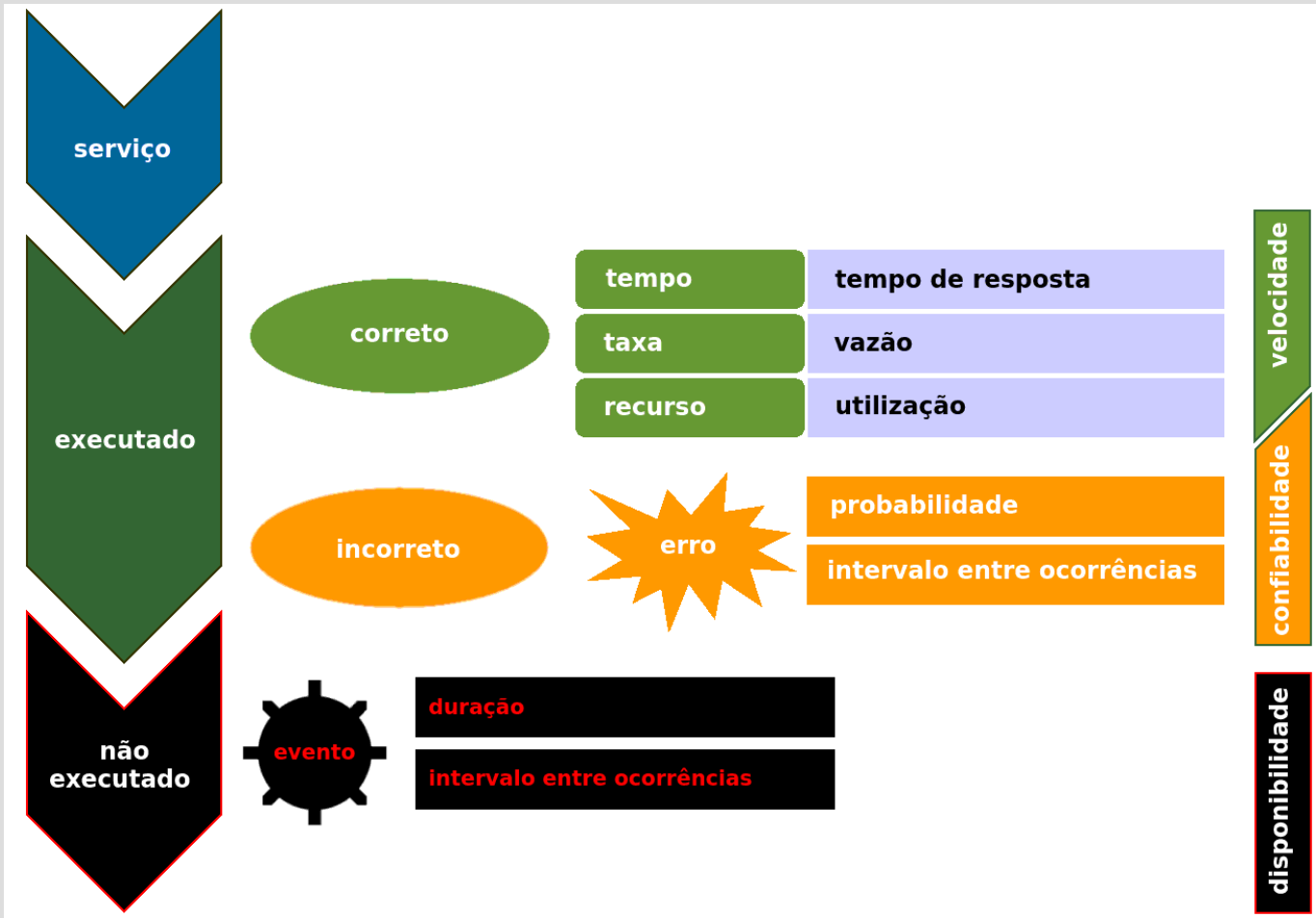


Brauliro G Leal

Avaliação de Desempenho de Sistemas



Quarta Edição

Juazeiro - BA  
2023

Copyright© 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023 by Brauliro Gonçalves Leal

O conteúdo deste livro eletrônico é totalmente livre para uso de qualquer natureza desde que citado a fonte. Toda e qualquer parte desta publicação pode ser reproduzida, distribuída ou transmitida de qualquer forma ou por qualquer meio, ou armazenada de qualquer forma ou em qualquer sistema desde que reconhecida a autoria.



