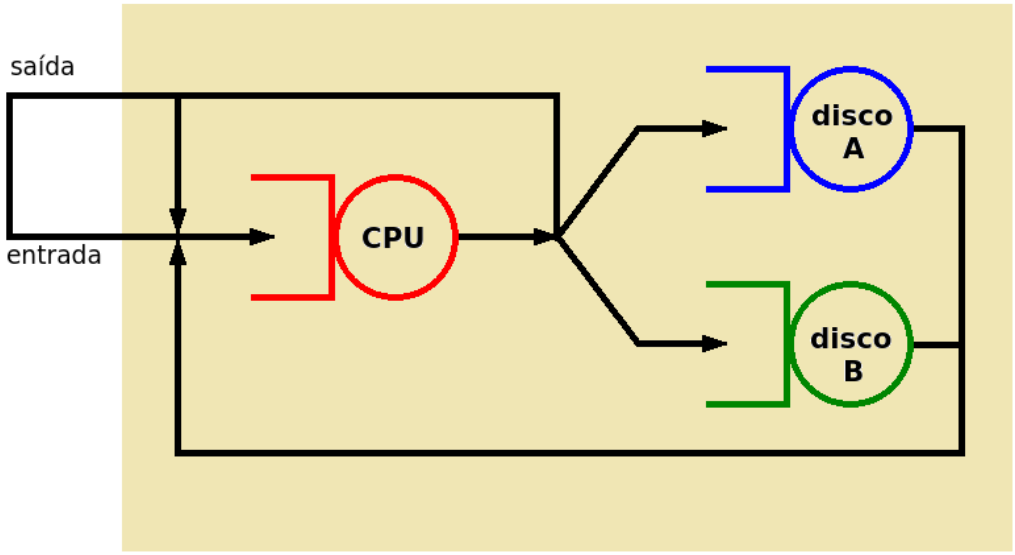


Figura 3.2 - Rede de Sistemas de Filas fechadas em que o fluxo de tarefas saída-entrada define a taxa de transferência do sistema.

Também é possível ter RSF mistas que estão abertas para algumas cargas de trabalho e fechada para outras. Figura 3.3 mostra um exemplo de um sistema misto com duas classes de tarefas: lote e interativas. O sistema é fechado para tarefas interativas e aberto para tarefas em lote. Todas as tarefas de uma mesma classe têm as mesmas demandas de serviço e iguais probabilidades de transição. As tarefas de uma classe são indistinguíveis. A análise de uma RSF mista é feita por classe de tarefa.

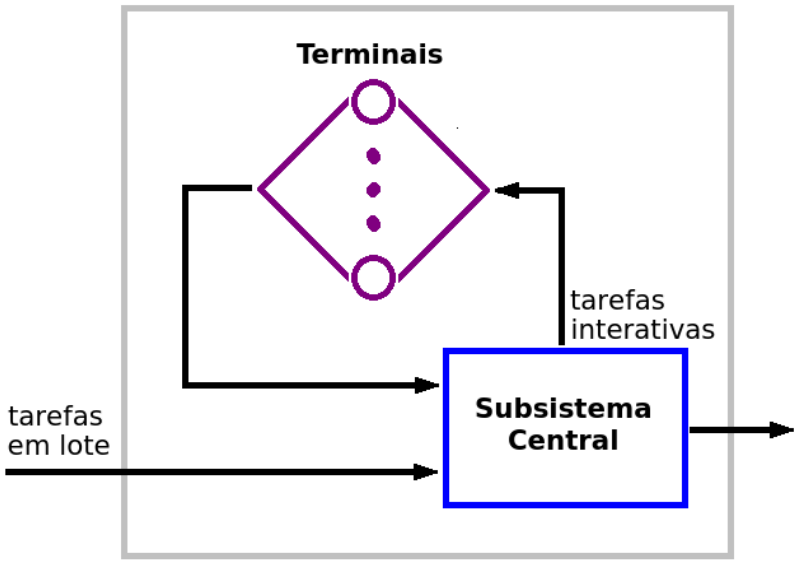


Figura 3.3 - RSF mista.

3.1. Modelos de Rede de Filas de Sistemas de Computadores

Scherr usou um modelo de lojas de reparação de máquinas para representar um sistema de tempo compartilhado com n terminais. O modelo reparação de máquinas é constituído por uma série de máquinas de trabalho e uma oficina com um ou mais reparadores. Sempre que uma máquina quebra, ela é colocada na fila para a reparação e manutenção por um técnico disponível.

O modelo de um sistema de tempo compartilhado é constituído por n terminais com um ou mais servidores. Terminais de usuários terminais geram pedidos (tarefas) que são atendidos pelo servidor, análogo ao reparador, Figura 3.4. Depois que uma tarefa é feita, ele espera no terminal de usuário durante um intervalo de tempo aleatório antes de realizar um novo ciclo.

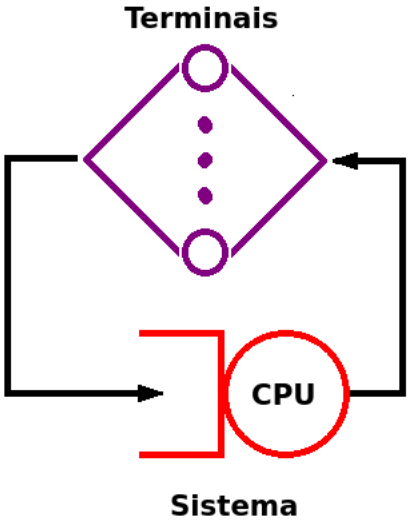


Figura 3.4 - Modelo de máquina reparadora como analogia para sistema computacional de tempo compartilhado.

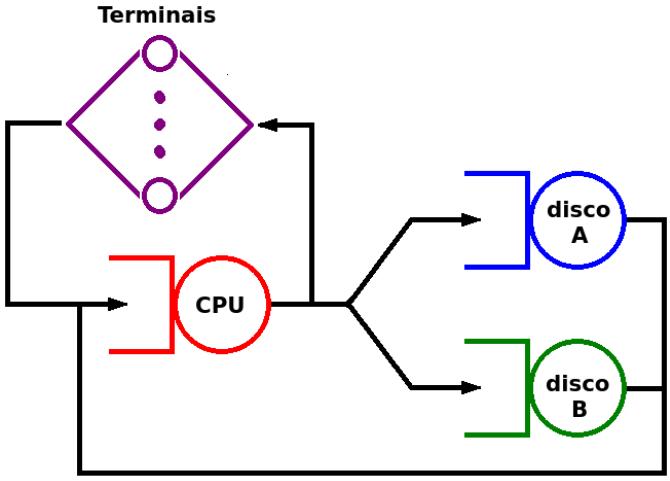


Figura 3.5 - Servidor central como modelo de sistema de tempo compartilhado.

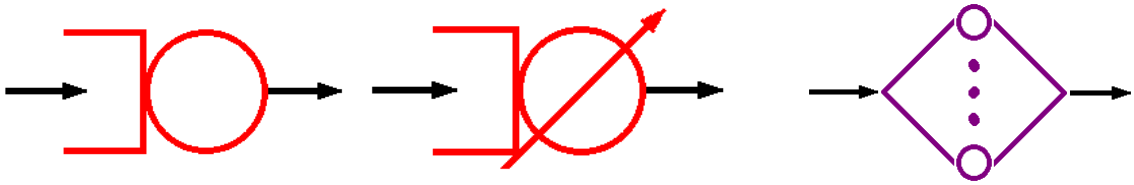
O modelo de servidor central mostrado na Figura 3.5 foi introduzido por Buzen em 1973. A CPU neste modelo é o Servidor Central que é conectado a outros dispositivos. Após o serviço dos dispositivos de E/S serem executados, as tarefas retornam para a CPU para processamento adicional e deixa-o quando a próxima E/S é requerida ou quando a tarefa estiver concluída.

Na modelagem de sistemas computacionais são encontrados três tipos de dispositivos, descritos na Tabela 3.1.

Tabela 3.1 - Os três tipos de dispositivos encontrados na modelagem de sistemas computacionais

Centro de Serviço de Capacidade Fixa	a maioria dos dispositivos têm um único Servidor cujo tempo de serviço não depende do número de tarefas no dispositivo. Tais dispositivos são chamados de centros de capacidade fixa. Por exemplo, a CPU de um sistema pode ser modelado como um centro de serviço de capacidade fixa, Figura 3.6a
Centros de Carga	finalmente, os dispositivos restantes são chamados de centros de carga dependentes de serviços desde que as suas taxas de

	serviço podem depender da carga ou o número de tarefas no dispositivo. A fila M/M/m (com m > 2) é um exemplo de um centro de serviços dependentes de carga. Sua taxa de serviço total aumenta à medida que mais e mais Servidores são usados. Um grupo de ligações paralelas entre dois nós de uma rede de computadores é um exemplo de um centro de serviços dependentes de carga, Figura 3.6b
Centro de Atraso	há dispositivos que não têm filas, as tarefas demandam a mesma quantidade de tempo nestes dispositivos, independentemente do número de tarefas no Servidor na mesma. Tais dispositivos podem ser modelados como um centro com servidores infinitos e são chamados de centros de atraso ou IS (Infinity Server). Um grupo de terminais dedicados geralmente é modelado como um centro de atraso, Figura 3.6c



(a) (b) (c)

Figura 3.6 - Tipos de dispositivos dos modelos de sistemas: a) centro de serviço de capacidade fixa; b) centro de serviços dependentes de carga; e c) um centro de atraso.

A seguir são descritas uma série de técnicas para resolver esses modelos de redes de filas. As técnicas são apresentadas na ordem da sua complexidade. A técnica mais simples é usar a análise operacional.

3.2. Leis Operacionais

Um grande número de problemas de análise de desempenho de sistemas de computador do dia-a-dia pode ser resolvidos usando algumas relações simples que não necessita de quaisquer suposições sobre a distribuição dos tempos de serviço ou de chegadas. Várias destas relações, chamadas Leis Operacionais, foram identificados originalmente por Buzen em 1976 e posteriormente confirmadas por Denning e Buzen em 1978.

A palavra operacional significa medida diretamente, ou seja, as quantidades operacionais são as quantidades que podem ser medidas diretamente durante um período finito de observação. Assim, as suposições operacionalmente testáveis são as hipóteses que podem ser verificadas por meio de medições.

Por exemplo, pode-se verificar por meio de medidas se a hipótese do número de entradas é igual ao número de saídas de um sistema particular. Portanto, esta hipótese, comumente chamada de hipótese do Balanço do Fluxo de Trabalho, é operacionalmente testável.

Por outro lado, não se pode definir, por meio de um conjunto de medição de tempos de serviço observados, se é ou não uma sequência de variáveis aleatórias independentes. Assim, a suposição de independência, apesar de comumente utilizada na modelagem de filas, não é operacionalmente testável.

Por exemplo, considere a visão caixa-preta de um dispositivo i mostrado na Figura 3.7. Se se observar o sistema por um tempo finito T, pode-se medir o número de chegadas A_i, o número de conclusões C_i e o tempo durante este período B_i. Todos estas são quantidades operacionais.

Figura 3.7 - RSF vista como uma caixa-preta.

A partir destas medidas, pode-se ainda obter as seguintes quantidades operacionais:

- Taxa de chegada: λ_i = número de chegadas/tempo = A_i/T
- Rendimento ou Vazão: X_i = número de conclusões/tempo = C_i/T
- Utilização: U_i = tempo ocupado/tempo total = B_i/T
- Tempo médio de serviço: S_i = tempo total do servidor/número de tarefas = B_i/C_i

Note-se que estas quantidades são variáveis operacionais que podem mudar de um período de observação para outro, mas há certas relações que mantêm em cada período de observação. Tais relações são chamadas Leis Operacionais.

3.2.1. Lei da Utilização (U)

Dado o número de conclusões C_i e o tempo ocupado B_i de um dispositivo i durante um período de observação T, tem-se a seguinte relação entre estas variáveis:

$$U_i = \frac{B_i}{T} = \frac{C_i}{T} \times \frac{B_i}{C_i} \quad (3.1)$$

$$U_i = X_i S_i \quad (3.2)$$

A Lei de Utilização, bem como as outras leis operacionais, são quantidades operacionais e não há necessidade de considerar os valores esperados destas variáveis aleatórias ou assumir uma distribuição de probabilidade para eles.

Exemplo 3.1 - Considere o problema de um *gateway* de rede em que pacotes chegam a uma taxa de 125 pacotes por segundo (pps) e o *gateway* leva uma média de dois milissegundos para encaminhá-las.

Vazão: X_i = taxa de saída = velocidade de chegada = 125 pps

Tempo de serviço: S_i = 0,002 s

Utilização: $U_i = X_i S_i = 125 \times 0{,}002 = 0{,}25 = 25\%$

3.2.2. Lei do Fluxo Forçado

Em um modelo aberto, o número de tarefas que sai do sistema por unidade de tempo define sua taxa de transferência. Em um modelo fechado nenhum trabalho, na verdade, deixa o sistema. No entanto, atravessando a ligação do lado de fora (o elo de ligação "Out" para "In" da Figura 3.2) é equivalente à saída do sistema e sua reentrada imediata, e a vazão do sistema é definido como o número de tarefas que atravessam esta ligação por unidade de tempo.

Se no período de observação T o número de tarefas que entram em cada dispositivo é igual ao número de tarefas concluidas, isto é, $A_i = C_i$ Pode-se afirmar que o dispositivo satisfaz a hipótese do Equilíbrio de Fluxo de Trabalho.

Se o período de observação de T é longo, a diferença A_i-C_i é geralmente pequena quando comparado com C_i. Será exato, se o comprimento da fila inicial em cada dispositivo for o mesmo que o comprimento da fila final.

Suponhamos que cada tarefa faz V_i pedidos para o i-ésimo dispositivo do sistema. Se o fluxo de trabalho está em equilíbrio, o número de trabalhos C_0 que atravessa a ligação externa e o número de tarefas C_i concluidas pelo i-ésimo dispositivo estão relacionados por

$$C_i = C_0 V_i \Rightarrow V_i = \frac{C_i}{C_0} \qquad 3.3$$

A variável V_i é a razão entre as tarefas concluidas pelo dispositivo de ordem i e a ligação do lado de fora. É, portanto, chamada relação de visita.

O rendimento do sistema durante o período de observação é:

Vazão do sistema: $X = \frac{trabalhos\ concluidos}{tempo\ total} = \frac{C_0}{T}$

$$U_i = X_i S_i \qquad 3.4$$

A taxa de transferência do i-ésimo dispositivo e o rendimento do sistema são, portanto, relacionados da seguinte forma:

Vazão do dispositivo: $X_i = \frac{C_i}{T} = \frac{C_i}{C_0} \times \frac{C_0}{T} = X V_i$.

A Lei do Fluxo Forçado é aplicada sempre que a hipótese do Balanço do Fluxo de Trabalho é verdadeiro.

É claro que, se o equilíbrio do fluxo de trabalho não é verdadeiro, então ou existem alguns trabalhos que chegaram durante o período de observação, mas não saem ou existiam tarefas no sistema no início do período de observação.

Em qualquer caso, é possível que esses trabalhos ainda não fez todas as V_i visitas ao i-ésimo dispositivo e $C_i \neq C_0 V_i$.

A Lei do Fluxo Forçado relaciona a vazão de dispositivos individuais do sistema.

Combinando a Lei do Fluxo Forçado e a Lei da Utilização obtem-se a utilização do i-ésimo dispositivo: $U_i = X_i V_i = X V_i S_i = X D_i$

Em que $D_i = X V_i$ é a demanda total do serviço no dispositivo i para todas as visitas de uma tarefa.

Esta equação, indica que as utilizações dos vários dispositivos no sistema são proporcionais à demanda total de trabalho D_i no dispositivo. Assim, o dispositivo com o maior D_i tem a maior utilização e é o dispositivo de

estrangulamento.

Exemplo 3.2 - Em um sistema de tempo compartilhado, Figura 3.5, os dados de registro de contabilidade produziu o seguinte perfil para programas do usuário. Cada programa requer 5 segundos de tempo de CPU e faz 80 pedidos de I/O no Disco A e 100 pedidos I/O no Disco B. O tempo médio dos usuários encontrado foi de 18 s. A partir das especificações do dispositivo, foi determinado que o Disco A leva 50 ms para satisfazer um pedido de E/S e o Disco B leva 30 ms por pedido. Com 17 terminais ativos, a taxa de transferência do Disco A observado foi de 15,70 pedidos de I/O por segundo. Qual é o rendimento do sistema e a utilizações do dispositivo.

Simbolicamente, são dadas a seguir:

$D_{CPU} = 5$

$V_A = 8$

$V_B = 10$

$Z = 18$

$S_A = 0{,}050$

$S_B = 0{,}030$

$N = 1$

$X_A = 15{,}70$ tarefas/s

Uma vez que os trabalhos devem visitar a CPU antes de ir para os discos ou terminais, a proporção visita CPU é

$V_{CPU} = V_A + V_B + 1 = 181$

O primeiro passo na análise operacional geralmente é determinar exigências totais de serviço D_i para todos os dispositivos. Neste caso,

$D_{CPU} = 5$

$D_A = S_A V_A = 0{,}050 \times 80 = 4$

$D_B = S_B V_B = 0{,}030 \times 100 = 3$ s

Usando a Lei de Fluxo Forçado, os débitos são

$X = X_A/V_A = 15{,}75/80 = 0{,}1963$ tarefas/

$X_{CPU} = X\ V_{CPU} = 0{,}1963 \times 181 = 35{,}48$ requisições/

$X_B = X\ V_B = 0{,}1963 \times 100 = 19{,}6$ requisições/s

Usando a Lei de Utilização, as utilizações do dispositivo são

$U_{CPU} = X\ D_{CPU} = 0{,}1963 \times 5 = 98$

$U_A = X\ D_A = 0{,}1963 \times 4 = 78{,}4$

$U_B = X\ D_B = 0{,}1963 \times 3 = 58{,}8\%$

3.2.3. Lei de Little

A Lei de Little é também uma Lei Operacional. O pressuposto operacionalmente testável é o do Equilíbrio do Fluxo de Trabalho, o número de entradas é igual ao número de saidas. Pode-se aplicar a Lei de Little e relacionar a fila de comprimento Q_i e o tempo de resposta R_i do i-ésimo dispositivo:

$$Número\ Médio\ de\ Tarefas = Taxa\ Média\ de\ Chegadas \times Tempo\ Médio \quad (3.5)$$

$$Q_i = \lambda_i R_i \quad (3.6)$$

Se o fluxo de trabalho está em equilíbrio, a taxa de chegada é igual à taxa de saída, e pode-se escrever a equação acima como:

$$Q_i = X_i R_i \quad (3.7)$$

Exemplo 3.3 - O comprimento médio da fila no sistema de computador do Exemplo 3.2 foram observadas como sendo 8,88, 3,19 e 1,40 tarefas na CPU, no Disco A e no Disco B, respectivamente. Quais foram os tempos de resposta destes dispositivos?