Atribuição-Compartilhalgual - esta licença permite que outros remixem, adaptem e criem a partir deste trabalho, mesmo para fins comerciais, desde que lhe atribuam o devido crédito e que licenciem as novas criações sob termos idênticos ([creativecommons.org/licenses](https://creativecommons.org/licenses)).

e-mail: [ccmp0077@gmail.com](mailto:ccmp0077@gmail.com)  
Quarta Edição Eletrônica: abril de 20232  
ISBN: a ser feito#

O autor

Brauliro Gonçalves Leal

Professor do Colegiado de Engenharia da Computação  
Universidade Federal do Vale do São Francisco  
Avenida Antônio Carlos Magalhães, 510 Santo Antônio  
Juazeiro/BA – Brasil      48.902-300

e-mail: [brauliro.leal@univasf.edu.br](mailto:brauliro.leal@univasf.edu.br)  
site: [www.univasf.edu.br/~brauliro.leal](http://www.univasf.edu.br/~brauliro.leal)

Sobre este documento

Este documento foi escrito em LibreOffice ([www.libreoffice.org](http://www.libreoffice.org)), suas imagens foram produzidas por Kolourpaint ([www.kolourpaint.org](http://www.kolourpaint.org)), seus programas foram editados na IDE Code::Blocks ([www.codeblocks.org](http://www.codeblocks.org)) e compilados com o GNU GCC ([gcc.gnu.org](http://gcc.gnu.org)). O Sistema Operacional utilizado foi GNU/Linux Mint Cinnamon.

Software livre é a melhor ideia.



# 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 perguntas centrais deste assunto podem ser respondidas?
- Quais são seus conceitos fundamentais?
- Como e quando usar os conceitos da Avaliação de Desempenho de Sistemas?
- Como a análise de casos particulares podem levar a conclusões mais amplas?
- Quais respostas não se pode encontrar a partir destes conceitos?
- Quais são suas limitações?

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

Do mesmo modo que frases verbais são organizadas em sentenças, e que as sentenças são organizadas em parágrafos que, por sua vez, são constituídas de palavras; a modelagem matemática pode ser organizada em sequências e estruturas tomadas das mais simples para as mais complexas, e vice-versa. Assim, sua linguagem pode ser aprendida suficientemente bem para que se compreenda a maioria das suas afirmações embora não sejam tão simples. E muito mais, alimentando a intuição dos leitores e os capacitando na formação de conceitos e imagens visuais dos processos inerentes a ela, algo que as pessoas já fazem em alguma medida no dia a dia. Além de introduzir os assuntos de forma incremental de modo a formar conceitos e aplicá-los.

O ambiente empresarial requer dados quantitativos e qualitativos, de modo a gerar conteúdos importantes e informações fundamentais. Não é concebível fazer parte de uma equipe que atua às cegas, sem informações, sem relatórios, sem análise ou balanceamento, sem saber o que gera mais retorno e o que pode prejudicar o desempenho.

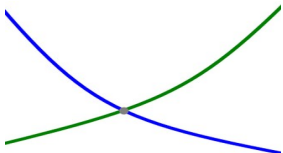
Com acesso à informação relevante, pode-se definir, por exemplo, as melhores estratégias e decisões a serem tomadas, montar um planejamento de acordo com os resultados obtidos

através dos dados, gerar maior vantagem competitiva e conseguir informações essenciais para o desempenho da empresa. Dessa forma, é possível fazer um estudo real de uma empresa e antever quais caminhos produzem melhores resultados. Mas, para que isso ocorra da melhor forma, é necessário saber interpretá-los.

Peço antecipadas desculpas por eventuais erros neste texto, que certamente existem, agradeço antecipadamente toda e qualquer correção ou sugestão de melhoramentos, que deverão ser enviadas para o e-mail da disciplina: [ccmp0077@gmail.com](mailto:ccmp0077@gmail.com).

Nota
<p>A Avaliação de Desempenho de Sistemas está em constante evolução. À medida que novas pesquisas e a própria experiência ampliam o nosso conhecimento, novas descobertas são realizadas. O autor desta obra buscou oferecer informações adequadas e de acordo com os padrões aceitos à época da sua publicação.</p>

O ótimo é o inimigo do bom.  
Voltaire



Sumário

- 1 Avaliação de Desempenho de Sistemas.....9
  - 1.1 Exercícios.....14
- 2 Métricas de Avaliação de Desempenho de Sistemas.....15
  - 2.1 Métricas de Velocidade.....17
    - 2.1.1 Tempo de Resposta.....17
    - 2.1.2 Vazão.....18
    - 2.1.3 Utilização.....20
  - 2.2 Métricas de Confiabilidade.....20
  - 2.3 Métricas de Disponibilidade.....20
  - 2.4 Outras Métricas.....21
  - 2.5 Caracterização das Métricas de Desempenho.....21
  - 2.6 Exercícios.....25
- 3 Medição.....27
  - 3.1 Técnicas e Ferramentas de Medição.....27
  - 3.2 Seleção e Caracterização de Carga.....28
  - 3.3 Monitores.....29
    - 3.3.1 Terminologia para Monitor.....31
    - 3.3.2 Classificação de Monitores.....31
    - 3.3.3 Monitores em Software.....32
    - 3.3.4 Monitores em Hardware.....32
    - 3.3.5 Monitores de Software versus de Hardware.....33
    - 3.3.6 *Firmware* e Monitores Híbridos.....35
  - 3.4 Benchmarking.....36
    - 3.4.1 *Benchmarks* Populares.....37
    - 3.4.2 *Benchmark* do SPEC.....37
    - 3.4.3 Benchmark de Roy Longbottom.....38
    - 3.4.4 UserBenchmark.....39
    - 3.4.5 PassMark.....39
- 4 Apresentação dos Resultados.....41
  - 4.1 Sumarização.....41
  - 4.2 Resumindo a Variabilidade.....42
  - 4.3 Resumindo Dados por Meio de Gráficos.....45
  - 4.4 Modelos de Regressão.....46
    - 4.4.1 Estimação dos Parâmetros do Modelo.....48
    - 4.4.2 Análise de Variância.....50
    - 4.4.3 Soma de Quadrados.....50
    - 4.4.4 Graus de Liberdade.....51
    - 4.4.5 Quadrado Médio.....51
- 5 Teoria das Filas.....55
  - 5.1 Sistema de Filas.....56
  - 5.2 Rede de Sistemas de Filas.....57
  - 5.3 Aplicações.....59
- 6 Leis Operacionais.....61
  - 6.1 Representando RSF.....61
  - 6.2 Organização Interna das RSF.....62
  - 6.3 Condições de Denning e Buzen.....63
  - 6.4 Quantidades Operacionais.....63
  - 6.5 Lei da Utilização.....64
  - 6.6 Lei do Fluxo.....65
  - 6.7 Lei de Little.....68
  - 6.8 Lei do Tempo de Resposta.....69
  - 6.9 Análise de Gargalo.....70
  - 6.10 Lei do Tempo de Resposta Interativo.....72
- 7 Análise de Valor Médio (AVM).....75
  - 7.1 Algoritmo da Análise do Valor Médio.....79

- 7.2 Análise do Valor Médio Aproximada.....85
- 8 Projeto e Análise de Experimentos.....90
  - 8.1 Experimento com Um Fator.....93
    - 8.1.1 Modelo.....95
    - 8.1.2 Cálculo dos Efeitos.....100
    - 8.1.3 Estimativa dos Erros Experimentais.....103
    - 8.1.4 Atribuição da Variação.....104
    - 8.1.5 Análise da Variação.....106
  - 8.2 Experimento Fatorial 2<sup>2</sup>.....108
    - 8.2.1 Cálculo dos Efeitos.....109
    - 8.2.2 Método da Tabela de Sinais.....110
    - 8.2.3 Atribuição da Variação.....110
  - 8.3 Experimento Fatorial 2<sup>k</sup>.....114
  - 8.4 Exercícios.....117
- 9 Referências.....119
- 10 Apêndice.....120
  - 10.1 Programa C++ para calcular os efeitos e a variação em experimentos 2<sup>k</sup> fatoriais. .120
  - 10.2 Programa html para calcular os efeitos e a variação em experimentos 2<sup>k</sup> fatoriais...123



# 1 Avaliação de Desempenho de Sistemas

Todo e qualquer Avaliação de Desempenho tem como objetivo principal auxiliar a tomada de decisão, devem ser capazes de fornecer informações tanto das partes (empresas) como do contexto do sistema avaliado. Neste sentido, as empresas estão inseridos na sociedade, no meio ambiente e nas questões estruturais corporativas.