No Exemplo 3.2, as vazões dos dispositivos foram determinados como sendo:

X_{CPU} = 35,4

X_A = 15,7

X_B = 19,6

A nova informação dada neste exemplo é:

Q_{CPU} = 8,8

Q_A = 3,1

Q_B = 1,40

Usando a Lei de Little, os tempos de resposta dos dispositivos são:

$R_{CPU} = Q_{CPU}/X$ = 8.88/35.48 = 0,250 s

$R_A = Q_A/X_A$ = 3.19/15.70 = 0,203 s

$R_B = Q_B/X_B$ = 1.40/19.6 = 0,071 s

3.2.4. Lei Geral do Tempo de Resposta

Todos os sistemas de tempo compartilhado pode ser dividido em dois subsistemas:

a) O subsistema de terminais
b) O subsistema central, que consiste nos dispositivos restantes, incluindo o processador

O sistema central é compartilhado pelos terminais de usuário.

É interessante notar que a Lei de Little pode ser aplicada a qualquer parte do sistema. O único requisito é que o Fluxo de Trabalho esteja em equilíbrio. Em particular, ele pode ser aplicado para o subsistema central, que dá:

$$Q = X\,R \tag{3.8}$$

Aqui, Q é o número total de tarefas no Servidor do sistema, R é o tempo de resposta do sistema e X é a taxa de transferência do sistema. Dado os comprimentos Q_i nos dispositivos, pode-se calcular Q:

$$Q = Q_1 + Q_2 + Q_3 + \cdots + Q_M = \sum_{i=1}^{M} Q_i \tag{3.9}$$

Substituindo Q_i da anterior obtem-se:

$$X\,R = X_1 R_1 + X_2 R_2 + X_3 R_3 + \cdots + X_M R_M = \sum_{i=1}^{M} X_i R_i \tag{3.10}$$

Dividindo ambos os lados desta equação por X e usando a Lei do Fluxo Forçado, obtem-se

$$R = V_1 R_1 + V_2 R_2 + V_3 R_3 + \cdots + V_M R_M = \sum_{i=1}^{M} V_i R_i \tag{3.11}$$

Esta é a chamada **Lei Geral de Tempo de Resposta**. É possível mostrar que essa lei é válida mesmo se o fluxo de trabalho não esteja em equilíbrio.

Intuitivamente, a lei estabelece que o tempo total gasto por uma tarefa em um servidor é o produto do tempo de visita pelo número de visitas ao servidor e o tempo total no sistema é igual à soma dos tempos nos vários servidores.

Exemplo 3.4 - Seja calcular o tempo de resposta para o sistema de compartilhamento de tempo de Exemplos 3.2 e 3.3.

Para este sistema:

$V_{CPU} = 18$

$V_A = 8$

$V_B = 100$

$R_{CPU} = 0,25$

$R_A = 0,20$

$R_B = 0,071$

O tempo de resposta do sistema é:

$R = R_{CPU}V_{CPU} + R_AV_A + R_BV_B = 0,250 \times 181 + 0,203 + 0,071 \times 80 \times 100 = 68,6$

O tempo de resposta do sistema é 68,6 s.

3.2.5. Lei do Tempo de Resposta Interativo

Em um sistema interativo, os usuários geram pedidos que são atendidos pelo subsistema central e as respostas voltam para o terminal. Depois de um tempo de reflexão Z, os usuários enviam o próximo pedido. Se o tempo de resposta do sistema é R, o tempo total do ciclo de pedidos é R + Z. Cada usuário gera cerca de T/(R+Z) pedidos no período de tempo T.

Se houver N usuários, a Vazão do Sistema X é dada por:

$$X = \frac{Número\ Total\ de\ Solicitações}{Tempo\ Total} \quad (3.12)$$

$$X_i = N\frac{T}{R+Z} \quad (3.13)$$

$$X = \frac{N}{R+Z} \quad (3.14)$$

$$R = \frac{N}{X} - Z \quad (3.15)$$

Esta é a **Lei do Tempo Resposta Interativa**.

Exemplo 3.5 - Para o sistema de tempo partilhado do Exemplo 3.2, pode-se calcular o tempo de resposta, utilizando a Lei do Tempo de Resposta Interativa como se segue:

X = 0,196

N = 1

Z = 18

Portanto, R= 17/0,1963-18 = 86,6-18 = 68,6 s

Isto é o mesmo que o obtido anteriormente no Exemplo 3.4.

3.3. Análise de Gargalos

Uma consequência da Lei do Fluxo Forçado é que a utilização de um dispositivos é proporcional à sua respectiva demanda total de serviço:

$$U_i \propto D_i \tag{3.16}$$

O dispositivo com a maior demanda total de serviço D_i tem a maior utilização e é chamado dispositivo de estrangulamento. Este dispositivo é o principal fator limitante na obtenção de um maior rendimento. Melhorar este dispositivo irá proporcionar o maior retorno em termos de rendimento do sistema. Melhorar outros dispositivos terá pouco efeito sobre o desempenho do sistema. Portanto, identificar o dispositivo gargalo deve ser o primeiro passo em qualquer projeto de melhoria de desempenho.

Suponha que o dispositivo b é o gargalo.

Isto implica que $D_b = D_{max}$ é a mais elevada dentre D_1, D_2, ..., D_M.

Em seguida, a taxa de transferência e os tempos de resposta do sistema são relacionados como se segue:

$$X(N) \leqslant min\left\{\frac{1}{D_{max}}, \frac{N}{D+Z}\right\} \tag{3.17}$$

$$R(N) \geqslant max\left\{D, \frac{N}{D_{max}} - Z\right\} \tag{3.18}$$

Centros de atraso não podem ser dispositivos de estrangulamento. Apenas

centros com Filas devem ser considerados para encontrar o gargalo D_{max}. A Figura 3.8 mostra os limites assintóticos para um caso típico.

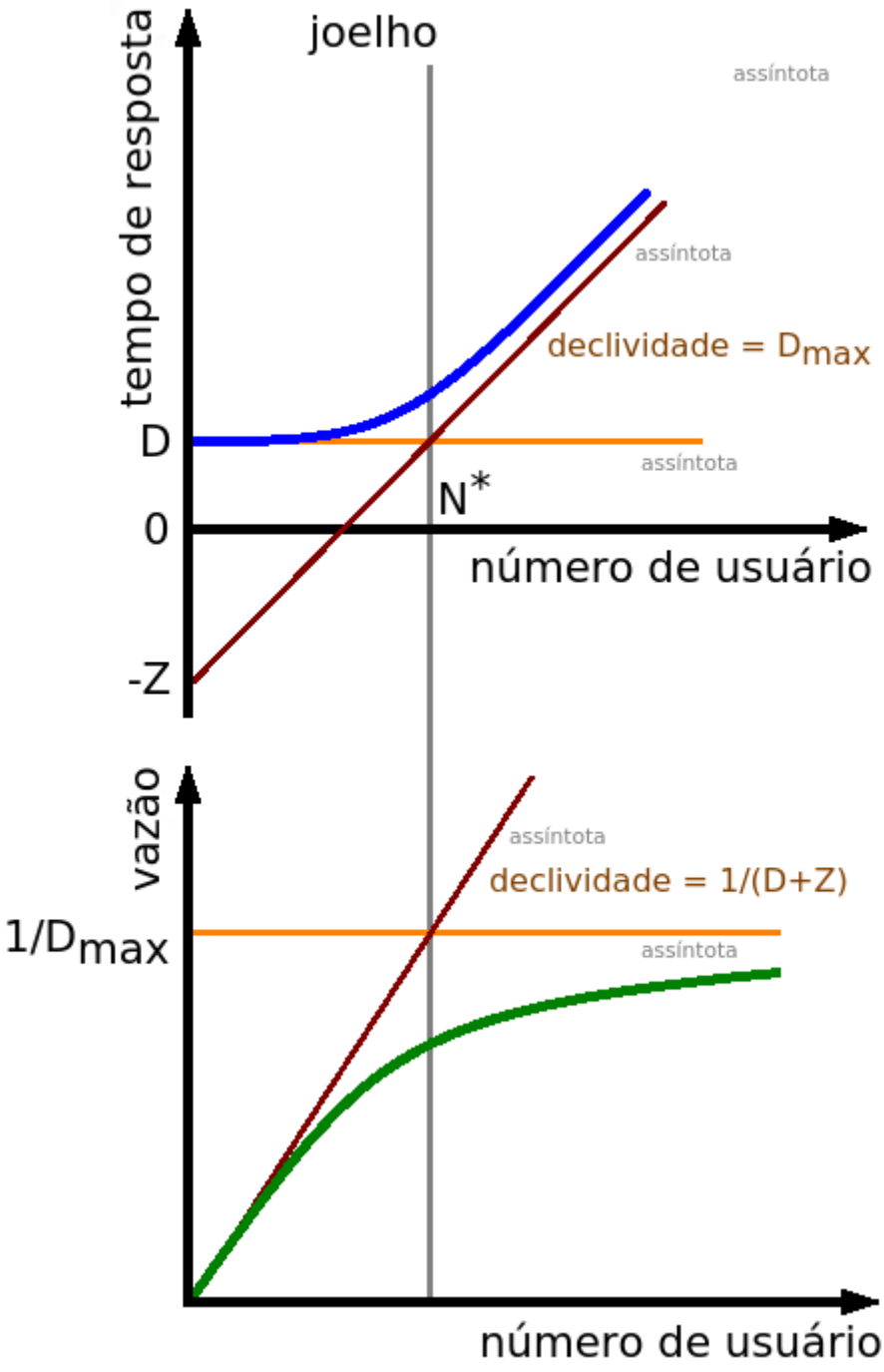


Figura 3.8 - Limites assintóticos típicos.

Os limites tanto do rendimento quanto do tempo de resposta consiste em duas linhas retas. Os limites do tempo de resposta consistem de uma linha reta horizontal (R = D) e uma linha que passa pelo ponto (-Z,O) com uma inclinação D_{max}. Os limites da taxa de transferência consistem de uma

linha horizontal em $X=1/D_{max}$ e a linha que passa pela origem com inclinação 1/(D+Z). O ponto de intersecção das duas linhas é chamado joelho. O joelho é o mesmo para o tempo de resposta e para o rendimento.

O número de tarefas N_J no joelho é dada por:

$$D = N_J D_{max} - Z \quad (3.19)$$

$$N_J = \frac{D+Z}{D_{max}} \quad (3.20)$$

Se o número de tarefas é maior do que N_J então pode-se dizer com certeza que há fila em algum lugar do sistema.

Os limites assintóticos são úteis na prática, uma vez que podem ser facilmente obtidos e explicado para as pessoas que não têm qualquer experiência em teoria de filas ou análise de desempenho.

Exemplo 3.6 - Para o sistema de tempo compartilhado considerado no Exemplo 3.2:

D_{CPU} =

D_A =

D_B = 3, Z = 18

$D = D_{CPU} + D_A + D_B = 5 + 4 + 3 = 12$

$Dmax = D_{CPU} = 5$

Os limites assintóticos são:

$X(N) \leq \min\{1/D_{max}, N/(D+Z)\} = \min\{1/5, N/30\}$

$R(N) \geq \max\{ D, ND_{max}-Z\} = \max\{12, 5N-18\}$

O joelho ocorre é: $12 = 5 N_J - 18$ ou $N_J = (12+18)/5 = 30/5 = 6$

Deste modo, se existirem mais do que seis usuários no sistema, existe, certamente, fila no sistema.

Exemplo 3.7 - Quantos terminais podem ser suportados pelo sistema de tempo compartilhado do Exemplo 3.2, se o tempo de resposta for mantida abaixo de 100 s?

Usando os limites assintóticos sobre o tempo de resposta, temos

$R(N) \geq \max \{12, 5N -18\}$

O tempo de resposta será maior do que 100 se $5N-18 \geq 100$

Isto é, $N \geq 23,6$. Assim, o sistema não pode suportar mais de 23 usuários para o tempo de resposta inferior a 100 s.

Desta forma, completa-se a discussão das Leis Operacionais. Um resumo das das Leis Operacionais é apresentado no Tabela 3.2.

Tabela 3.2 - Resumo das Leis Operacionais

Lei da Utilização	$U_i = X_i S_i = X D_i$
Lei do Fluxo Forçado	$X_i = X V_i$
Lei de Little	$Q_i = X_i R_i$
Lei Geral do Tempo de Resposta	$R = \sum R_i V_i$
Lei do Tempo de Resposta Interativo	$R = \frac{N}{X} - Z$
Limites Assintóticos	$X(N) \leqslant min\{\frac{1}{D_{max}}, \frac{N}{D+Z}\}$ $R(N) \geqslant max\{D, \frac{N}{D_{max}} - Z\}$
Símbolos	
D	soma das demandas de serviço nos dispositivos, ΣD_i
D_i	demanda total de serviço por tarefa do i-ésimo dispositivo, V_i
D_{max}	demanda de serviço no dispositivo gargalo = max{D_i}
N	número de tarefas no sistema
Q_i	número de tarefas no i-ésimo dispositivo
R	tempo de resposta do sistema
R_i	tempo de resposta por tarefa do i-ésimo dispositivo
S_i	tempo de serviço por tarefa do i-ésimo dispositivo
U_i	utilização do i-ésimo dispositivo
V_i	número de tarefas executadas pelo i-ésimo dispositivo
X	rendimento do sistema
X_i	rendimento do i-ésimo dispositivo
Z	tempo aleatório (*think time*)

3.4. Análise do Valor Médio

A Análise do Valor Médio é uma técnica que estende os resultados das Leis Operacionais para a análise de RSF. Esta técnica permite a análise de sistemas de computador que podem ser modeladas como uma RSF fechada. No entanto, os resultados sobre RSF abertas são apresentados primeiramente, uma vez que são mais fáceis de obter e dado que ajudam a compreender os resultados para sistemas fechados.

3.4.1. Análise de Redes Abertas de Filas

Modelos Abertos de Filas são usados para representar sistemas de processamento de transações, tais como sistemas web e servidores de modo geral. Nestes sistemas, a taxa de chegada de transações não é dependente da carga sobre o sistema de computador. As chegadas de transação são modeladas como um processo de Poisson com um taxa média de chegada λ.

Assume-se que todos os dispositivos do sistema de computador podem ser modelados quer como centros de serviços de capacidade fixa (um único servidor com tempo de serviço exponencialmente distribuído) ou centros de atraso (servidores infinitos com o tempo de serviço exponencialmente distribuídos).

Para os centros de serviços com capacidade fixa em uma rede de filas aberta, o tempo de resposta é dada por:

$$R_i = S_i(1+Q_i) \qquad 3.21$$

Uma maneira fácil de entender essa relação é a de considerar uma tarefa marcada que flui através do sistema. Ao chegar no dispositivo i, esta tarefa vê outras Q_i tarefas à frente (incluindo a que está no Servidor) e espera o tempo Q_i S_i. Incluindo o tempo para ele mesma, esta tarefa deverá esperar o tempo de resposta total igual a $S_i(1+Q_i)$. Observe que esta equação não é uma Lei Operacional. Ela assume que o serviço é "sem memória", uma suposição que não é testável operacionalmente. Para calcular o tempo de resposta da tarefa marcada sem a suposição "sem memória" é preciso saber quanto tempo a tarefa que está no Servidor já consumiu.

A equação anterior, combinado as Leis Operacionais, são suficientes para obter valores médios dos parâmetros de desempenho do sistema, como mostrado em seguida.

Assumindo o Equilíbrio de Fluxo de Trabalho, o rendimento do sistema é igual à taxa de chegada: $X = \lambda$.

A taxa de transferência do dispositivo de ordem i, utilizando a Lei do Fluxo Forçado é $X_i = XV_i$.

A utilização do dispositivo de ordem i, utilizando a Lei de Utilização é $U_i = X_i S_i = X V_i S_i = X D_i$.

O tamanho da fila do dispositivo i, usando a lei de Little é

$$Q_i = X_i R_i = X_i S_i(1+Q_i) = U_i(1+Q_i) \rightarrow Q_i = \frac{U_i}{(1-U_i)} \quad .$$

Substituindo esta expressão para Q_i na $R_i = S_i(1+Q_i)$, os tempos de resposta do dispositivo são:

$$Q_i = \frac{U_i}{1-U_i} \rightarrow R_i = S_i(1+Q_i) \rightarrow R_i = \frac{S_i}{1+U_i}$$

Em centros de atraso, existem infinitos Servidores e, portanto, o tempo de

resposta é igual ao tempo de serviço, independente do tamanho da fila. O tamanho da fila, neste caso, indica o número de tarefas recebidas pelo Servidor, pois não há espera. Assim, o tempo de resposta e as equações de comprimento de fila para centros de atraso são:

$$R_i = S_i$$
$$Q_i = R_i X_i = S_i X V_i = X D_i = U_i$$

Note-se que a utilização do centro de atraso representa a média do número de tarefas que o servidor recebe e não precisa ser menor do que 1.

Todas as equações para analisar redes abertas de filas estão resumidas na Tabela 3.3.

Tabela 3.3 -Quadro da análise de RSF abertas

Entradas	
X	taxa de chegada externa, rendimento do sistema
S_i	tempo de serviço por visita ao i-ésimo dispositivo
V_i	número de visitas ao i-ésimo dispositivo
M	número de dispositivos (não incluindo terminais)
Saídas	
Q_i	média do número de tarefas no i-ésimo dispositivo
R_i	tempo de resposta do i-ésimo dispositivo
R	tempo de resposta do sistema
U_i	utilização do i-ésimo dispositivo
N	número médio de tarefas no sistema
Total da demanda de serviços	$D_i = S_i V_i$
Utilização de dispositivo	$U_i = X D_i$
Vazão de dispositivo	$X_i = XV_i$
Tempos de resposta de dispositivo	$X_i = XV_i$ (centros de serviços) S_i (centros de atraso)
Comprimento da fila de dispositivo	$Q_i = \frac{U_i}{(1-U_i)}$ (centros de serviços) U_i (centros de atraso)
Tempo de resposta do sistema	S_i
Número de tarefas no sistema	$N = \sum Q_i$

Exemplo 3.8 - A Figura 3.5 representa um modelo de filas de servidor de arquivos que consiste de uma CPU e dois Discos A e B. Medidas de um sistema distribuído com seis sistemas de clientes que fazem solicitações ao servidor de arquivos produziram os seguintes dados:

Intervalo de observação = 3.600

Número de pedidos de clientes = 10.80

Tempo ocupado na CPU = 1.728

Tempo ocupado no Disco A = 1.512

Tempo ocupado no Disco B = 2.592

Número de visitas (pedidos I/O) para o Disco A = 75.60

Número de visitas (pedidos I/O) para o Disco B = 86.400

X = 10.800/3.600 = 3 pedidos de clientes por segund

V_A = 75.600/10.800 = 7 solicitações de visitas por cliente para o disco

V_B = 86.400/10.800 = 8 solicitações de visitas por cliente para o disco B

V_{CPU} = 1 + 7 + 8 = 16 visitas de clientes para CP

D_{CPU} = 1.728/10.800 = 0,16 s de CPU por solicitação do client

D_A = 1.512/10.800 = 0,14 s do Disco A por solicitação do client

D_B = 2.592/10.800 tempo = 0,24 s do Disco B por solicitação do cliente

S_{CPU} = 0,16/16 = 0,01 s por visita à CP

S_A = 0,14/7 = 0,02 s por visita ao Disco

S_B = 0,24/8 = 0,03 s por visita ao Disco B

Com estes parâmetros de entrada, faz-se a analisar do sistema:

Utilizações de dispositivos usando a Lei de Utilização:

$U_{CPU} = X\ D_{CPU}$ = 3 x 0,16 = 0,4

$U_A = X\ D_A$ = 3 × 0,14 = 0,4

$U_B = X\ D_B$ = 3 x 0,24 = 0,72

Os tempos de resposta do dispositivo usando a Equação (34.2) são

$R_{CPU} = S_{CPU}/(1-U_{CPU})$ = 0,01/(1-0,48) = 0,0192 s

$R_A = S_A/(1-U_A)$ = 0,02/(1-0,42) = 0,0345 s

$R_B = S_B/(1- U_B)$ = 0,03/(1-0,72) = 0,107 s

Tempo de resposta do servidor $R = \Sigma R_i V_i$ = 16×0,0192 + 7×0,0345 + 8×0,107 = 1,406 s

O modelo pode ser usado para responder a algumas das perguntas típicas.