Dentre os grandes sistemas atuais, os Sistemas Computacionais prestam relevantes serviços para a sociedade, entretanto eles são úteis na medida em que seus resultados e seu comportamento estão dentro das metas de desempenho e comportamento pré-definidos.

Os Sistemas Computacionais não podem ser considerados onipresentes, mas seus serviços mudaram a paisagem socioeconômica de grande parte do globo. Suas aplicações cotidianas se intensificam, produzindo cada vez mais um mundo interconectado, substituindo modos de trabalhar e modernizando processos. São fontes de criação e renovação contínua. Pode-se observar na Tabela 1, o crescimento global do número médio de dispositivos e conexões per capita no mundo e por região (Cisco, 2020); estes valores aumentarão em todos as regiões, variando de 36,4 a 67,9%, com previsão de cerca de 28 bilhoes de dispositivos conectados no mundo em 2023. A Tabela 1 ilustra a projeção, para 2021, da interconexão da Internet no mundo (Tecnoblog, 2020), com o total projetado igual a 8.200 Tb/s; o crescimento indicado tem como base o ano de 2018.

Tabela 1 Número médio de dispositivos e conexões per capita, os dados ano de 2023 são projeções

Região	2018	2023	Variação (%)
América do Norte	8,2	13,4	63,4
América Latina	2,2	3,1	40,9
Ásia-Pacífico	2,1	3,1	47,6
Europa Central e Oriental	2,5	4,0	60,0
Europa Ocidental	5,6	9,4	67,9
Oriente Médio e África	1,1	1,5	36,4
Global	2,4	3,6	50,0

Fonte: Adaptado de Cisco (2020).

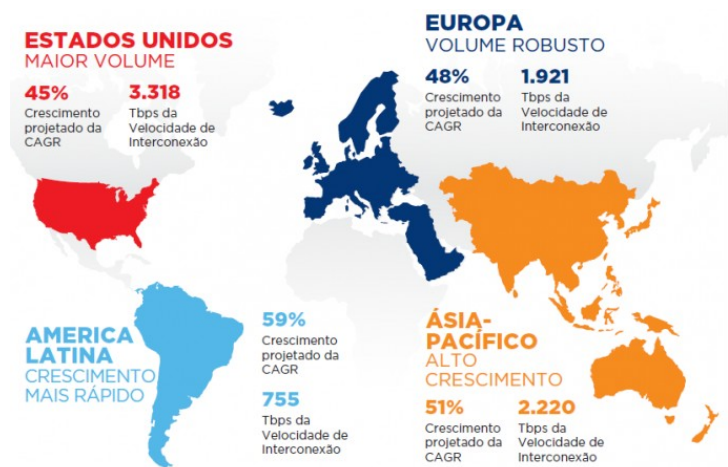


Figura 1 A interconexão da Internet pelo mundo – projeção para 2021 tendo como base 2018 (Tecnoblog, 2020).

Os Sistemas Computacionais existentes precisam ser mantidos ou atualizados; os novos devem ser projetados e construídos. Uma forma de fazê-lo é por meio da Avaliação de Desempenho de Sistema (ADS) que trata sistemas existentes, permitindo reprojotá-los e reconfigurá-los; e também trata de sistemas inexistentes, permitindo projetá-los. O modelo de entrega de produtos e serviços da Engenharia da Computação são seus projetos de hardware e/ou softwares e suas técnicas de produção.

Exemplo 1.1 Caracterização Geral dos Sistemas Computacionais

Sistemas Computacionais não são entidades naturais, eles são artefatos produzidos pela Técnica Humana, são desenvolvidos com base na Ciência dos últimos 400 anos.  
Os Sistemas Computacionais são dispositivos eletrônicos (hardware) capazes de processar informações de acordo com programa (software) neles instalados. O hardware pode conter várias camadas de software a depender do serviço, ou serviços, prestados por ele. A união do hardware com o software permite soluções configuráveis e, integrados com a Internet, podem interagir e formar as redes mais diversas.  
Os Sistemas Computacionais são projetados para fins específicos, sendo cada funcionalidade detalhada nos manuais técnicos e datasheets, fornecidos pelo fabricante. Estes sistemas evoluem e novas versões se sucedem. Eles também se integram cada vez mais, formando redes e redes de redes, conectando outras estruturas sociais, urbanas, nacionais e internacionais e, enfim, todo planeta.