Por exemplo, podemos querer quantificar o impacto das seguintes alterações:

c) Aumentar o número de clientes para 8
d) Usar um cache de Disco B com uma taxa de acerto de 50%, embora aumente a sobrecarga de CPU de 30% e o tempo de serviço do disco B (por I/O) de 10%
e) Usar um servidor de baixo custo com apenas um disco (Disco A) e direcionar todas as solicitações de I/O para ele.

1. Aumentar o número de clientes para 8

Assumindo que os novos clientes farão pedidos semelhantes aos medidos, a taxa de chegada de solicitação vai subir por um fator de 8/6. Como a taxa de chegada foi três pedidos por segundo, com mais clientes se tornará quatro pedidos por segundo. A nova análise é a seguinte:

$X = 4$ pedidos/s

$U_{CPU} = X\ D_{CPU} = 4 \times 0,16 = 0,6$

$U_A = X\ D_A = 4 \times 0,14 = 0,5$

$U_B = X\ D_B = 4 \times 0,24 = 0,96$

$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,01 / (1 - 0,64) = 0,0278$

$R_A = S_A/(1-U_A) = 0,02 / (1 - 0,56) = 0,0455$

$R_B = S_B/(1-U_B) = 0,03 / (1 - 0,96) = 0,75$ s

$R = 16 \times 0,0278 + 7 + 8 \times 0,0455 \times 0,75 = 6,76$ s

Assim, se o número de clientes é aumentado de 6 para 8, o tempo de resposta do servidor irá degradar-se por um factor de 6.76/1.406 = 4,8

2. Usar um cache de Disco B com uma taxa de acerto de 50%, embora aumente a sobrecarga de CPU de 30% e o tempo de serviço do disco B (por I/O) de 10%

A segunda questão requer mudar V_B, S_{CPU} e S_B da seguinte forma:

$V_B = 0,5 \times 8 = 4$

$S_{CPU} = 1,3 \times 0,01 = 0,013 \rightarrow D_{CPU} = 0,208$

$S_B = 1,1 \times 0,03 = 0,033 \rightarrow D_B = 4 \times 0,033 = 0,132$ s

A análise do sistema modificado é a seguinte:

$U_{CPU} = X\ D_{CPU} = 3 \times 0,208 = 0,62$

$U_A = X\ D_A = 3 \times 0,14 = 0,4$

$U_B = X\ D_B = 3 \times 0,132 = 0,396$

$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,013/(1-0,624) = 0,0346$

$R_A = S_A/(1-U_A = 0,02/(1-0,42) = 0,0345$

$R_B = S_B/(1-U_B) = 0,033/(1-0,396) = 0,0546$ s

$R = 16 \times 0,0346 + 0,0345 + 7 \times 4 \times 0,0546 = 1,013$ s

Assim, ao usar um cache para o Disco B, o tempo de resposta do servidor vai melhorar por (1,406-1,013)/1,406 = 28%

3. Usar um servidor de baixo custo com apenas um disco (Disco A) e direcionar todas as solicitações de I/O para ele

A terceira questão requer ajuste V_A e V_B. Assumindo que não há espaço suficiente no Disco A capaz de acomodar todos os arquivos nos dois discos. A análise é a seguinte:

$V_B =$

$V_A = 7 + 8 = 1$

$D_{CPU} = 0,16$ s (como antes

$D_A = 15 \times 0,02 = 0,3$

$U_{CPU} = X\ D_{CPU} = 3 \times 0,16 = 0,4$

$U_A = X\ D_A = 3 \times 0,3 = 0,90$

$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,01/(1-0,48) = 0,0192$

$R = S_A/(1-U_A) = 0,02/(1-0,90) = 0,2$ s

$R = 16 \times 0,0192 + 15 \times 0,2 = 3,31$ s

Assim, se substituir os dois discos por um lado, o tempo de resposta do servidor irá degradar-se por um factor de 3.31/1.406 = 2,35

3.4.2. Análise do Valor Médio

A Análise do Valor Médio (Mean Value Analysis - MVA), como o nome indica, calcula o desempenho médio. O cálculo de variância não é possível por meio desta técnica. A Análise do Valor Médio se aplica a redes com uma variedade de disciplinas de serviço e distribuições de tempo de serviço. No entanto, inicialmente a discussão se limitará aos centros de capacidade fixa de serviço.

Dada uma RSF fechada com N tarefas, Reiser e Lavenberg mostraram em 1980 que o tempo de resposta do dispositivo de ordem i é dado por:

$$R_i(N) = S_i[1 + Q_i(N-1)] \quad (3.1)$$ 3.22

Aqui, $Q_i(N-1)$ é o comprimento médio da fila do dispositivo de ordem i, com N-1 tarefas na rede.

Uma maneira de explicar esta equação é a de considerar o que acontece quando adicionamos uma tarefa de ordem N marcada na rede com N-1 tarefas. Ao chegar ao centro de serviço i, a tarefa marcada vê $Q_i(N-1)$ tarefas à sua frente (incluindo a que está no Servidor) e espera o tempo $Q_i(N-1)S_i$ antes de receber o serviço. Incluindo o tempo de serviço para si mesmo, a tarefa deve esperar o tempo de resposta total igual $S_i[1+Q_i(N-1)]$.

Dado o desempenho de N-1 usuários, a Equação 3.1, juntamente com as Leis Operacionais permitem calcular o desempenho para N usuários.

A partir de N = 0 o desempenho para qualquer número de usuários pode ser calculado de forma iterativa, como mostramos a seguir.

Tendo em conta os tempos de resposta dos dispositivos individuais, o tempo de resposta do sistema, utilizando a Lei do Tempo de Resposta Geral é $R(N) = \sum V_i R_i(N)$

$$R_i(N) = S_i[1 + Q_i(N-1)] \quad (3.1)$$ 3.23

O rendimento do sistema usando a Lei do Tempo de Resposta Interativo é

$$X(N) = \frac{N}{R}(N) + Z \quad (3.2)$$ 3.24

As vazões dos dispositivos medido em termos de tarefas por segundo são

$$X_i(N) = X(N)V_i \quad (3.3)$$ 3.25

Os comprimentos das filas nos dispositivos com N tarefas pela Lei de Little

são

$$Q_i(N) = X_i(N)\,R_i(N) = X(N)\,V_i\,R_i(N) \quad (3.4) \qquad 3.26$$

Estas equações assumem que todos os dispositivos são centros de capacidade fixa enfileirados.

Se um dispositivo é um centro de atraso (servidores infinitos), que não espera pelo serviço, o tempo de resposta é igual ao tempo de serviço, independente do tamanho da fila. O comprimento da fila, neste caso, indica o número de tarefas que o servidor recebe. Assim, a equação de tempo de resposta para os centros de atraso é simplesmente

$$R_i(N) = S_i \quad (3.5) \qquad 3.27$$

As Equações 3.2 e 3.3 para a vazão de dispositivos e comprimentos de fila se aplicam a centros de atraso.

As Equações 3.1 a 3.5 definem uma iteração do MVA. O procedimento é inicializado para N = 0 usuários e $Q_i(0) = 0$. O procedimento completo é descrito no Tabela 3.4.

Tabela 3.4 - Procedimento completo do MVA

Entradas	
N	número de usuários
Z	tempo de pensar
M	número de dispositivos (não incluindo terminais)
S_i	tempo de serviço por visita ao i-ésimo dispositivo
V_i	número de visitas ao i-ésimo dispositivo
Saídas	
X	rendimento do sistema
Q_i	número médio de postos de trabalho no i-ésimo dispositivo
R_i	tempo de resposta do i-ésimo dispositivo
R	tempo de resposta do sistema
U_i	utilização do i-ésimo dispositivo
Inicialização	para i = 1 até M faça $Q_i = 0$
Iteração	
	para n = 1 até N faça
	inicio
	para i = 1 até M faça
	inicio
	Ri = Si(1+Qi) ou Ri = Si
	$R = \Sigma R_i V_i$
	X = N/(Z+R)
	fim
	fim
Vazão do dispositivo	$X_i = X\,V_i$

Utilizações do dispositivo	$U_i = X S_i V_i$

Exemplo 3.9 - Considere o modelo de rede de filas da Figura 3.5, cada usuário faz 10 pedidos do I/O ao Disco A e cinco pedidos I/O ao Disco B. Os tempos de serviço por visita para os Discos A e B são 300 e 200 ms, respectivamente. Cada pedido leva 2 s de tempo de CPU, e o tempo do usuário é de 4 s. Analise o sistema usando MVA.

Parâmetros do sistema:

$S_A = 0,3$, $V_A = 10 \rightarrow D_A =$

$S_B = 0,2$, $V_B = 5 \rightarrow D_B = 1$

$D_{CPU} = 2$, $V_{CPU} = V_A + V_B + 1 = 16 \rightarrow S_{CPU} = 0,125$

$Z = 4$, $N = 20$

Inicialização

Número de usuários: $N = 0$

Comprimentos da Fila nos Dispositivos: $Q_{CPU} = 0$, $Q_A = 0$, $Q_B = 0$

Iteração 1

Número de usuários: $N = 1$

Tempos de resposta do dispositivo:

$R_{CPU} = S_{CPU} (1+Q_{CPU}) = 0,125 (1+0) = 0,12$

$R_A = S_A (1+Q_A) = 0,3(1+ 0) = 0,$

$R_B = S_B (1+Q_B) = 0,2 (1+ 0) = 0,2$

Tempo de resposta do sistema: $R = R_{CPU}V_{CPU}+R_AV_A+R_BV_B = 0,125\times16+0,3\times10+0,2\times5 = 6$

Rendimento do sistema: $X = N/(R+Z) = 1/(6+4) = 0,1$

Comprimentos de fila do dispositivo:

$Q_{CPU} = X R_{CPU} V_{CPU} = 0,1 \times 0,125 \times 16 = 0,$

$Q_A = X R_A V_A = 0,1 \times 0,3 \times 0,3 = 1$

$Q_B = X R_B V_B = 0,1 \times 0,2 \times 5 = 0,1$

Iteração 2

Número de usuários: $N = 2$

Tempos de resposta do dispositivo:

$R_{CPU} = S_{CPU} (1+Q_{CPU}) = 0,125 (1 + 0,2) = 0,1$

$R_A = S_A (1 + Q_A) = 0,3 (1 + 0,3) = 0,3$

$R_B = S_B(1 + Q_B) = 0,2 (1 + 0,1) = 0,22$

Tempo de resposta do sistema: $R = R_{CPU}V_{CPU}+R_AV_A+R_BV_B = 0,15\times16+0,39\times10+0,22\times5 = 7,4$

Rendimento do sistema: $X = N/(R+Z) = 2/(7,4+4) = 0,175$

Comprimentos de fila do dispositivo:

$Q_{CPU} = X R_{CPU} V_{CPU} = 0,175 \times 0,15 \times 16 = 0,42$

$Q_A = X\ R_A V_A = 0{,}175 \times 0{,}39 \times 10 = 0{,}68$

$Q_B = X\ R_B\ V_B = 0{,}175 \times 0{,}22 \times 5 = 0{,}193$

As iterações podem continuar para valores mais elevados de N. Eles podem ser facilmente implementados usando qualquer pacote de software de planilha eletrônica. Os tempos de resposta, rendimentos e comprimentos de fila no final destas iterações estão listados na Tabela 3.5.

Tabela 3.5 - Resultados da MVA para o Exemplo 3.9

N	Tempo de Resposta				Vazão do Sistema	Comprimento da Fila		
	CPU	Disco A	Disco B	Sistema		CPU	Disco A	Disco B
2	0.150	0.390	0.220	7.400	0.175	0.421	0.684	0.193
3	0.178	0.505	0.239	9.088	0.229	0.651	1.158	0.273
4	0.206	0.647	0.255	11.051	0.266	0.878	1.721	0.338
5	0.235	0.816	0.268	13.256	0.290	1.088	2.365	0.388
6	0.261	1.009	0.278	15.659	0.305	1.275	3.081	0.424
7	0.284	1.224	0.285	18.216	0.315	1.433	3.858	0.449
8	0.304	1.457	0.290	20.888	0.321	1.564	4.684	0.466
9	0.321	1.705	0.293	23.647	0.326	1.670	5.551	0.477
10	0.334	1.965	0.295	26.470	0.328	1.752	6.450	0.485
11	0.344	2.235	0.297	29.340	0.330	1.816	7.374	0.490
12	0.352	2.512	0.298	32.245	0.331	1.865	8.318	0.493
13	0.358	2.795	0.299	35.176	0.332	1.901	9.276	0.496
14	0.363	3.083	0.299	38.126	0.332	1.928	10.245	0.497
15	0.366	3.374	0.299	41.089	0.333	1.948	11.223	0.498
16	0.369	3.667	0.300	44.064	0.333	1.963	12.207	0.499
17	0.370	3.962	0.300	47.045	0.333	1.974	13.195	0.499
18	0.372	4.259	0.300	50.032	0.333	1.981	14.187	0.499
19	0.373	4.556	0.300	53.022	0.333	1.987	15.181	0.500
20	0.373	4.854	0.300	56.016	0.333	1.991	16.177	0.500

Note que o MVA é aplicável apenas se a rede é em forma de produto. Isto significa que a rede deve satisfazer as condições de Equilíbrio do Fluxo de Trabalho. Além disso, a análise, como apresentado aqui, assume que todos os centros de serviço ou são centros de capacidade fixa de serviços ou centros de atraso. Em ambos os casos, assumimos os tempos de serviço é exponencialmente distribuídos.

```
int main( void )
```

```
 float Sa = 0.001
 Va = 1.0
 Da = 0.01
 Sb = 0.001
 Vb = 1.0
 Db = 0.01
 Dcpu = 0.01
 Vcpu = Va + Vb + 1.0
 Scpu = 0.001
 Z = 4.0
 Qcpu = 0.0
 Qa = 0.0
 Qb = 0.0
 Rcpu, Ra, Rb, R, X
 int N = 30, n

printf( "------------------------------------------------------------------
----\n")
 printf( " n\tRcpu\tRa \tRb \tR \tX \tQcpu\tQa \tQb \n")

printf( "------------------------------------------------------------------
----\n")
 for( n = 1; n <= N; n++ )
 Rcpu = Scpu*(1.0 + Qcpu)
 Ra = Sa *(1.0 + Qa )
 Rb = Sb *(1.0 + Qb )
 R = Rcpu*Vcpu + Ra*Va + Rb*Vb
 X = n/(R+Z)
 Qcpu = X*Rcpu*Vcpu
 Qa = X*Ra*Va
 Qb = X*Rb*Vb
 printf("%4d %8.3f %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f %7.3f\n",
n,Rcpu,Ra,Rb,R,X,Qcpu,Qa,Qb)
printf( "------------------------------------------------------------------
----\n")
 return 0
}
```

3.5. Outras Redes de Filas

A rede de enfileiramento mais simples é uma série de k Sistemas de Fila M/M/1 com tempo de serviço exponencial e chegadas de Poisson, conforme mostrado na Figura 3.9.

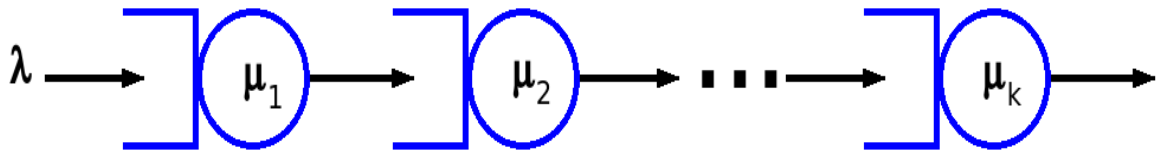


Figura 3.9 - Rede de enfileiramento simples com k Sistemas de Fila M/M/1.

As tarefas que deixam um Sistema de Fila imediatamente entram no Sistema de Fila seguinte. Pode-se mostrar que cada Sistema de Fila individual nesta série pode ser analisado independentemente dos outros Sistemas de Fila. Cada Sistema de Filas tem a taxa de chegada λ, bem

como uma taxa de saída μ. Se μ_i é a taxa de serviço do SF_i, então a utilização do i-ésimo SF $\rho_i = \lambda/\mu_i$. A probabilidade de n tarefas no i-ésimo SF, $p_i(n_i) = (1-\rho_i)\rho^{i \times n}$

A probabilidade conjunta dos comprimentos de fila de k SF pode ser calculada simplesmente multiplicando as probabilidades individuais, por exemplo:

$$P(n_1, n_2, n_3, \cdots, n_k) = (1-\rho_1)\rho_1^{1xn}(1-\rho_2)\rho_2^{2xn}\cdots(1-\rho_k)\rho_k^{kxn} = \prod_{i=1}^{k}(1-\rho_i)\rho_i^{ixn} \quad 3.28$$

$$P(n_1, n_2, n_3, \cdots, n_k) = p_1(n_1)\,p_2(n_2)\,p_3(n_3)\cdots p_k(n_k) = \prod_{i=1}^{k} p_i(n_i) \quad 3.29$$

Esta rede de filas é, portanto, uma rede em forma de produto. Em geral, esta fórmula se aplica a qualquer RSF na qual a expressão para a probabilidade de equilíbrio tem a seguinte forma:

$$P(n_1, n_2, n_3, \cdots, n_k) = \frac{1}{G(N)} \prod_{i=1}^{k} f_i(n_i) \quad 3.30$$

Quando $f_i(n_i)$ é uma função do número de tarefas no SF de ordem i, G(N) é uma constante de normalização e é uma função do número total de tarefas no sistema, N.

Exemplo 3.10 - Considere um sistema fechado com dois SF e N tarefas que circulam entre as filas, como mostrado na Figura 3.10. Ambos os Servidores têm um tempo de serviço exponencialmente distribuídos. Os tempos médios de serviço são 2 e 3 s, respectivamente.

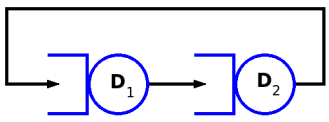


Figura 3.10 - Rede fechada de dois SF.

A probabilidade de ter n_1 tarefas na primeira fila e $n_2 = N-n_i$ tarefas na segunda Fila pode ser demonstrado que é:

$$P(n_1, n_2) = \frac{2^{n_1} 3^{n_2}}{3^{N+1} - 2^{N+1}} \qquad 3.31$$

Neste caso, a constante de normalização $G(N) = 3^{N+1} - 2^{N+1}$. As probabilidades de estado são produtos de funções do número de tarefas nas filas. Assim, trata-se de uma rede na forma do produto.

Redes na forma de produto são mais fáceis de analisar do que redes de outras formas. O conjunto de redes que têm uma solução de forma produto está sendo continuamente ampliado pelos pesquisadores. A primeira delas foi modelada por Jackson que mostrou em 1963 que o método de calcular probabilidade conjunta acima é válida para qualquer rede arbitrária aberta de filas de k SF com tempos de serviço exponencialmente distribuídos, Figura 3.8.

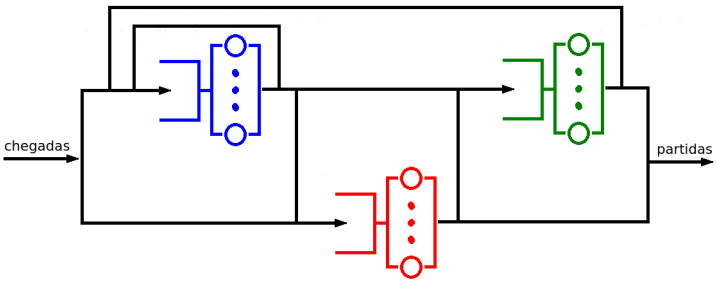


Figura 3.11 - Caso geral de RSF aberta.

Em particular, se cada um dos SF possui um único Servidor, a distribuição do tamanho da fila é dada pela equação:

$$P(n_1, n_2, n_3, \cdots, n_k) = p_1(n_1)\, p_2(n_2)\, p_3(n_3) \cdots p_k(n_k) = \prod_{i=1}^{k} p_i(n_i) \qquad 3.32$$

No entanto, não é correto supor que cada SF seja M/M/m independente e com processo de chegada Poisson. Em geral, o fluxo interno de tais redes não é Poisson. Particularmente, se não houver qualquer *feedback* na rede, de modo que as tarefas possam voltar aos Servidores anteriormente visitados, os fluxos internos não são Poisson. É surpreendente que, apesar dos fluxos não serem Poisson, os SF são separáveis e podem ser analisados como se fossem SF M/M/m independentes.

Os resultados de Jackson foram mais tarde estendido para redes fechadas por Gordon e Newell em 1967. Eles mostraram que as RSF fechadas com k

Servidores e com tempos de serviço exponencialmente distribuídos também têm uma solução na forma do produto. Baskett, Chandy, Muntz e Palacios mostraram em 1975 que as soluções na forma de produto existem para uma classe mais ampla de redes.

4. Ferramentas Estatísticas

Os resultados de medições e simulações podem ser diferentes a cada repetição do experimento. A variação nos resultados de simulações é inerente ao processo devido ao uso de números aleatórios. Ao passo que a variação em medições se deve ao processo experiemental. Ao comparar alternativas é necessário considerar a variabilidade dos resultados. A simples comparação de médias pode levar a resultados insatisfatórios. A análise de dados produz resultados e sua interpretação é a base para o processo de tomada de decisão.

Um dos passos importantes em todos os estudos de avaliação de desempenho é a apresentação dos resultados finais. O objetivo final da análise de desempenho é ajudar na tomada de decisões. Uma análise, cujos resultados não podem ser compreendidos por tomadores de decisão é de pouca valia. É da responsabilidade do analista garantir que os resultados da análise sejam encaminhados para os tomadores de decisão tão clara e simples quanto possível. Isto requer o uso prudente de palavras, imagens e gráficos para explicar os resultados e a análise.

4.1. Sumarização

Na forma mais condensada, um único número pode ser apresentado para dar a característica fundamental de um conjunto de dados. Este único número é chamado a média geral dos dados. Para ser significativo, ele deve ser representativo de uma grande parte do conjunto de dados

Três alternativas populares para resumir uma amostra são especificar sua média, mediana e moda. Estas medidas são o que os estatísticos chamam de índices de tendência central. O nome é baseado no fato que estas medidas especificam a localização do centro de distribuição das observações na amostra.

A média da amostra é obtida tomando a soma de todas as observações e dividindo esta soma pelo número de observações na amostra. A mediana da amostra é obtida através da organização das observações em uma

ordem, e tomando a observação de que está no meio da série. Se o número de observações é ímpar, a média dos dois valores médios é usado como a mediana. A moda da amostra é o ponto central do pico do histograma dos dados amostrais. Para as variáveis categóricas moda é dado pela categoria de maior frequência.

A palavra amostra nos nomes destes medidas significa o fato de que os valores são baseados em apenas uma amostra. No entanto, se está claro a partir do contexto que a discussão é sobre uma única amostra, e não há nenhuma ambiguidade, os nomes mais curtos média, mediana, moda pode ser usado.

A média e a mediana sempre existem e são únicas. Dado qualquer conjunto de observações, a média e mediana pode ser determinada. A moda, por outro lado, pode não existir. Um exemplo disto seria se todas as observações forem iguais. Além disso, mesmo se a moda existir, ela pode não ser única. Pode haver mais de uma moda, ou seja, pode haver mais do que um pico no histograma.

4.2. Selecionando entre a média, mediana e moda

Um erro comum é especificar o índice de tendência central errado. O fluxograma da Figura 4.1 mostra um conjunto de diretrizes para selecionar a medida de tendência central adequada.

A primeira consideração é o tipo de variável. Se a variável é categórica, a moda é a única medida, a que descreve adequadamente seus dados. Um exemplo de dados categóricos é o tipo de processador em várias estações de trabalho. Uma declaração como "o processador mais frequentemente usado em estações de trabalho é o Intel Xeon" faz sentido. A média ou mediana do tipo de processador não tem significado.

A segunda consideração na escolha do índice é saber se o total de todas as observações é de interesse. Se sim, então a média é um indicador adequado da tendência central. Por exemplo, o tempo total da CPU durante cinco consultas é um número significativo. Por outro lado, se contar o número de janelas na tela durante cada consulta, o número total de janelas durante cinco consultas não parece ser significativo. Se o total for de interesse, deve-se especificar a média. Se o total não é de interesse, a escolha é entre a mediana e a moda. Se o histograma é simétrico e unimodal, a média, mediana e moda são todas iguais e assim não importa qual seja especificada.

Se o histograma é enviesado, a mediana é mais representativa de uma observação típica do que a média. Por exemplo, o número de unidades de disco em estações de trabalho de engenharia deverá ter distribuição desigual, e, portanto, é adequado especificar o número mediano. Uma maneira simples de determinar a assimetria para as amostras pequenas é

examinar a relação entre o máximo e mínimo, x_{max}/x_{min}, das observações. Se a razão é grande, os dados são distorcidos.

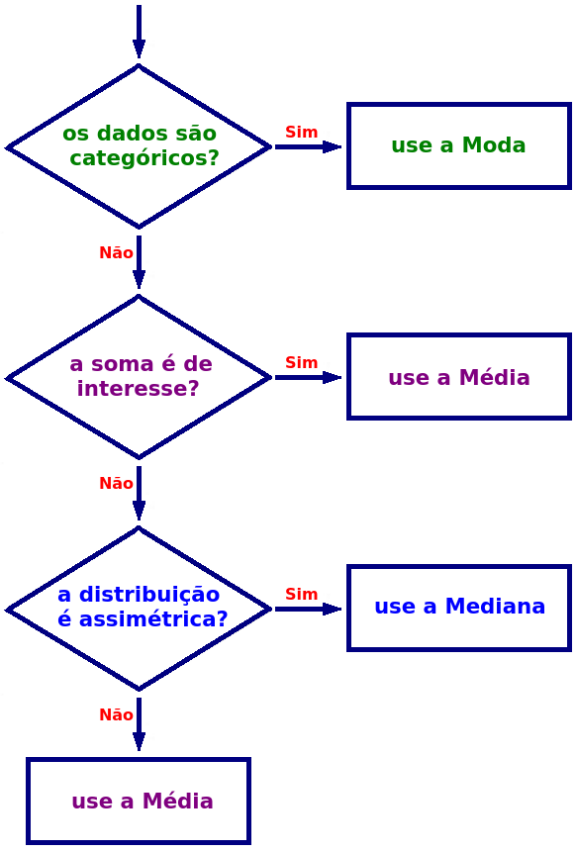


Figura 4.1 - Selecionando entre a média, meadian e moda.

A seguir, exemplos de seleções de índices de tendência central:

- recurso do sistema mais utilizado: recursos são categóricos e, portanto, a moda deve ser usada
- duração: tempo total é de interesse e assim a média é a escolha adequada
- carga em um computador: a mediana é preferível devido a uma distribuição altamente enviesada
- configuração média: medianas do número de dispositivos de memória, tamanhos, e número de processadores são usadas para especificar a configuração de um modo geral, devido à assimetria da distribuição

4.2.1. Resumindo a Variabilidade

Dado um conjunto de dados, resumi-lo por um único número raramente é suficiente. É importante incluir a variabilidade no resumo dos dados. Isto porque dado dois sistemas com o mesmo desempenho médio, um pode variar muito em torno de sua média e outro pouco. Por exemplo, a Figura 4.2 mostra histogramas dos tempos de resposta de dois sistemas. Ambos têm o mesmo tempo de resposta médio de 2 s. No caso (a), o tempo de

resposta está sempre próximo do seu valor médio, enquanto que, no caso (b), o tempo de resposta pode variar de 1 s a 1 minuto, porexemplo. Qual sistema é o preferido? A maioria das pessoas prefere o sistema com baixa variabilidade.

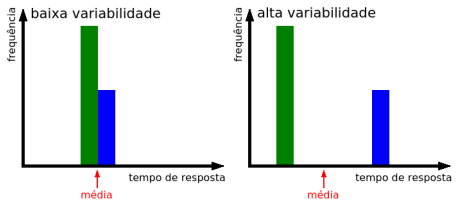


Figura 4.2 - Histogramas de tempos de resposta de dois sistemas.

Variabilidade é especificado por meio de uma das seguintes medidas, que são chamados de índices de dispersão:

- Alcance - diferença entre os valores máximo e mínimo observados
- Desvio padrão ou Variância
- Percentis 10-90
- Alcance entre interquantil
- Desvio médio absoluto

O intervalo de valores pode ser facilmente calculado, por meio dos valores mínimo e máximo. A variabilidade é medida pela diferença entre os valores máximo e o mínimo. Quanto maior a diferença, maior é a variabilidade. Na maioria dos casos, a faixa não é muito útil. O mínimo muitas vezes chega a ser zero e o máximo chega a ser um *outlier*, longe de valores típicos. A menos que haja uma razão para a variável ser delimitada entre dois valores, o valor máximo aumenta com o aumento do número de observações, o mínimo continua a diminuir com o número de observações, e não há nenhum ponto estável que dá uma boa indicação da faixa real. A conclusão é que o intervalo é útil se, e somente se, existe uma razão para acreditar que a variável é limitada. A faixa dá a melhor estimativa desses limites.

A variância de uma amostra de n observações $\{x_1, x_2,..., x_n\}$ é calculada como se segue

$$s^2=\frac{1}{n-1}\sum_{i=1}^{n}(x_i-x_m)^2 \qquad 4.1$$

com $x_m=\frac{1}{n}\sum_{i=1}^{n}x_i$.

A quantidade s^2 é chamada variância da amostra e sua raiz quadrada s é chamado desvio padrão da amostra. A palavra amostra pode ser descartada se não há ambiguidade e é claro a partir do contexto que as quantidades se referem a apenas uma amostra. Observe que no cálculo da

variância, a soma dos quadrados $\sum(x_i-x_m)^2$ é dividida por n-1 e não n. Porque apenas n-1 das n diferenças (x_i-x_m) são independentes. O número de termos independentes em uma soma é também chamado de seus graus de liberdade. Na prática, o principal problema com a variância é que é expressa em unidades que são o quadrado das unidades das observações. A alteração da unidade de medida tem um efeito sobre a magnitude numérica ao quadrado da variância. Por este motivo, é preferível usar o desvio padrão. É na mesma unidade que a média, o que nos permite compará-lo com a média. Assim, se o tempo de resposta médio é de 2 s e o desvio padrão é de 2 s, há uma variabilidade considerável, por outro lado, se o desvio padrão for de 0,2 s, o mesmo seria considerado significaticamente pequeno

De fato, a razão entre o desvio padrão da média ou o coeficiente de variação (COV), é ainda melhor porque leva a escala de medição (unidade de medida), em consideração a variabilidade. Um COV igual a 5 é grande já um COV igual 0,2 (ou 20%) é pequena, não importa qual seja a unidade

Percentis são um meio popular de também especificar a dispersão. Especificando o percentual de 5 e 95-percentil de uma variável tem o mesmo impacto que especificar seu mínimo e máximo. No entanto, isso pode ser feito por qualquer variável, mesmo para as variáveis sem limites. Expressa como uma fração quando entre 0 e 1 (em vez de uma percentagem), os percentis são também chamados de quantil. Assim 0,9-quantil é o mesmo que o percentil-90. Os percentis múltiplos de 10% são chamados de decis. Assim, o primeiro decil é o 10-percentil, o segundo decil é o 20-percentil e assim por diante. Quartis divide os dados em quatro partes iguais a 25, 50 e 75%. Assim, 25% das observações são inferiores ou iguais às Q_1, primeiro quartil, 50% das observações são menores ou iguais ao segundo quartil Q_2 e 75% são inferiores ou iguais a Q_3, terceiro quartil. Observe que o segundo quartil Q_2 também é a mediana. O α-quantil pode ser estimada pela classificação das observações e tendo a posição do [(n-1)α +1]-ésimo elemento do conjunto ordenado. Aqui, [.] é utilizado para designar o arredondamento para o número inteiro mais próximo. Para quantidades exatamente a meio caminho entre dois inteiros, use o menor número inteiro.

O intervalo entre Q_3 e Q_1 é chamada amplitude interquartil de dados (IQR - *Interquartile Range*). A metade deste intervalo é chamado de Semi-IQR (SQIR), Isto é,

$$SQIR=\frac{Q_3-Q_1}{2}=\frac{x_{0.75}-x_{0.25}}{2} \qquad 4.2$$

Outra medida de dispersão é o desvio médio absoluto, s_a, que é calculado da seguinte forma $s_a=\frac{1}{n}\sum_{i=1}^{n}|x_i-x_m|$.

A principal vantagem do desvio médio absoluto sobre o desvio padrão é que não são necessárias nem multiplicação e nem raiz quadrada.

Entre os índices de dispersão anteriores, o alcance é consideravelmente afetada por valores discrepantes. A variância da amostra também é afetada por valores extremos, mas o efeito é menor do que na alcance. O desvio médio absoluto é o próximo na resistência a *outliers*. O alcance entre interquantil é muito resistente a *outliers*. É preferível o desvio padrão pelas mesmas razões que a mediana é o preferido para a média. Assim, se a distribuição é muito assimétrica, valores extremos são altamente propensos e a SIQR é o mais representativo da dispersão nos dados do que o desvio padrão. Em geral, o SIQR é usado como um índice de dispersão sempre que a média é utilizada como índice de tendência central.

Finalmente, deve ser mencionado que todos os índices de dispersão anteriores aplicam-se apenas para os dados quantitativos. Para dados qualitativos (categóricos), a dispersão pode ser especificada, dando o número de categorias mais frequente que o percentual de dados compreendem, por exemplo, os 90% superior.

Tabela 4.1 - Sumarizando dados observados

Dados	amostra com n observações $\{x_i\}$, i ∈ [1,n]
1. Média Aritmética da amostra	$m = \frac{1}{n}\sum_{i=1}^{n} x_i$
2. Média Geométrica da amostra	$m_g = \sqrt[n]{\prod_{i=1}^{n} x_i}$
3. Média Harmônica da amostra	$m_h = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$
4. Mediana da amostra	$m_d = x_{\frac{n-1}{2}}$, se n ímpar $m_d = \frac{1}{2}\left(x_{\frac{n}{2}} + x_{\frac{n+1}{2}}\right)$, se n par x_i é a i-ésima observação da amostra ordenada.
5. Moda da amostra	observação mais frequente (dados categóricos)
6. Variância da amostra	$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - m)^2$
7. Desvio padrão da amostra	$s = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} (x_i - m)^2}$
8. Coeficiente de Variação	$COV = \frac{s}{m}$
9. Coeficiente de Assimetria	$CA = \frac{1}{ns^3}\sum_{i=1}^{n} (x_i - m)^3$
10. Amplitude	diferença entre os valores máximo e mínimo
11. Percentis: p-percentil	$x_p = x_{\lfloor 1+(n-1)p \rfloor}$

12. Semi-IQR	$SQIR = \frac{Q_3 - Q_1}{2} = \frac{x_{0.75} - x_{0.25}}{2}$
13. Desvio médio absoluto	$s_a = \frac{1}{n}\sum_{i=1}^{n} \lvert x_i - m \rvert$

O exemplo a seguir ilustra o cálculo de percentis e SIQRs

Exemplo 4.1 - Em um experimento, que foi repetida 32 vezes, o tempo de CPU medida foi encontrados são {3.1, 4.2, 2.8, 5.1, 2.8, 4.4, 5.6, 3.9, 3.9, 2.7, 4.1, 3.6, 3.1, 4.5, 3.8, 2.9, 3.4, 3.3, 2.8, 4.5, 4.9, 5.3, 1.9, 3.7, 3,2, 4,1, 5,1, 3,2, 3,9, 4,8, 5,9, 4,2}

O conjunto ordenado é {1.9, 2.7, 2.8, 2.8, 2.8, 2.9, 3.1, 3.1, 3.2, 3.2, 3.3, 3.4, 3.6, 3.7, 3.8, 3.9, 3.9, 3.9, 4.1, 4.1, 4.2, 4.2, 4.4 , 4.5, 4.5, 4.8, 4.9, 5.1, 5.1, 5.3, 5.6, 5.9}

Então

O percentil 10 é dada por [1+(31)(0,10)] = elemento 4 = 2,8

O percentil 90 é dada por [1+(31)(0,90)] = elemento 29 = 5,1

O primeiro quartil Q1 é dada por [1+(31)(0,25)] = elemento 9 = 3,2

Dada a mediana por Q2 é [1+(31)(0,50)] = elemento 16 = 3,9

O Q3 terceiro quartil é dado por [1+(31)(0,75)] = elemento 24 = 4,5

Assim, SQIR = $(Q_3\text{-}Q_1)/2$ = (4.5-3.2)/2 = 0.65.