Cada avaliação de desempenho requer um conhecimento detalhado do sistema (real ou projetado) e uma escolha cuidadosa de metodologias, cargas e ferramentas de análise. Em geral, os problemas chegam ao engenheiros, por vezes, mal definidos e, transformá-los em enunciados tratáveis, requer mais arte do que engenharia. Para um mesmo sistema, pode-se obter diferentes avaliações, que dependendo do detalhamento do problema, da experiência, do estilo e da intuição do engenheiro. Tendo em vista estes aspectos, Jain (1991) considera o processo de Avaliação de Desempenho de Sistemas também uma arte.

Exemplo 1.2 Análise de Desempenho de Sistemas Computacionais

Sistemas Computacionais são constituídos por dispositivos de hardware e software. O hardware corresponde às partes eletrônicas e mecânicas que possibilitam a existência do software e a interação com o usuário. O software fornece as interfaces para que usuários executem seus pacotes. Estes sistemas realizam atividades humanas, ou de outros sistemas computacionais, por meio do processamento de pacotes. Os Sistemas Computacionais são caracterizados, grosso modo, pelos elementos: hardware, software, usuários e pacotes. A importância dos Sistemas Computacionais reside no seu conjunto de capacidades e comportamentos, as quais caracterizam seu desempenho em nível operacional. A Avaliação de Desempenho de Sistemas no âmbito das engenharias pode ser realizada tanto por medição quanto por meio da modelagem matemática. A partir de um modelo matemático aceitavelmente e representativo de um sistema, pode-se aplicar técnicas matemáticas para descrever o seu comportamento e também estimar sua performance comparada a indicadores previamente estabelecidos. A relevância da Avaliação do Desempenho dos Sistema Computacionais reside na sua abrangência cada vez maior. A interação entre pessoas, empresas e governos, mediada pela tecnologia tanto em nível local quanto global, é um novo estímulo para mudanças sociais e econômicas, promovendo cada vez mais a produção de riqueza; além de ser o suporte para a criação de conhecimento com a aplicação da Inteligência Artificial, Ciência dos Dados e Internet das Coisas, dentre outras áreas aplicadas.

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 envolvem 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.

Exemplo 1.3 Internet das coisas

Internet das Coisas - IdC (em inglês: Internet of Things - IoT) é um conceito que se refere à interconexão digital de objetos cotidianos com a internet, conexão dos objetos mais do que das pessoas.

A IdC é uma rede de objetos físicos (veículos e os outros dispositivos móveis, prédios, sensores e aqueles com tecnologia embarcada e conexão com a rede) capaz de reunir e de transmitir dados. Ela se integra à internet na medida que estes objetos possuam capacidade de comunicação.

A IdC, e seus objetos constituintes, requer comunicação e capacidade de processamento e armazenamento de dados. Pode ser um sistema básico de coleta e transmissão de dados ou possuir grande capacidade nestes quesitos. Esta rede de recursos possibilita controle e gerenciamento dos seus objetos, além de prover serviços segundo suas capacidades.

A IdC surgiu em consequência dos avanços de várias áreas – sistemas embarcados, microeletrônica, comunicação e sensoriamento. Ela tem potencial de uso nas mais diversas áreas das atividades humanas, principalmente quando associada à Inteligência Artificial,

Computação em Nuvem e Ciência de Dados, dentre outras áreas emergentes. Com a IdC o mundo fica cada vez mais nivelado pelos dados e as divisões históricas, regionais e geográficas vão ficando cada vez mais irrelevantes.  
Fonte: Wikipédia.

A avaliação de desempenho requer métricas, técnicas e ambiente de medição. A seleção destes quesitos são passos essenciais e 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.

A avaliação de desempenho de sistemas de computação faz uso da simulação, modelagem analítica e medição, Figura 2.

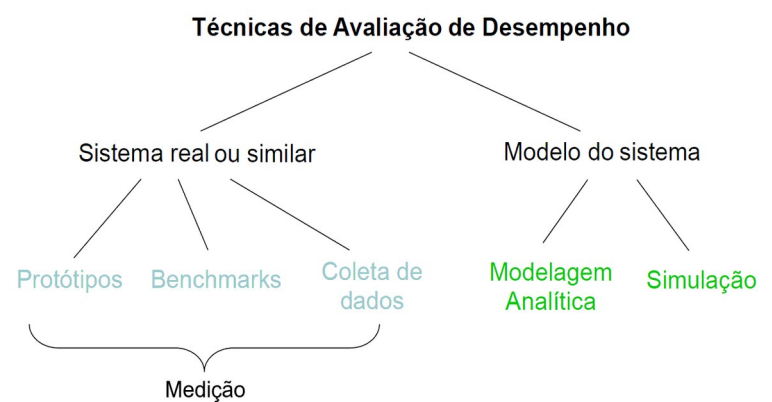


Figura 2 Técnicas de Avaliação de Desempenho.

Há uma série de considerações que ajudam a decidir a técnica a ser utilizada, listadas na Tabela 2 e ordenadas do mais para o menos importante.

Tabela 2 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	programação	instrumentação
Precisão	baixo	moderado	varia
Avaliação do <i>trade-off</i> <sup>1</sup>	fácil	moderado	difícil
Custo	médio	pequeno	alto
Negociação	média	baixa	alta

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

A questão fundamental para decidir a técnica de avaliação é a fase do ciclo de vida do sistema. 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. A Modelagem Analítica e Simulação também podem ser usadas nas situações em que a medição não é possível, mas, em geral, é mais convincente para as 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. Simulações podem levar 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 pode 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.

## 1.1 Exercícios

## 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 3.

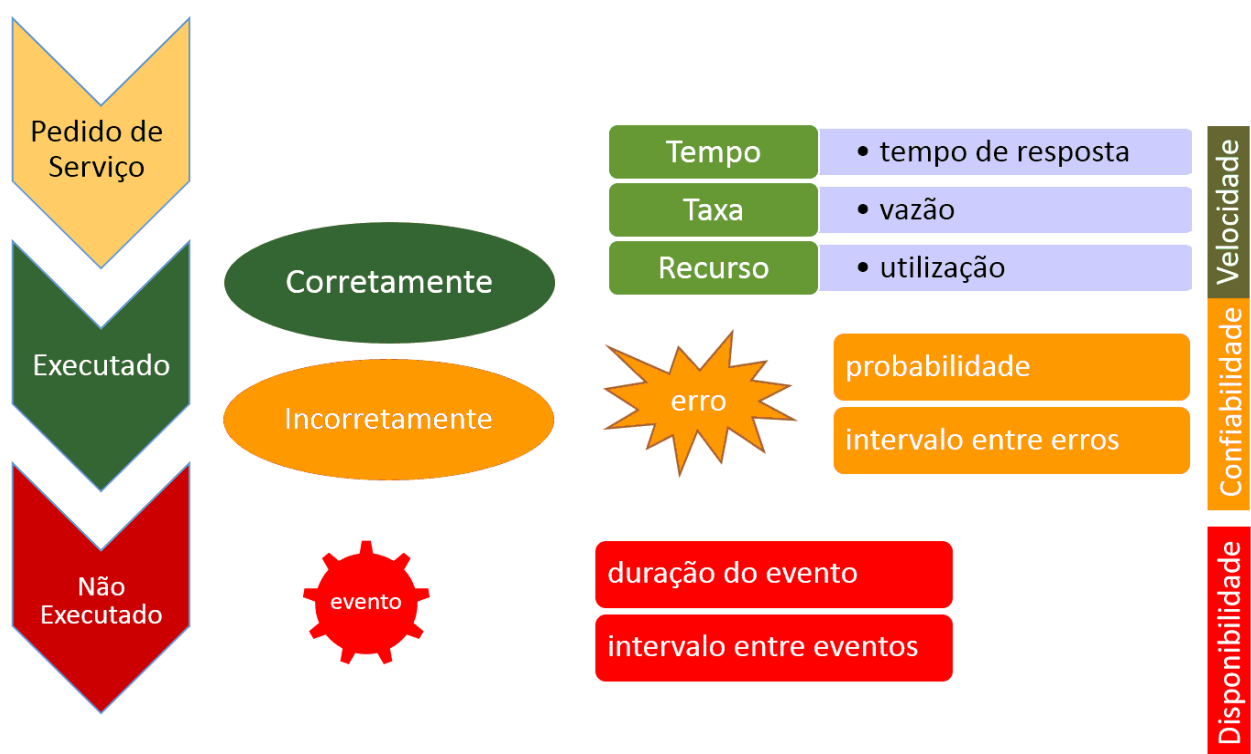


Figura 3 Três possíveis resultados de uma solicitação de serviço.