4.3. Resumindo Dados por Meio de Gráficos

Gráficos como gráficos de linhas, de barras, de setores e histogramas são comumente usados para apresentar resultados de desempenho. Além disso, há uma série de gráficos que foram desenvolvidos especificamente para a análise de desempenho de sistemas de computador, estes são gráficos de Gantt, gráficos Kiviat e gráfico Schumacher.

Há uma série de razões pelas quais um gráfico pode ser usado para apresentação de dados no lugar de uma explicação textual. Primeiro de tudo, uma imagem vale mais que mil palavras. Um gráfico economiza tempo dos leitores e apresenta a mesma informação de forma mais concisa. Também pode ser usado para o interesse do leitor. A maioria dos leitores acha mais fácil ver os gráficos para captar rapidamente os principais pontos do estudo e ler o texto apenas para obter mais detalhes. Um gráfico também é uma boa maneira de enfatizar ou esclarecer um ponto, para reforçar uma conclusão e para resumir os resultados de um estudo.

O Tabela 4.2 apresenta uma lista de verificação para tornar mais fácil verificar o emprego adequado de gráficos. A lista é organizada de modo que um "sim" como resposta para cada questão, em geral, leva a um gráfico melhor. No entanto, em alguns casos, um analista pode conscientemente decide não seguir uma sugestão se isso ajuda em

transmitir a mensagem pretendida.

Na prática, é necessário fazer várias tentativas antes de chegar ao gráfico final. Várias faixas de escala diferentes e pares de variável {x,y} devem ser julgados, e o gráfico que apresenta a mensagem mais precisamente, simplesmente, de maneira concisa e logicamente deve ser escolhido.

Tabela 4.2 - Checklist para bons gráficos

1. Os dois eixos de coordenadas são mostrados e rotulados?
2. Os rótulos dos eixos são de auto-explicativos e concisos?
3. As escalas e divisões são mostradas em ambos os eixos?
4. Os valores mínimo e máximo do intervalo dos eixos são adequado para apresentar o máximo de informações?
5. O número de curvas é razoavelmente pequeno? (Um gráfico de linha geralmente não deve ter mais de seis curvas).
6. Todos os gráficos usam a mesma escala? (Escalas múltiplas no mesmo gráfico são confusas)
7. Há curva que pode ser removida sem reduzir as informações?
8. As curvas em um gráfico de linhas são rotuladas individualmente?
9. As células em um gráfico de barras estão individualmente rotuladas?
10. Todos os símbolos no gráfico estão acompanhados de explicações textuais?
11. As linhas que se cruzam são desenhadas utilizando padrões de linhas diferentes para evitar a confusão?
12. As unidades de medida estão indicadas?
13. A escala horizontal cresce da esquerda para a direita?
14. A escala vertical cresce de baixo para cima?
15. As linhas de grade auxiliam na leitura das curvas? (Se não, as linhas de grade não deve ser mostradas)
16. A figura como um todo tornam as informações disponíveis para o leitor?
17. As escalas são contíguas? (Quebras na escala devem ser evitadas ou claramente demonstradas)
18. A ordem das barras em um gráfico de barras está sistematizada? (Alfabética, temporal, ou ordenação do melhor para o pior é preferível em comparação à colocação aleatória)
19. Se o eixo vertical representa uma quantidade aleatória, são mostrados os intervalos de confiança?
20. Não existem curvas, símbolos ou textos sobre o gráfico que pode ser removido sem afetar as informações?
21. Existe um título para o gráfico?
22. O título do gráfico é autoexplicativo e conciso?
23. Para gráficos de barras com intervalo de classes desiguais, a área e a largura representam a frequência e a intervalo, respectivamente?
24. As variáveis plotadas neste gráfico dão mais informação do que outras alternativas?
25. O gráfico claramente comunica a mensagem pretendida?
26. A figura é referenciada e discutida no texto do relatório?

4.4. Modelos de Regressão

Entre os modelos estatísticos utilizados por analistas, os modelos de regressão são os mais comuns. Um modelo de regressão permite estimar ou prever uma variável aleatória como uma função de várias outras variáveis. A variável estimada é chamado a variável resposta e as

variáveis utilizadas para prever a resposta são chamados de variáveis de previsão, os preditores ou fatores. A análise de regressão assume que todas as variáveis de previsão são quantitativos para que as operações aritméticas, como adição e multiplicação sejam significativas.

O objetivo de discutir modelos de regressão é duplo. Primeiro, o de destacar os erros que comumente os analistas cometem ao usar tais modelos. Em segundo lugar, os conceitos utilizados em modelos de regressão, como intervalos de confiança para os parâmetros do modelo, são aplicáveis a outros tipos de modelos. Em particular, o conhecimento desses conceitos é necessária para compreender a análise de projetos experimentais a serem discutidos.

Embora as técnicas de regressão podssam ser utilizada para desenvolver uma variedade de modelos lineares e não lineares, o seu uso mais comum é para encontrar o melhor modelo linear. Tais modelos são chamados de modelos de regressão linear. Para simplificar o problema, inicialmente, limitamos nossa discussão para o caso de uma única variável de previsão. Devido à sua simplicidade, tais modelos são chamados de modelos de regressão linear simples.

O primeiro problema no desenvolvimento de um modelo de regressão é definir o que se entende por um bom modelo e um modelo ruim. A Figura 4.3 mostra três exemplos de dados medidos e as tentativas de ajuste de modelos lineares. Os dados de medição é mostrado por pontos dispersos, enquanto que o modelo é mostrado por uma linha reta. A maioria das pessoas concorda que o modelo nos dois primeiros casos, parece razoavelmente perto dos dados, enquanto que para o terceiro não parece ser um bom modelo. O que é bom sobre os primeiros dois modelos? Uma resposta possível é a de que a linha de modelos, nos dois primeiros casos, está perto de observações mais do que no terceiro caso. Assim, é óbvio que a qualidade do modelo deve ser medida pela distância entre os pontos observados e as linhas de modelo. A próxima etapa, então, é medir a distância.

Os modelos de regressão tentam minimizar a distância medida na vertical entre o ponto de observação e a linha do modelo (ou curva). A motivação para isso é a seguinte. Dado qualquer valor da variável x preditor, pode-se estimar a resposta correspondente utilizando o modelo linear, simplesmente lendo o valor de y na linha de modelo em dado valor x. O segmento de linha que une esta "previsão" e o ponto observado vertical já que ambos os pontos têm a mesma coordenada x. O comprimento do segmento de linha é a diferença entre a resposta observada e a resposta prevista. Isto é chamado de erro residual da modelagem, ou simplesmente erro. Os termos resíduo e erro são utilizados alternadamente.

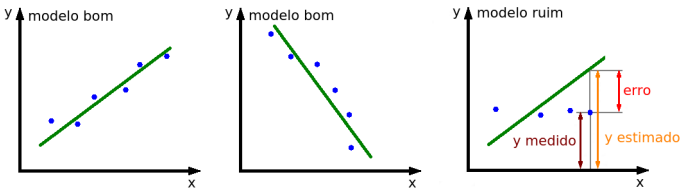


Figura 4.3 - Modelos de regressão bons e ruins.

Alguns dos erros são positivos, porque a resposta estimada é menor do que a resposta observada enquanto que outros são negativos. Uma exigência óbvia seria ter erro global zero, isto é, os erros positivos e negativos se anularem. Infelizmente, há muitas linhas que irão satisfazer este critério. É necessário critérios adicionais. Um tal critério poderia ser o de escolher a linha que minimiza a soma dos quadrados dos erros. Este critério é chamado de Método dos Mínimos Quadrados e é utilizado para definir o melhor modelo.

A definição matemática do critério de mínimos quadrados é a seguinte. Suponha que o modelo linear é $\hat{y} = b_0+b_1x$, em que $\hat{y}$ é a resposta prevista quando a variável preditora é x. Os parâmetros b_0 e b_1 são os parâmetros da regressão, determinados a partir dos dados. Dado n pares de observação $\{(x_1,y_1), \ldots, (x_n,y_n)\}$, a resposta estimada $\hat{y}_i$ da i-ésima observação é $\hat{y}_i = b_0+b_1x_i$. O erro é $e_i = y_i - \hat{y}_i$.

O melhor modelo linear é dada pelos valores dos parâmetros de regressão que minimizem a soma dos erros ao quadrado (SSE, *Sum of Squared Residuals*):

$$SSE=\sum_{i=1}^{n} e_i{}^2=\sum_{i=1}^{n} (y_i-b_0-b_1x_i)^2 \qquad 4.3$$

sujeito a restrição de erro médio nulo, ou seja, $\sum_{i=1}^{n} e_i=\sum_{i=1}^{n} (y_i-b_0-b_1x_i)=0$.

Pode-se demonstrar que este problema de minimização restrita é equivalente a minimizar a variância dos erros.

4.4.1. Estimação dos Parâmetros do Modelo

Os parâmetros da regressão que dão variância do erro mínima são:

$$b_1 = \frac{\sum_{i=1}^{n} x_i y_i - n x_m y_m}{\sum_{i=1}^{n} x_i^2 - n x_m{}^2} \qquad 4.4$$

$$b_0 = y_m - b_1 x_m \qquad 4.5$$

onde $x_m = \frac{1}{n}\sum_{i=1}^{n} x_i$, média dos valores das variáveis de previsão e $y_m = \frac{1}{n}\sum_{i=1}^{n} y_i$, média das respostas.

Exemplo 4.2 - O número de E/S de disco e os tempos de processamento de sete programas foram medidos como {(14, 2), (16, 5), (27, 7 (42, 9), (39, 10), (50, 13), (83, 20)}.

Um modelo linear para prever o tempo de CPU, como uma função do E/S de disco pode ser desenvolvido como se segue.

Dado n = 7, $\sum x_i y_i$ = 3375, $\sum x_i$ = 271, $\sum x_i^2$ = 13855, $\sum y_i$ = 66, $\sum y_i^2$ = 828, x_m = 38,71 e y_m = 9,43.

Portanto, $b_1 = \sum x_i y_i - n x_m y_m)/(\sum x_i^2 - n x_m^2) = (3375-7\times38.71\times9.43)/(13855-7\times38.71^2) = 0.243$

$b_0 = y_m - b_1 x_m = 9.43-0.2438\times38.71 = -0.008$

O modelo linear desejado é Tempo de CPU = -0,0083 + 0,2438 (número de E/S de disco).

Um gráfico de dispersão dos dados é mostrado na Figura 4.4. Uma linha reta com interceção com 0Y em -0,0083 e declividade 0,2438 é também mostrada nesta figura. Note-se que a linha passa próximo dos pontos dos valores observados.

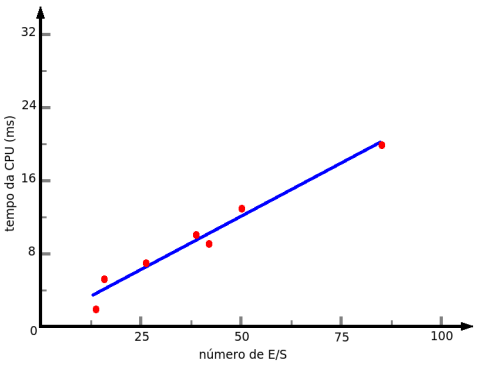


Figura 4.4 - Dispersão de dados de tempo de E/S de disco e CPU.

Tabela 4.3 - Cálculo do erros dos dados de tempo de E/S de Disco e CPU

x_i	y_i	$\hat{y}_i$	e_i	e_i^2
14	2	3,4043	-1,4043	1,9721
16	5	3,8918	1,1082	1,2281
27	7	6,5731	0,4269	0,1822
42	9	10,2295	-1,2295	1,5116
39	10	9,4982	0,5018	0,2518
50	13	12,1795	0,8205	0,6732
83	20	20,2235	-0,2235	0,0500
∑=271	66	66,0000	0,0000	5,8690

Na Tabela 4.3 estão listados o tempo de CPU previsto pelo modelo, os valores medidos, erros e erros quadrados para cada um das sete observações. O SSE é 5,869. Este é o SSE mínimo possível. Quaisquer outros valores de b_0 e b_1 daria um maior SSE.

4.4.2. Análise de Variância

O particionamento da variação em parte explicada e não explicada é útil na prática, já que pode ser facilmente apresentados pelo analista para os tomadores de decisão. Por exemplo, é mais fácil para eles entender que uma regressão que "explica apenas 709/6 da variação" não é tão como a que "explica 90% da variação". A próxima pergunta é quão boa é a variação explicada? A resposta a esta questão estatística é obtida pela chamada Análise de Variância (ANOVA). Esta análise essencialmente testa a hipótese de que o SSR (Soma de Quadrado da Regressão) é inferior ou igual à SSE.

4.4.2.1. Soma de Quadrados

A decomposição da soma de quadrados e nos graus de liberdade associados a variável resposta y, isto é, o desvio de uma observação em relação à média pode ser decomposto como o desvio da observação em relação ao valor ajustado pela regressão mais o desvio do valor ajustado em relação à média, isto é, podemos escrever (y_i-y_m) como:

$$(y_i-y_m)=(y_i-y_m+\hat{y}_i-\hat{y}_i)=(\hat{y}_i-y_m)+(y_i-\hat{y}_i) \quad 4.6$$

Elevando cada componente desta equação ao quadrado e somando para todo o conjunto de observações, e considerando que $\sum_{i=1}^{n} e_i=\sum_{i=1}^{n}(y_i-\hat{y}_i)=0$, obtem-se:

$$\sum_{i=1}^{n}(y_i-y_m)^2=\sum_{i=1}^{n}(\hat{y}_i-y_m)^2+\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \quad (4.7)$$

em que:

$SQT=\sum_{i=1}^{n}(y_i-y_m)^2$ - Soma de Quadrado Total

$SQR=\sum_{i=1}^{n}(\hat{y}_i-y_m)^2$ - Soma de Quadrado da Regressão

$SQE=\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$ - Soma de Quadrado dos Erros (dos Resíduos)

Ou seja, $SQT=SQR+SQE$, é a decomposição a Soma de Quadrados Total em Soma de Quadrados da Regressão e Soma de Quadrados dos Erros.

4.4.2.2. Graus de Liberdade

Cada uma das somas dos quadrados tem um grau de liberdade (gl) que corresponde ao número de dados necessários para calculá-lás.

A SQT tem n-1 graus de liberdade uma vez que o parâmetro y_m deve ser calculado a partir dos dados antes do cálculo da SQT.

A SQR, tem 1 grau de liberdade, uma vez que para sua abtenção são utilizados a diferença entre $\hat{y}_i$ e y_m, o primeiro requer dois parâmetros da regressão (b_0 e b_1) e, o segundo, um parâmetro.

A SQE tem apenas n-2 graus de liberdade uma vez que para sua abtenção são calculados dois parâmetros da regressão (b_0 e b_1) a partir dos dados antes do seu cálculo.

Assim, as somas e seus graus de liberdade associados são os seguintes:

$$SQT=SQR+SQE \quad (4.8)$$
$$n-1=1+n-2 \quad (4.9)$$

Note-se que os graus de liberdade se adicionam de forma semelhante à da soma dos quadrados. Este fato pode ser utilizado para verificar se os graus de liberdade foram atribuídos corretamente.

4.4.2.3. Quadrado Médio

A divisão da soma de quadrados pelos respectivos graus de liberdade é o quadrado médio.

Quadrado Médio Total - $QMT = \frac{SQT}{n-1}$

Quadrado Médio da Regressão - $QMR = \frac{SQR}{1}$

Soma de Quadrado dos Erros - $QME = \frac{SQE}{n-2}$

A relação da decomposição da variabilidade não existe mais nesse caso.

Tabela 4.4 - Tabela ANOVA para a Regressão Linear Simples

Fonte	gl	SQ	QM
Regressão (R)	1	$SQR = \sum_{i=1}^{n} (\hat{y}_i - y_m)^2$	QMR = SQR
Resíduo (E)	n-2	$SQE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$	$QME = \frac{SQE}{n-2}$
Total (T)	n-1		

Assumindo que os erros são independentes e normalmente distribuídos e que todos eles são identicamente distribuídos (com a mesma média e variância), segue-se que y são também normalmente distribuídos desde que x são estocásticos. A soma dos quadrados de variáveis normais tem uma distribuição χ^2. Assim, SQT, SQR e SQE tem distribuição χ^2 com graus de liberdades acima estabelecidos.

Dadas as somas de quadrados SQR e SQE com graus de liberdade gl_R e gl_E, respectivamente, a razão $(SQR/gl_R)/(SQE/gl_E)$ tem uma distribuição F com gl_R graus de liberdade do numerador e gl_E graus de liberdade do denominador, isto decorre da definição da distribuição F.

A hipótese de que a soma SQR é inferior ou igual a SQE é rejeitada ao nível de significância de α se a razão é maior do que o (1-α)-quantil da distribuição F. Assim, a relação computada é comparada com $F[1-\alpha, gl_R, gl_E]$ obtida a partir da tabela de quantis F, as somas dos quadrados são considerados significativamente diferentes se a razão calculada é maior do que a tabelada. Este procedimento é também conhecido como teste F.

O teste F pode ser utilizado para verificar se SSR é significativamente mais elevado do que SSE pelo cálculo da razão entre $(SSR/gl_R) / (SSE/gl_E)$, onde gl_R e gl_E são graus de liberdade para o SSR e SSE, respectivamente. A

quantidade SSR/gl_R é chamado de quadrado médio da regressão (MSR). Em geral, qualquer soma de quadrados, divididos pelos seus graus de liberdade dá o quadrado correspondente média.

Assim razão MSR/MSE tem distribuição F[1,n-2], isto é, uma distribuição F com 1 grau de liberdade no numerador e n-2 graus de liberdade no denominador. Se a razão calculada é maior do que o valor tabelado, as variáveis de previsão explicar uma fração significativa da variação da resposta.

Um arranjo tabular é conveniente para conduzir o teste F é mostrado na Tabela 4.3. A tabela é disposta de modo que o cálculo pode ser feito coluna por coluna a partir da esquerda. Tal como indicado no quadro, o desvio padrão do erro pode ser estimada tomando a raiz quadrada de MSE, o que é uma estimativa da variância de erro.

Deve salientar-se que o teste F é equivalente a testar a hipótese nula de que y não depende de qualquer x_i, isto é, contra uma hipótese alternativa de que y depende de pelo menos um x_i, e, portanto, b_1 é nulo. Se a razao calculada for menor do que o valor tabelado, a hipótese nula não pode ser rejeitada ao nível de significância indicado.

Em modelos de regressão simples, existe apenas uma variável de previsão e, consequentemente, o teste F reduz-se ao teste de $b_1 = 0$. Assim, se o intervalo de confiança de b_1 não inclui zero, o parâmetro é diferente de zero, a regressão explica uma parte significativa da variação de resposta e o teste F não é necessário.

Exemplo 4.3 - Para os dados no disco da memória de CPU do Exemplo 4.1, a análise de variância é mostrada na Tabela 4.4. A partir da tabela, vemos que a razão F calculado for maior do que a obtida a partir da tabela, e assim faz a regressão explicar uma parte significativa da variação.

Tabela 4.5 - Cálculo do erros dos dados de tempo de I/O de Disco e CPU

x_i	y_i	$\hat{y}_i$	$(\hat{y}_i - y_m)^2$	$(y_i - \hat{y}_i)^2$
14	2	3,4043	-1,4043	1,9721
16	5	3,8918	1,1082	1,2281
27	7	6,5731	0,4269	0,1822
42	9	10,2295	-1,2295	1,5116
39	10	9,4982	0,5018	0,2518
50	13	12,1795	0,8205	0,6732
83	20	20,2235	-0,2235	0,0500
∑=271	66	66,0000	0,0000	5,8690

n = 7

$y_m = 66/7 = 9,4$

Tabela 4.6 - Tabela ANOVA para a Regressão Linear Simples

Fonte	gl	SQ	QM	F	$F_{tabelado}$
Regressão (R)	1	SQR = 379,22	QMR = 379,22	F = 379,22/1,14 = 332,65	F[0,95;1;5] = 5,99
Resíduo (E)	5	SQE = 5,7	QME = 5,7/5 = 1,14		
Total (T)	6				

Memória de cálculo:

i	x_i	y_i	$\hat{y}_i$	$(\hat{y}_i-y_m)^2$	$(y_i-\hat{y}_i)^2$
1	14	2	3,4	$(3,4-9,4)^2=36,0$	$(3,4-2)^2=2,0$
2	16	5	3,9	$(3,9-9,4)^2=30,3$	$(3,9-5)^2=1,2$
3	27	7	6,6	$(6,6-9,4)^2=7,8$	$(6,6-7)^2=0,2$
4	42	9	10,2	$(10,2-9,4)^2=0,6$	$(10,2-9)^2=1,4$
5	39	10	9,5	$(9,5-9,4)^2=0,0$	$(9,5-10)^2=0,3$
6	50	13	12,2	$(12,2-9,4)^2=7,8$	$(12,2-13)^2=0,6$
7	83	20	20,2	$(20,2-9,4)^2=116,6$	$(20,2-20)^2=0,0$
∑	271	66	66	199,22	5,7

$n = 7$ $y_m = 66/7 = 9,4$

Note-se que no Exemplo 4.3 a regressão passou no teste F, indicando que a hipótese de todos os parâmetros serem zero não pode ser aceita.

4.5. Projeto e Análise de Experimentos

O desempenho muitas vezes depende mais de um fator, como a vazão e a carga de trabalho. A análise adequada requer que os efeitos de cada fator sejam isolados uns dos outros de modo que declarações significativas possam ser feitas sobre os diferentes níveis do fator, por exemplo, vazões diferentes. Tal análise é o tema principal desta seção. As técnicas apresentadas nesta seção irá permitir:

1. Criar um bom conjunto de experimentos para a medição ou simulação
2. Desenvolver um modelo que melhor descreva os dados obtidos
3. Estimar a contribuição de cada alternativa (por exemplo, cada processador e cada carga de trabalho) para o desempenho
4. Isolar os erros de medição
5. Estimar intervalos de confiança para os parâmetros do modelo
6. Verificar se as alternativas são significativamente diferentes
7. Verificar se o modelo é adequado

O objetivo de um projeto experimental adequado é obter o máximo de informação com o número mínimo de experimentos. Isso economiza trabalho considerável que teria sido gasto na coleta de dados. A análise adequada de experimentos também ajuda a separar os efeitos de vários fatores que podem afetar o desempenho. Além disso, permite determinar

se um fator tem um efeito significativo ou se a diferença observada é simplesmente devido a variações aleatórias causadas por erros de medição e parâmetros que não foram controlados. Vários termos novos que são usados no planejamento experimental e análise são explicados primeiramente.

4.5.1. Terminologia

Os termos que são usados em desenho experimental e análise são explicados usando o exemplo de um desenho para estudar uma estação de trabalho pessoal

O problema é projetar uma estação de trabalho pessoal, onde várias opções têm que ser feitas. Primeiro, um microprocessador tem de ser escolhido para a CPU, as alternativas do microprocessador são Intel, AMD e Apple. Segundo, o tamanho de memória tem que ser escolhido dentre 1 GB, 2 GB e 4 GB. Terceiro, a estação de trabalho pode ter uma, duas, três ou quatro unidades de disco. Em quarto lugar, a carga de trabalho nas estações de trabalho pode ser de um dos três tipos: secretariado, gestão ou científico. O desempenho também depende das características do usuário, sendo os usuários de uma escola, faculdade ou pós-graduação. A Tabela 4.7 discute os termos frequentemente utilizados no projeto e análise de experimento.

Tabela 4.7 - Termos utilizados em projeto e análise de experimento

Variável Resposta	o resultado de um experimento é chamado variável resposta. Geralmente a variável resposta é o desempenho medido do sistema. Por exemplo, no estudo de projeto da estação de trabalho a variável resposta poderia ser o rendimento expresso em tarefas concluídas por unidade de tempo, ou tempo de resposta para as tarefas ou qualquer outra métrica. Uma vez que as técnicas de desenho experimental são aplicáveis para qualquer tipo de medidas, não apenas medições de desempenho, o termo resposta é usado no lugar de performance por ser mais geral.
Fatores	cada variável que afeta a variável resposta e que tem várias alternativas é chamado de fator. Por exemplo, existem cinco fatores no estudo do projeto de estação de trabalho. Os fatores são tipo de CPU, tamanho da memória, número de unidades de disco, carga de trabalho utilizado e nível de escolaridade do usuário. Os fatores são também chamados de variáveis preditoras ou preditores.
Níveis	os valores que um fator pode assumir são chamados os seus níveis. Em outras palavras, cada nível de um fator constitui uma alternativa para esse fator. Por exemplo, no estudo de projeto de estação de trabalho o tipo CPU tem três níveis: Intel, AMD e Apple. O Tamanho da memória tem três níveis: 1 GB, 2 GB ou 4 GB. O número de unidades de disco tem quatro níveis: 1, 2, 3 ou 4. A carga de trabalho possui três níveis: secretariado, gestão ou científico. Finalmente, os usuários podem ser colocados em um dos três níveis de ensino dentre diplomados numa escola, graduados e pós-graduados. O termo

	tratamento também é usado na literatura de delineamento experimental no lugar de níveis.
Fatores Primários	os fatores cujos efeitos devem ser quantificados são chamados fatores primários. Por exemplo, no estudo de projeto de estação de trabalho, pode ser principalmente interessados em quantificar apenas o efeito do tipo de CPU, tamanho da memória e número de unidades de disco. Assim, existem três fatores principais neste caso.
Fatores Secundários	os fatores que causam impacto no desempenho, mas cujo impacto não estamos interessados em quantificar são chamados de fatores secundários. Por exemplo, no estudo da estação de trabalho podem não estar interessados em determinar se o desempenho com pós-graduação é melhor do que com os graduados da faculdade. Da mesma forma, não queremos quantificar a diferença entre as três cargas de trabalho. Estes são os fatores secundários.
Replicação	repetição de todos ou alguns experimentos é chamado de replicação. Por exemplo, se todos os experimentos em um estudo são repetidos três vezes, o estudo diz-se de três repetições.
Desenho	o desenho experimental consiste em especificar o número de experimentos, as combinações de níveis do fator para cada experimento e o número de repetições de cada experimento. Por exemplo, no estudo do projeto de estação de trabalho, pode-se realizar experimentos correspondentes a todas as combinações possíveis de níveis de cinco fatores. Isso exigiria 3×3×4×3×3 ou 324 experimentos. Poderíamos repetir cada experimento cinco vezes, levando a um total de 1215 observações. Este é um projeto experimental possível. Mais tarde, outros possíveis projetos experimentais serão descritos.
Unidade Experimental	qualquer entidade que é usada para o experimento é chamada uma unidade experimental. Geralmente apenas as unidades experimentais que considerão um dos fatores em estudo são de interesse. Por exemplo, no estudo de projeto da estação de trabalho, os usuários contratados para usar a estação de trabalho, enquanto as medições estão sendo realizadas, podem ser considerados a unidade experimental. Não é de interesse unidades experimentais, apesar de afetar a resposta, um dos objetivos do projeto experimental é o de minimizar o impacto da variação entre as unidades experimentais.
Interação	dois fatores A e B são ditos interagirem se o efeito de um depende do nível do outro. Por exemplo, Tabela 4.5 mostra o desempenho de um sistema com dois fatores. Como o fator A é alterado do nível A_1 ao nível A_2, o desempenho aumenta em 2, independentemente do nível de fator de B. Neste caso não há interação. A Tabela 4.6 mostra outra possibilidade. Neste caso, como o fator A é alterado do nível A_1 ao nível A_2, o desempenho aumenta, quer por dois ou por três, dependendo se B está no nível B_1 ou nível B_2, respectivamente. Os dois fatores interagem neste caso. A apresentação gráfica deste exemplo é dada na Figura 4.4; no caso a, as linhas são paralelas, indicando que não há interação. No caso b, as linhas não são paralelas, indicando interação.

Tabela 4.8 - Fatores não interagem

	A1	A2

B1	3	5
B2	6	8

Tabela 4.9 - Fatores que interagem

	A1	A2
B1	3	5
B2	6	9

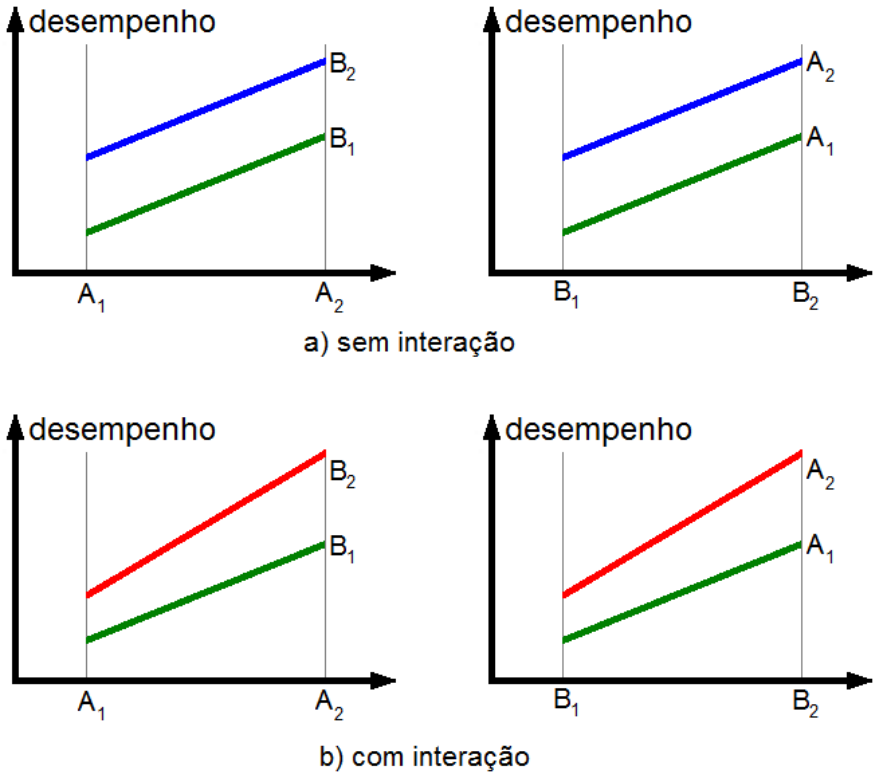


Figura 4.5 - Apresentação gráfica de fatores que a) interagem e b) não interagem.

4.5.2. Erros Comuns em Experimentação

Os analistas inexperientes, que não estão cientes do desenho experimental e técnicas de análise, muitas vezes obtém conclusões equivocadas devido aos seguintes erros

Tabela 4.10 - Erros comuns em experimentação

A variação devido ao erro experimental é ignorada
Cada valor medido é um valor aleatório. Cada vez que ele é repetido, o valor medido pode ser ligeiramente diferente, mesmo se todos os fatores controláveis são mantidos no mesmo valor. Na tomada de decisões com base em medições, é importante isolar o efeito de erros. A variação devido a um fator deve ser comparada com aquela devido a erros antes de tomar uma decisão sobre o seu efeito. Analistas inexperientes que não estão cientes disto, atribuem toda variação aos fatores e ignora completamente os erros
A variação devido ao erro experimental é ignorada
cada valor medido é um valor aleatório. Cada vez que ele é repetido, o valor medido pode ser ligeiramente diferente, mesmo se todos os fatores controláveis são mantidos no mesmo valor. Na tomada de decisões com base em medições, é importante isolar o efeito de erros. A variação devido a um fator deve ser comparada com aquela devido a erros antes de tomar uma decisão sobre o seu efeito. Analistas inexperientes que não estão cientes disto, atribuem toda variação aos fatores e ignora completamente os erros
Parâmetros importantes não são controlados
anteriormente, foi salientado que a lista de parâmetros deve incluir todas as cargas de trabalho, ambiente e parâmetros do sistema que afetam o desempenho. Apenas alguns destes parâmetros são selecionados como fatores e tem seus valores variados no experimento. Por exemplo, quando se comparam duas estações de trabalho, o usuário da estação de trabalho tem um efeito significativo sobre o desempenho medido. No entanto, se o efeito dos usuários não é corretamente contabilizado, os resultados podem não ser significativos
Efeitos de diferentes fatores não são isolados
um analista pode variar vários fatores simultaneamente e, em seguida, pode não ser capaz de atribuir a mudança no desempenho a qualquer fator em particular. Para evitar isso, alguns analistas usam desenhos experimentais muito simples que levam a problemas, o que será discutido a seguir
Designs simples variando um fator por vez são usados
tal projeto é um desperdício des recursos. Ele requer muitos experimentos para obter a mesma informação. Com delineamento experimental adequado, é possível obter intervalos de confiança para os efeitos com o mesmo número de experimentos
Interações são ignoradas
muitas vezes o efeito de um fator depende do nível de outros fatores. Por exemplo, o efeito da adição de 1 KB de cache pode depender do tamanho do programa. Tais interações não podem ser estimadas em designs que varia um fator por vez
Muitos experimentos são conduzidos
o número de experimentos é uma função do número de fatores e seus níveis. É melhor dividir o projeto em várias etapas cada um usando um pequeno desenho em vez de usar um projeto enorme, com muitos fatores e níveis. Na primeira etapa, o número de fatores e os níveis devem ser pequenos. Tal projeto vai ajudar a depurar o processo experimental e também ajudar a descobrir os fatores que não são significativas e não devem ser incluídas no projeto. O primeiro projeto também vai dizer se os pressupostos da análise estão satisfeitas e se todas as transformações de dados são obrigatórias. Mais fatores e níveis podem ser adicionados em etapas subsequentes. As técnicas experimentais de projeto e análise apresentadas a seguir ajudam a evitar esses problemas

4.6. Tipos de Projetos Experimentais

Existem inúmeras variedades de desenhos experimentais. Os três modelos mais utilizados são projetos simples, fatoriais completos e fatoriais fracionados. As explicações destes projetos e suas vantangens e desvantagens são dadas a seguir.

4.6.1. Designs Simples

Em um projeto simples, inicia-se com uma configuração típica e segue variando um fator de cada vez para ver como cada fator afeta o desempenho. Por exemplo, no estudo de projeto da estação de trabalho discutido anteriormente, uma configuração típica pode consistir de uma CPU Intel com duas unidades de disco executando uma tarefa de gestão por um graduado da faculdade. O desempenho desta configuração é medido em primeiro lugar. Em seguida, faz variar o primeiro fator, a CPU e, em seguida, o desempenho é comparado com outros processadores na mesma configuração e carga de trabalho. Isso nos ajudará a decidir qual CPU é a melhor. Em seguida, alterar o número de unidades de disco para um, três e quatro, comparando o desempenho de forma a encontrar o número ideal. Dado k fatores, com o i-ésimo fator com n_i níveis, um design simples requer apenas n experimentos, onde $n = 1 + \sum_{i=1}^{k} (n_i - 1)$.

Entretanto, neste projeto não se faz o melhor uso do esforço despendido. Não é estatisticamente eficiente. Além disso, sem os fatores de interação, este projeto pode levar a conclusões erradas. Por exemplo, se o efeito da CPU depende do tamanho da memória, a combinação ideal não pode ser determinada até que todas as possibilidades sejam avaliadas. Este projeto, portanto, não é recomendado.

4.6.2. Projeto Fatorial Completo

Um projeto fatorial completo utiliza todas as combinações possíveis com todos os níveis de todos os fatores. Um estudo de desempenho com k fatores, com o i-ésimo fator com n_i níveis, exige n experimentos, onde $n = \prod_{i=1}^{k} n_i$.

No estudo de projeto de estação de trabalho, o número de experimentos

seria n = (3 CPUs) × (3 níveis de memória) × (4 unidades de disco) × (3 cargas de trabalho) × (3 níveis de ensino) = 324 experimentos.

A vantagem de um design fatorial completo é que todas as combinações possíveis de configuração e carga de trabalho são examinadas. Podemos encontrar o efeito de cada fator, incluindo os fatores secundários e suas interações. O principal problema é o custo do estudo. Levaria muito tempo e recursos financeiros para realizar estes experimentos, especialmente quando se leva em conta a possibilidade de que cada um desses experimentos pode ter que ser repetido várias vezes. Há três maneiras de reduzir o número de experimentos:

1. reduzir o número de níveis de cada fator.
2. reduzir o número de fatores.
3. usar fatoriais fracionados.

A primeira alternativa é especialmente recomendada. Em alguns casos, pode-se tentar apenas dois níveis de cada fator e determinar a importância relativa de cada um deles. Um projeto fatorial completo em que cada um dos k fatores é usado em dois níveis exige experimentos 2^k. Este é um projeto muito popular e é chamado de projeto 2^k. Depois que a lista de fatores ter sido substancialmente reduzida, pode-se tentar mais níveis por fator. A terceira alternativa de planejamento, fatorial fracionado, é descrito na próxima seção.

4.6.3. Design Fatorial Fracionado

Às vezes, o número de experimentos necessários para um design fatorial completo é muito grande. Isso pode acontecer se o número de fatores ou seus níveis é grande. Pode não ser possível a utilização de um design fatorial completo devido à despesa ou o tempo necessário. Nesses casos, pode-se usar apenas uma fração do projeto fatorial completo. O exemplo a seguir ilustra o processo

Exemplo 4.4 - Considere apenas quatro dos cinco fatores em estudo da estação de trabalho. Vamos ignorar o número de unidades de disco para este exemplo. Temos quatro fatores, cada um em três níveis. Portanto, o número de experimentos necessários é n = (3 CPUs) × (3 níveis de memória) × (3 cargas de trabalho) × (3 níveis de ensino) = 81 experimentos. O modelo fatorial completo composto por 81 experimentos é o projeto chamado 3^4. Um planejamento fatorial fracionário 3^{4-2} consistindo de apenas nove experimentos é mostrada na Tabela 4.8. Observe que cada um dos quatro fatores é usado três vezes em cada um dos seus três níveis

Tabela 4.11 - Uma amostra planejamento fatorial fracionário

Experimento	CPU	Nível de Memória	Carga de Trabalho	Nível Educacional
1	Intel	4 GB	Gerencial	Graduação
2	Intel	2 GB	Científico	Pós-Graduação
3	Intel	1 GB	Secretaria	Colégio

4	AMD	4 GB	Científico	Colégio
5	AMD	2 GB	Secretaria	Graduação
6	AMD	1 GB	Gerencial	Pós-Graduação
7	Apple	4 GB	Secretaria	Pós-Graduação
8	Apple	2 GB	Gerencial	Colégio
9	Apple	1 GB	Científico	Graduação

Para todas as vantagens, há uma desvantagem correspondente. Planejamentos fatoriais fracionados economizam tempo e dinheiro quando comparado aos fatoriais completos. No entanto, as informações obtidas a partir de um planejamento fatorial fracionado são menores que o obtido a partir de um design fatorial completo. Por exemplo, pode não ser possível obter interações entre todos os fatores. Por outro lado, se algumas das interações são conhecidas e insignificantes, isso não pode ser considerado um problema e o tempo e os custos de um projeto fatorial completo não pode ser justificada.

4.7. Experimentos com fator único

Projetos de um fator são usados para comparar várias alternativas de uma única variável categórica. Por exemplo, pode-se usar tal concepção para a comparação de vários processadores, vários sistemas de computadores ou vários esquemas de cache. As técnicas para analisar tais projetos são apresentados neste capítulo. Não há limite no número de níveis que o fator pode tomar. Em particular, ao contrário dos desenhos 2^k, o número de níveis pode ser superior a 2.

4.7.1. Modelo

O modelo utilizado em projetos de um fator único é

$$y_{ij}=\mu+\alpha_j+e_{ij} \qquad 4.10$$

Aqui, y_{ij} é a i-ésima reposta (ou observação) com o fator de nível j (isto é, a j-ésima alternativa), μ é a resposta média, α_j é o efeito da j-ésima alternativa, e e_{ij} é o termo de erro. Os efeitos são calculados de modo que eles somam zero $\sum \alpha_j=0$.

4.7.2. Cálculo dos Efeitos

Os dados medidos em um design de um fator consiste em r observações para cada uma das alternativas. Há um total de r×a observações, que são dispostos em um uma matriz r×a de modo que as r observações pertencentes à alternativa j de um vetor coluna. Fazendo y_{ij} denotar a entrada i na coluna j e substituindo as respostas observadas na Equação do modelo, obtem-se r×a equações. Adicionando estas equações, obtem-se

$$\sum_{i=1}^{r}\sum_{j=1}^{a} y_{ij} = r \times a\,\mu + \sum_{j=1}^{a}\alpha_j + \sum_{i=1}^{r}\sum_{j=1}^{a} e_{ij} \qquad 4.11$$

Desde que os efeitos α_j adicionados sejam zero (pelo design) e se quer que o erro médio seja zero, a equação anterior torna-se

$$\sum_{i=1}^{r}\sum_{j=1}^{a} y_{ij} = r \times a\,\mu + 0 + 0 \qquad 4.12$$

O parâmetro μ do modelo, portanto, é dado po

$$\mu = \frac{\sum_{i=1}^{r}\sum_{j=1}^{a} y_{ij}}{r \times a} \qquad 4.13$$

A quantidade do lado direito é a chamada média geral de todas as a×r respostas, é denotada por $\ddot{y}_{..}$. Os dois pontos no índice indicam que a média é feito ao longo de ambas as dimensões (linhas e colunas) da matriz. Isso deve ser distinguido das médias de coluna, que são obtidos por uma média de respostas pertencentes a uma determinada coluna (ou linha). A média da coluna quer dizer para a coluna j-ésima é denotado por $\ddot{y}_{.j}$ e é calculado da seguinte forma $\ddot{y}_{.j} = \frac{\sum_{i=1}^{r} y_{ij}}{r}$, substituindo μ + α_j + e_{ij} por y_{ij}, obtemos $\ddot{y}_{.j} = \frac{\sum_{i=1}^{r}(\mu+\alpha_j+e_{ij})}{r} = \frac{r\mu + r\alpha_j + \sum_{i=1}^{r} e_{ij}}{r} = \mu + \alpha_j$.

Aqui assumimos que os termos de erro para observações r pertencente a cada alternativa somam zero. O parâmetro α_j pode assim ser estimado como segue $\alpha_j = \ddot{y}_{.j} - \mu = \ddot{y}_{.j} - \ddot{y}$.

Exemplo 4.5 - Em um estudo de comparação de código, o número de bytes necessários para codificar uma carga de trabalho em três diferentes processadores R, V e Z foi medida cinco vezes cada um (uma vez que cada um dos diferentes programadores pediram para o código de mesma carga de trabalho). Os dados medidos estão mostrados na Tabela 4.9.

Tabela 4.12 - Análise do Estudo Comparativo do Tamanho do Código

R (b)	V (b)	Z (b)
144	101	130
120	144	180
176	211	141
288	288	374
144	72	302

Ao analisar esses dados, as cinco observações foram assumidas para a mesma carga de trabalho e as entradas em uma única linha são consideradas independentes. Se as entradas estiverem relacionadas, uma análise de dois fatores teria que ser usada. A análise de um fator, apresentada aqui, só é válida se as linhas não representam qualquer fator adicional. A análise é mostrada na Tabela 4.10. Aqui, um número de níveis é 3, e o número de repetições é 5. Nós somamos cada coluna para encontrar a soma da coluna e em seguida, adicione somas das coluna para obter a soma total. As somas são divididas pelo respectivo número de observações para obter as médias. As diferenças entre as médias de coluna e a média geral dão os efeitos da coluna

Os resultados são interpretados da seguinte forma. Um processador médio requer 187,7 bytes de armazenamento. Os efeitos dos processadores R, V e Z são -13,3, -24,5 e 37,7, respectivamente. Isto é, R requer 13,3 b a menos que a média dos processadores, V requer 24,5 b menos que a média dos processadores, e Z requer 37,7 b a mais que a média dos processadores

Tabela 4.13 - Dados de um Estudo Comparativo de Tamanho Código

	R	V	Z	
	144	101	130	
	120	144	180	
	176	211	141	
	288	288	374	
	144	72	302	
Soma da Coluna	$\sum y_{.1} = 872$	$\sum y_{.2} = 816$	$\sum y_{.3} = 1127$	$\sum y_{..} = 2815$
Mádia da Coluna	$\ddot{y}_{.1} = 174,4$	$\ddot{y}_{.2} = 163,2$	$\ddot{y}_{.3} = 225,4$	$\mu = \ddot{y}_{..} = 187,7$
Efeito da Coluna	$\alpha_1 = \ddot{y}_{.1} - \ddot{y}_{..} = -13,3$	$\alpha_2 = \ddot{y}_{.2} - \ddot{y}_{..} = -24,5$	$\alpha_3 = \ddot{y}_{.1} - \ddot{y}_{..} = 37,7$	

4.7.3. Estimativa dos Erros Experimentais

Uma vez que os parâmetros do modelo foram computados, podemos estimar a resposta para cada uma das alternativas $\hat{y}_j = \mu + \alpha$.

A diferença entre as respostas estimada e medida representa erro experimental. Se computarmos os erros experimentais em cada uma das observações ar, o erro médio deve ser zero pois os parâmetros μ e α_j foram calculados assumindo que a soma dos erros para cada coluna é zero. A variância dos erros pode ser estimada a partir da soma dos erros ao quadrado (SSE)

$$SSE=\sum_{i=1}^{r}\sum_{j=1}^{a} e_{ij}^2 \qquad 4.14$$

Exemplo 4.6 - Computação de erros para o estudo de comparação Código tamanho do Exemplo 4.5 é a seguinte:

$$\begin{bmatrix} 144 & 101 & 130 \\ 120 & 144 & 180 \\ 176 & 211 & 141 \\ 288 & 288 & 374 \\ 144 & 72 & 302 \end{bmatrix} = \begin{bmatrix} 187{,}7 & 187{,}7 & 187{,}7 \\ 187{,}7 & 187{,}7 & 187{,}7 \\ 187{,}7 & 187{,}7 & 187{,}7 \\ 187{,}7 & 187{,}7 & 187{,}7 \\ 187{,}7 & 187{,}7 & 187{,}7 \end{bmatrix} + \begin{bmatrix} -13{,}3 & -24{,}5 & 37{,}7 \\ -13{,}3 & -24{,}5 & 37{,}7 \\ -13{,}3 & -24{,}5 & 37{,}7 \\ -13{,}3 & -24{,}5 & 37{,}7 \\ -13{,}3 & -24{,}5 & 37{,}7 \end{bmatrix} + \begin{bmatrix} -30{,}4 & -62{,}2 & -95{,}4 \\ -54{,}4 & -19{,}2 & -45{,}4 \\ 1{,}6 & 47{,}8 & -84{,}4 \\ 113{,}6 & 125{,}8 & 148{,}6 \\ -30{,}4 & -91{,}2 & 76{,}6 \end{bmatrix} \qquad 4.15$$

Cada observação foi dividida em três partes: a μ média geral, o efeito do processador α_j, e os resíduos. A notação matricial é usada para todas as três partes. A soma dos quadrados de entradas na matriz residual é SSE = $(-30{,}4)^2 + (-54{,}4)^2 + \ldots + (76{,}6)^2 = 94365{,}20$

4.7.4. Atribuição da Variação

A variação total de y em um design experimental de um fator pode ser atribuída aos fatores e aos erros. Para isso, eleva-se ao quadrado ambos os lados da equação do modelo

$$y_{ij}^2=\mu^2+\alpha_j^2+e_{ij}^2+2\mu\,\alpha_j+2\mu e_{ij}+2\alpha_j e_{ij} \qquad 4.16$$

Adicionando os ar termos correspondentes das equações, obtem-se

$$\sum_{i=1}^{r}\sum_{j=1}^{a} y_{ij}^2=\sum_{i=1}^{r}\sum_{j=1}^{a}\mu^2+\sum_{i=1}^{r}\sum_{j=1}^{a}\alpha_j^2+\sum_{i=1}^{r}\sum_{j=1}^{a} e_{ij}^2+termos\,de\,produtos\,cruzados \qquad 4.17$$

Os termos de produto cruzado de todos os complementos é zero pois $\sum \alpha_j=0$ e $\sum\sum e_{ij}=0$.

A equação anterior, expressa em termos de somas de quadrados, pode ser escrita como

$$SSY = SS0 + SSA + SSE \quad (4.18)$$

onde SSY é a soma dos quadrados de y, SS0 é a soma dos quadrados das médias, SSA é a soma dos quadrados dos efeitos, e SSE é a soma de erros quadrados. Note-se que SS0 e SSA podem ser facilmente calculadas da seguinte forma

$$SS0 = \sum_{i=1}^{r}\sum_{j=1}^{a} \mu^2 = r \times a\mu^2 \quad (4.19)$$

$$SSA = \sum_{i=1}^{r}\sum_{j=1}^{a} \alpha_j^2 = r\sum_{j=1}^{a} \alpha_j^2 \quad (4.20)$$

Assim, SSE pode ser calculada facilmente a partir SSY sem calcular os erros individuais. A variação total de y (SST) é definida com

$$SST = \sum_{i=1}^{r}\sum_{j=1}^{a} (y_{ij} - \ddot{y})^2 = \sum_{i=1}^{r}\sum_{j=1}^{a} y_{ij}^2 - r \times a\,\ddot{y}^2 = SSY - SS0 = SSA + SSE \quad (4.21)$$

A variação total pode ser dividida em duas partes, SSA e SSE, que representam o que é explicado e as partes não explicadas da variação. Eles podem ser expressos em percentagem da variação total. A alta porcentagem de variação explicada indica um bom modelo

Exemplo 4.7 - Para o estudo de comparação Código Tamanho do Exemplo 4.5

SSY = 1.442 + 1.202 + ... + 3.022 = 633.63

SS0 = arμ^2 = 3×5×(187,7)^2 = 528.281,

SSA = r$\sum\alpha_j^2$ = 5[(-13,3)^2 + (-24,5)^2 + (37,6)^2] = 10.992,

SST = SSY - SS0 = 633.639,0 - 528.281,7 = 105.357,

SSE = SST - SSA = 105.357,3 – 10.992,1 = 943.65.

Porcentagem de variação explicada pelos processadores = 100 ×10.992,13/105.357,3 = 10.4%.

Os restantes 89,6% da variação no tamanho do código é devido a erros experimentais, que neste caso poderia ser atribuída a diferenças de programador. A questão de saber se 10,4% - a contribuição processadores de variação - é estatisticamente significativa é abordada na próxima seção.

4.7.5. Análise da Variação

Na alocação da variação a diferentes fatores é uma abordagem informal muito útil na prática. Nesse enfoque, qualquer fator que explicou um alto percentual de variação foi considerado importante. Essa importância deve ser diferenciada de significado, que é um termo estatístico. Para determinar se um fator tem um efeito significativo sobre a resposta, os estatísticos comparam a sua contribuição para a variação dos erros. Se a variação inexplicada (devido a erros) é alta, um fator que explica uma grande fração da variação pode vir a ser estatisticamente insignificante. O procedimento estatístico para analisar o significado de vários fatores é chamado de Análise de Variância (ANOVA). O procedimento para experimentos com um fator é muito semelhante ao que foi explicado anteriormente. Para entender a ANOVA, considere a soma dos quadrados - SSY, SS0, SSA e SSE. Cada uma das somas dos quadrados tem um grau de liberdade associado que corresponde ao número de valores independentes necessários para computá-las. Os graus de liberdade para as somas são as seguintes

$$SSY = SS0 + SSA + SSE \quad (4.22)$$

$$r \times a = 1 + (a-1) + a(r-1) \quad (4.23)$$

A soma SSY consiste em uma soma de termos a x r, todos os quais podem ser escolhidos independentemente. Este, portanto, tem a x r graus de liberdade. A soma SS0 consiste em um único termo μ^2 que é repetida a x r vezes. Uma vez um valor para μ foi escolhido, SS0 pode ser computada. Assim, SS0 tem um grau de liberdade

A soma SSA contém uma soma de um termo α_j^2 mas apenas a-1 delas são independente a soma de α_j é zero. Portanto, SSA tem um a-1 graus de liberdade.

A soma SSE consiste em a x r termos de erro, dos quais apenas a(r-1) podem ser independentemente escolhidos. Isto é porque os r erros correspondente a r repetições de cada experimento devem somar zero. Assim, somente r-1 erros em cada um dos experimentos são independentes. Observe que os graus de liberdade dos dois lados das equações anteriores também se somam. Isso verifica se os graus de liberdade foram corretamente atribuídos.

O Teste F pode ser usado para verificar se SSA é significativamente maior que SSE. Assumindo que os erros são normalmente distribuídos, SSE e SSA têm distribuições Qui-quadrado. A relação $(SSA/gl_A)/(SSE/gl_e)$, onde gl_A = a-1 e gl_e = a(r-1) são graus de liberdade (gl) para SSA e SSE, respectivamente, tem uma distribuição F com numerador gl_A e gl_e graus de liberdade no denominador. Se a proporção calculada é maior do que o quantil $F_{[1-\alpha,\ glA,\ gle]}$ obtido a partir da tabela de quantis das variáveis F, SSA é considerado significativamente maior que SSE. A quantidade SSA/gl_A é chamado de quadrado médio de A (MSA). De forma similar,

SSE/gl_e é chamado de quadrado médio dos erros (MSE). Se a relação calculado MSA/MSE é maior do que o valor lido da tabela de quantis da variável F, o fator é assumido para explicar uma fração significativa da variação. Um arranjo tabular conveniente para realizar o teste F é mostrado na Tabela 4.14.

Tabela 4.14 - Tabela ANOVA para experimentos de um fator

Componente	Suma de Quadrados	Percentual de Variação	Graus de Liberdade	Quadrado da Média	F Calculado	F Tabelado
y	$\sum\sum y_{ij}^2$		ar			
$\ddot{y}_{..}$	SS0 = arμ2		1			
$y-\ddot{y}_{.}$	SST=SSY-SS0	100	ar-1			
A	SSA = $r\sum\alpha_j^2$	100(SSA/SST)	a-1	MSA = SSA/(a-1)	MSA/MSE	F[1-α,a-1,a(r-1)]
e	SSE=SST-SSA	100(SSE/SST)	a(r-1)	MSE = SSE/[a(r-1)]		

$s_e = MSE^{1/2}$

Exemplo 4.8 - O ANOVA para o estudo de comparação Código Tamanho do Exemplo 4.5 é mostrada na Tabela 4.15. Da tabela, vemos que o valor de F calculado é menor que o da tabela e, portanto, mais uma vez concluímos que a diferença observada nos tamanhos dos códigos é principalmente devido a erros experimentais e não a qualquer diferença significativa entre os processadores

Tabela 4.15 - Tabela ANOVA para o Estudo de Comparação Código Tamanho

Componente	Soma de Quadrados	Percentual de Variação	Graus de Liberdade	Quadrado da Média	F-Calculado	F-Tabelado
y	633.639,00					
$\ddot{y}_{..}$	528.281,69					
$y-\ddot{y}_{..}$	105.357,31	100,0	14			
A	10992.13	10,4	2	5.496,1	0,7	2,8
e	94.365,20	89,6	12	7.863,8		

$s_e = MSE^{1/2} = 7.863,77^{1/2} = 88,68$

4.8. Experimentos com 2^k fatores

Um projeto experimental 2^k fatores é utilizado para determinar o efeito de k fatores, cada um dos quais tem duas alternativas ou níveis. Esta classe de desenho fatorial merece discussão especial pois é fácil de analisar e ajuda a ordenar os fatores em ordem de impacto. No início de um estudo de desempenho, o número de fatores e seus níveis geralmente são grandes. Um projeto fatorial completo com um número tão grande de fatores e níveis não pode ser a melhor utilização dos esforços disponíveis. O primeiro passo deve ser o de reduzir o número de fatores e escolher aqueles fatores que têm impacto significativo sobre o desempenho.

Muitas vezes, o efeito de um fator é unidirecional, ou seja, o desempenho de forma contínua diminui ou aumenta continuamente quando o fator varia do mínimo ao máximo. Por exemplo, o desempenho deverá melhorar quando o tamanho da memória é aumentado ou quando o número de unidades de disco é aumentado. Em tais casos, podemos começar a experimentar, o nível mínimo e o máximo do fator. Isso nos ajudará a decidir se a diferença no desempenho é significativa o suficiente para justificar uma análise pormenorizada.

A fim de explicar os conceitos de designs 2^k, é útil começar com um caso simples de apenas dois fatores (k = 2). Este caso especial é apresentado nas seções seguintes. Após o desenvolvimento deste caso, serão generalizados os conceitos para um número maior de fatores.

4.9. 2^2 Fatoriais

Um projeto experimental 2^2 é um caso especial de um projeto 2^k fatorial com k = 2. Neste caso, há dois fatores e dois níveis cada. Tal projeto pode ser facilmente analisado utilizando um modelo de regressão, conforme mostrado pelo seguinte exemplo.

Exemplo 4.9 - Considere o problema de estudar o impacto do tamanho da memória e tamanho do cache no desempenho de uma estação de trabalho sendo projetada. Dois níveis de cada um desses dois fatores são escolhidos para a simulação inicial. O desempenho da estação de trabalho em milhões de instruções por segundo (MIPS) é listada na Tabela 4.16.

Tabela 4.16 - Desempenho em MIPS

Tamanho do Cache (kB)	Tamanho da Memória 4 GB	Tamanho da Memória 16 GB
1	15	45
2	25	75

Seja definir duas variáveis x_A e x_B da seguinte forma:

x_A = -1, se 4 GB de memóri

x_A = +1, se 16 GB de memóri

x_B = -1, se 1 KB de cache

x_B = +1, se 2 KB de cache

O desempenho em MIPS y agora pode ser estimado a partir de x_A e x_B usando um modelo de regressão não-linear da forma:

$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B$

Substituindo os quatro observações no modelo, temos as seguintes quatro equações:

$15 = q_0 - q_A - q_B + q_A$

$45 = q_0 + q_A - q_B - q_A$

$25 = q_0 - q_A + q_B + q_A$

$75 = q_0 + q_A + q_B + q_{AB}$

Estas quatro equações podem ser resolvidas para as quatro incógnitas. A equação de regressão é:

$y = 40 + 20\ x_A + 10\ x_B + 5\ x_A\ x_B$

O resultado é interpretado da seguinte forma. O desempenho médio é de 40 MIPS, o efeito de memória é de 20 MIPS, o efeito de memória cache é de 10 MIPS e da interação entre a memória e o cache representa 5 MIPS.

4.9.1. Cálculo dos Efeitos

Em geral, todo o projeto 2^2 pode ser analisado usando o método do Exemplo 4.9. No caso geral, suponha que y_1, y_2, y_3 e y_4 representam as quatro respostas observadas. A correspondência entre os níveis do fator e as respostas é mostrada na Tabela 4.17.

Tabela 4.17 - Análise do delineamento 2^2

Experimento	A	B	y
1	-1	-1	y_1
2	1	-1	y_2
3	-1	1	y_3
4	1	1	y_4

O modelo para um projeto de 2^2 é: $y = q_0 + q_A\ x_A + q_B\ x_B\ q_{AB}\ x_A\ x_B$

Substituindo os quatro observações no modelo, temos:

$y_1 = q_0 - q_A - q_B + q_A$

$y_2 = q_0 + q_A - q_B - q_A$

$y_3 = q_0 - q_A + q_B - q_A$

$y_4 = q_0 + q_A + q_B + q_{AB}$

Resolvendo essas equações para q_i temos:

$q_0 = 1/4(y_1 + y_2 + y_3 + y_4$

$q_A = 1/4(-y_1 + y_2 - y_3 + y_4$

$q_B = 1/4(-y_1 - y_2 + y_3 + y_4$

$q_{AB} = 1/4(y_1 - y_2 - y_3 + y_4)$

Note que as expressões para q_A, q_B, e q_{AB} são combinações lineares das respostas de tal forma que a soma dos coeficientes é zero. Tais expressões são chamadas de contrastes.

Notar também que os coeficientes de y_i na equação para q_A são idênticos aos níveis de A listadas na Tabela 4.14. Assim, q_A pode ser obtido multiplicando as colunas A e y na tabela. Isto também é verdade para q_B e q_{AB}, o que pode ser obtido pela multiplicação das respectivas colunas de nível pela coluna de resposta. Estas observações leva-nos ao método de tabela de sinal para efeitos de cálculo, que é descrito a seguir.

4.9.2. Método de Tabela para Calcular os Efeitos Sinal

Para um projeto de 2^2, os efeitos podem ser facilmente calculado através da preparação de uma matriz de sinal 4×4, como mostrado na Tabela 4.18.

Tabela 4.18 - Método da Tabela de Sinais para o Cálculo dos Efeitos de Projetos 2^2

I	A	B	AB	y
1	-1	-1	1	15
1	1	-1	-1	45
1	-1	1	-1	25
1	1	1	1	75
160	80	40	20	Total
40	20	10	5	Total/4

A primeira coluna da matriz é rotulada I e é constituída por 1’s. As próximas duas colunas, A e B, contêm basicamente todas as combinações possíveis de -1 e 1. A quarta coluna, rotulada AB, é o produto das entradas das colunas A e B. As quatro observações são listadas no vetor coluna desta matriz. O vetor coluna rotulado por y consiste na resposta correspondente aos níveis de fator listados nas colunas A e B.

O próximo passo é multiplicar as entradas na coluna I por aqueles da coluna y e colocar a soma em coluna I. As entradas na coluna A são agora

multiplicado por aqueles da coluna y e a soma é contabilizado em coluna A. Esta operação da multiplicação da coluna é repetida para as outras duas colunas da matriz.

Os valores em cada coluna são divididas por 4 para dar os coeficientes correspondentes do modelo de regressão. Geralmente, 1 não está explicitamente escrito nas entradas da matriz. O sinal de mais ou menos é suficiente para denotar 1 ou -1, respectivamente.

4.9.3. Atribuição da Variação

A importância de um fator é medida pela proporção da variação total na resposta que é explicada pelo fator. Assim, se dois fatores explicam 90 e 5% da variação da resposta, o segundo fator pode ser considerado sem importância em muitas situações práticas.

A variação da amostra de y pode ser calculada como segue: $s_e^2 = \sum(y_i - \ddot{y})^2 / (2^2-1)$

Aqui, ÿ denota a média de respostas de todos os quatro experimentos. O numerador do lado direito da equação acima é chamado a variação total de y ou Soma dos Quadrados Total (SST):

$$SST = \sum(y_i - \ddot{y})^2$$

Para um projeto de 2^2, a variação pode ser dividida em três partes:

$$SST = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$$

Antes de apresentar uma derivação dessa equação, é útil compreender o seu significado. As três partes do lado direito representam a parcela da variação total explicada pelo efeito de A, B e a interação AB, respectivamente. Assim, $2^2 q_A^2$ é a parte da SST que é explicada pelo fator A. Chama-se a soma dos quadrados devido a A e é denotado como SSA. Da mesma forma, SSB é $2^2 q_B^2$ e SSAB (devido à interação AB) é $2^2 q_{AB}^2$. Assim,

$$SST = SSA + SSB + SSAB$$

Estas partes podem ser expressas como uma fração, por exemplo,

Fração da variação explicada por A = SSA/SST, expressa em percentagem, essa fração fornece uma maneira fácil de avaliar a importância do fator A.

Os fatores que apresentam uma elevada percentagem de variação são

considerados importantes. Deve ser salientado que a variação é diferente de variância. Assim, um fator que explica 60% da variação pode ou não pode explicar 60% da variância total de y. A percentagem de variância explicada é bastante difícil de calcular. O percentual de variação, por outro lado, é fácil de computar e fácil de explicar para os tomadores de decisão.

A derivação da Equação para SST segue agora.

Derivação 4.1. O modelo utilizado em um projeto 2^2 é

$$y_i = q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{AiBi}, i \in [1, 2^2] \quad 4.24$$

As colunas x_A, x_B e $x_A x_B$ da matriz de design na Tabela 4.18 tem as seguintes propriedades:

1. A soma das entradas em cada coluna é zero: $\sum x_{Ai} = 0$, $\sum x_{Bi} = 0$ e $\sum x_{Ai} x_{Bi} = 0$.

2. A soma dos quadrados de entradas em cada coluna é 4: $\sum x_{Ai}^2 = 4$, $\sum x_{Bi}^2 = 4$ e $\sum x_{Ai} x_{Bi} = 4$.

3. As colunas são ortogonais uma vez que o produto interno de quaisquer duas colunas é zero: $\sum x_{Ai} x_{Bi} = 0$, $\sum x_{Ai}(x_{Ai} x_{Bi}) = 0$ e $\sum x_{Bi}(x_{Ai} x_{Bi}) = 0$.

Essas propriedades nos permitem calcular a variação total da seguinte forma:
Média da amostra:

$$\ddot{y} = \frac{1}{4}\sum y_i = \frac{1}{4}\sum (q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{AiBi}) \quad 4.25$$

$$\ddot{y} = \frac{1}{4}\sum q_0 + \frac{1}{4}\sum q_A x_{Ai} + \frac{1}{4}\sum q_B x_{Bi} + \frac{1}{4}\sum q_{AB} x_{AiBi} = q_0 \quad 4.26$$

Variação Total:

$$SST = \sum (y_i - \ddot{y})^2 = \sum (q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi} - \ddot{y})^2 \quad 4.27$$

$$SST = \sum (q_A x_{Ai})^2 + \sum (q_B x_{Bi})^2 + \sum (q_{AB} x_{Ai} x_{Bi})^2 + termos\ do\ produto \quad 4.28$$

$$SST = q_A^2 \sum x_{Ai}^2 + q_B^2 \sum x_{Bi}^2 + q_{AB}^2 \sum x_{Ai}^2 x_{Bi}^2 \quad 4.29$$

$$SST = 4 q_A^2 + 4 q_B^2 + 4 q_{AB}^2 \quad 4.30$$

Os termos dos produtos cruzados das equações precedentes são zero devido à ortogonalidade das colunas.

Exemplo 4.10 - No caso do estudo da memória cache

$\ddot{y} = 1/4(15+55+25+75) = 40$

$SST = \sum(y_i-\ddot{y})^2 = \sum(25^2 + 15^2 + 15^2 + 35^2) = 2100$ ou

$SST = 4\times20^2 + 4\times10^2 + 4\times5^2 = 2100$

Assim, a variação total é de 2100, dos quais 1.600 (76%) podem ser atribuídas à memória, 400 (19%) podem ser atribuídos à cache, e apenas 100 (5%) podem ser atribuídas à interação.

A porcentagem de variação ajuda o pesquisador decidir se é ou não de valor para investigar um fator ou interação. Por exemplo, no estudo de memória cache, a variação de 5% devido à interação parece insignificante. O primeiro fator a ser mais aprofundado é o tamanho da memória, o que explica 76% da variação. O cache é menos importante porque explica apenas 19% da variação.

Exemplo 4.11 - Duas redes de memória de interconexão chamada Omega e Crossbar foram comparados utilizando simulação. Dois diferentes padrões de referência de memória de endereço chamada Aleatória e Matrix foram usados. Como o nome indica o padrão de referência endereços aleatórios da memória com uma probabilidade uniforme de referência. O segundo modelo simulou um problema de multiplicação de matrizes em que cada processador (de um sistema multiprocessador) está fazendo uma parte da multiplicação. Para manter a análise simples, muitos fatores que eram conhecidos por afetar o desempenho das redes de interconexão foram mantidos fixos em um nível como segue:

1. Número de processadores foi fixado em 1
2. Solicitações em fila em blocos e não por buffe
3. Comutação de circuitos foi usado em vez de comutação de pacote
4. Arbitragem aleatória foi usada em vez de round robi
5. *Interleaving* infinito de memória foi usada de modo que não havia contenção no banco de memória

Um planejamento 2^2 fatorial foi utilizado. As atribuições dos símbolos é mostrado na Tabela 4.19. Três diferentes métricas de desempenho foram calculadas utilizando simulação: vazão média (T), Tempo de trânsito de 90% em ciclos (N) e tempo médio de resposta (R). O desempenho medido é mostrado na Tabela 4.20. Os efeitos, calculado usando o método de tabela de sinais, são apresentados na Tabela 4.21. A tabela também contém porcentagem de variação explicada.

Tabela 4.19 - Fatores utilizados no estudo da Rede de Interconexão

Símbolo	Fator	Nível -1	Nível +1
A	Tipo de Rede	Crossbar	Omega
B	Padrão de Endereço Utilizado	Aleatório	Matrix

Tabela 4.20 - Respostas medidas no Estudo da Rede de Interconexão

A	B	Resposta		
		T	N	R
-1	-1	0.6041	3	1.655
1	-1	0.4220	5	2.378
-1	1	0.7922	2	1.262

1	1	0.4717	4	2.190

Tabela 4.21 - Efeitos médios para o Estudo da Rede de Interconexão

Parâmetros	Estimativa da Média			Variação Explicda (%)		
	T	N	R	T	N	R
q_0	0.5725	3.5	1.871			
q_A	0.0595	-0.5	-0.145	17.2	20	10.9
q_B	-0.1257	1.0	0.413	77.0	80	87.8
q_{AB}	-0.0346	0.0	0.051	5.8	0	1.3

Os resultados são interpretados da seguinte forma:

- A vazão média é de 0,5725. O rendimento é em grande parte afetada pelo padrão de referência, que faz uma diferença de ±0,1257 e, portanto, explica 77% da sua variação. O tipo de rede contribui 0,0595 para o throughput. A Rede Omega é muito superior à média, e a Rede Crossbar é muito inferior à média. Assim, a diferença líquida entre os dois tipos de redes é 0,119. A escolha da rede é afetada pelo padrão de endereço uma vez que existe uma pequena interação. Dependendo do padrão de endereço e combinação de rede, o rendimento pode subir ou descer por 0,0346.
- O tempo de trânsito de 90% também é afetado principalmente pelo padrão de endereço. Como q_A é negativo, o tempo de trânsito é maior para A = -1 ou Rede Crossbar. Isso se aplica tanto padrões de endereço, pois não há interação entre o padrão de endereço e o tipo de rede.
- O tempo de resposta também depende principalmente do padrão de endereço. A interação entre o tipo de padrão e da rede é baixa.

Assim, percebe-se que todas as três métricas são mais afetadas pelos padrões de endereço do que pelo tipo de rede. Isso ocorre porque os padrões de endereço escolhidos são muito diferentes.

4.10. Experimento Fatorial 2^k Geral

Um projeto experimental 2^k é utilizado para determinar o efeito de k fatores, cada um dos quais tem duas alternativas ou níveis. Nós já discutimos o caso especial de dois fatores (k = 2) nas duas últimas seções. Agora vamos generalizar a análise para mais de dois fatores.

As técnicas de análise desenvolvidas até agora para projetos 2^2 podem ser estendidas para um projeto de 2^k. Dados k fatores em dois níveis cada, são obrigatórios um total de 2^k experimentos. A análise produz 2^k efeitos. Estes incluem os k efeitos principais, $C_{k,2}$ interações de dois fatores, $C_{k,3}$ interações de três fatores, e assim por diante. O método de tabela de sinal para analisar os resultados e alocar a variação também é válido. Será ilustrado com um exemplo.

Exemplo 4.12 - Para projetar uma máquina LISP, os três fatores que precisam ser estudadas são: tamanho do cache, tamanho de memória e se um ou dois processadores serão usados. Os três fatores e suas atribuições de nível são mostrados na Tabela 4.22. O projeto 2^3 e do desempenho medido em MIPS é mostrada na Tabela 4.23.

Tabela 4.22 - Fatores e Níveis no Exemplo 4.12

Fator	Nível -1	Nível +1
Tamanho da memória, A	4 GB	16 GB
Tamanho do cache, B	1 KB	2 KB
Número de CPU, C	1	2

Tabela 4.23 - Resultados de uma experiência 2^3

Tamanho do Cache (KB)	4 GB		16 GB	
	1 CPU	2 CPU's	1 CPU	2 CPU's
1	14	46	22	58
2	10	50	34	86

Para analisar o experimento, foi preparada uma tabela de sinais como mostrado na Tabela 4.24. Como mostrado na última linha desta tabela, os efeitos de memória cache e os processadores são $q_A = 10$, $q_B = 5$, e $q_C = 20$, respectivamente. As três interações de dois fatores são $q_{AB} = 5$, $q_{AC} = 2$, e $q_{BC} = 3$. O q_{ABC} interação de três fatores é 1.

Tabela 4.24 - Um exemplo de Tabela Sinais

I	A	B	C	AB	AC	BC	ABC	y
1	-1	-1	-1	1	1	1	-1	14
1	1	-1	-1	-1	-1	1	1	22
1	-1	1	-1	-1	1	-1	1	10
1	1	1	-1	1	-1	-1	-1	34
1	-1	-1	1	1	-1	-1	1	46
1	1	-1	1	-1	1	-1	-1	58
1	-1	1	1	-1	-1	1	-1	50
1	1	1	1	1	1	1	1	86
320	80	40	160	40	16	24	9	Total
40	10	5	20	5	2	3	1	Total/8

A parcela da variação explicada por vários fatores e interações são proporcionais ao quadrado dos efeitos. A SST pode ser calculado por meio dos efeitos da seguinte forma:

$$SST = 2^3(q_A^2 + q_B^2 + q_C^2 + q_{AB}^2 + q_{AC}^2 + q_{BC}^2 + q_{ABC}^2)$$

$$SST = 8\ (102 + 52 + 202 + 52 + 22 + 32 + 12)$$

$$SST = 800 + 200 + 3200 + 200 + 32 + 72 + 8 = 4512$$

A parcela da variação explicada pelos sete efeitos são 800/4512 (18%), 200/4512 (4%), 3200/4512 (71%), 200/4512 (4%), 32/4512 (1%), 72/4512 (2%) e 8 / 4512 (0%), respectivamente.

4.11.Exercícios

1. Demonstre que $\sum(y_i-y_m)^2 = \sum(\hat{y}_i-y_m)-\sum(y_i-\hat{y}_i)^2$
2. O desempenho de um sistema que está sendo projetado depende dos seguintes três fatores:
   a) CPU tipo: Intel, AMD, Apple
   b) Tipo de sistema operacional: Windows, Linux, Unix
   c) Tipo de unidade de disco: A, B
3. Quantos experimentos são necessárias para analisar o desempenho se
   a) Há interação significativa entre os fatores
   b) Não há interação entre os fatores
   c) As interações são pequenas comparadas com os efeitos principais
4. Analise o desenho 2^3 mostrado na Tabela 4.25.
   a) Quantifique os efeitos principais e as interações
   b) Quantifique as porcentagens de variação explicada
   c) Classificaque as variáveis na ordem decrescente de importância

Tabela 4.25 - Um projeto 2^3

	A_1		A_2	
	C_1	C_2	C_1	C_2
B_1	100	15	120	10
B_2	40	30	20	50

5. Referências

Jain, R. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley-Interscience, 1991.

Menascé, Daniel A.; Almeida, Virgílio A. F. Planejamento de Capacidade para Serviços na Web: Métricas, modelos e métodos. Rio de Janeiro: Campus, 2003.

HINES, W. W. et al. Estatística Aplicada e Probabilidade para Engenheiros. LTC, 2003.
PRADO, D. Teoria das Filas e da Simulação. 2ª ed. IDNG, 2004.
LAW A. M. Simulation Modeling and Analysis. Pearson Education, 2006.