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

- corretamente, sucesso
- incorretamente, fracasso parcial
- não ser executado, fracasso total

As métricas associadas aos três resultados da Figura 3 são: 1) serviço bem-sucedido; 2) serviço com erro; e 3) serviço 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.

Um único erro ou falha pode comprometer todo o funcionamento do sistema. Para evitar essa situação, há os sistemas tolerantes a falhas, 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.

## 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.

### 2.1.1 Tempo de Resposta

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

Os valores médios do tempo de resposta podem ser influenciados pelos seus valores extremos. Pacotes com tempo de respostas pequenos e outros 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 pode ser definido como o intervalo entre o pedido do usuário e a resposta do sistema, como mostrado na Figura 4. Pode-se definir o tempo de resposta de outras formas. Esta definição é a mais simples e de ordem prática.



Figura 4 Definição do tempo de resposta para pedidos e respostas instantâneos.

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*).

### 2.1.2 Vazão

Vazão ou *throughput* é definida como a taxa temporal em que os pacotes são atendidos pelo sistema.

Em geral, cada sistema de computação tem sua unidade de vazão. Para o processamento em lotes a vazão é medida em pacotes por segundo. Para sistemas interativos a vazão é medida em solicitações por segundo. Para CPU a vazão 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 é medida em transações por segundo (TPS).

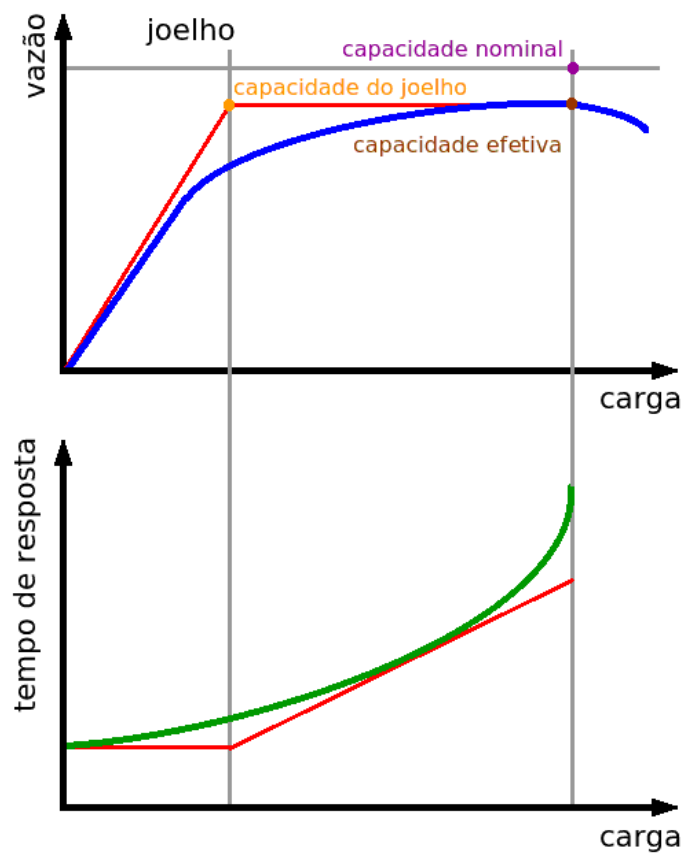


Figura 5 Capacidade de um sistema de computação.

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 5. A taxa de transferência máxima possível sob condições ideais de trabalho é chamada de capacidade nominal (*nominal capacity*) do sistema.

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-especificada. 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*).

A relação entre taxa máxima alcançável (capacidade utilizável) e a capacidade nominal é chamada eficiência (*efficiency*).

Por exemplo, se a taxa de transferência máxima de uma LAN (*Local Area Network*) é igual a 100 Mbps e sua capacidade utilizável é de apenas 85 Mbps, sua eficiência é de 85%.

### 2.1.3 Utilização

A utilização (*utilization*) de um recurso é medida como a fração do tempo utilizado pelo recurso (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, 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.

## 2.2 Métricas de 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.

Por exemplo, no caso da internet 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*).

## 2.3 Métricas de 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.

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, um site pode não estar disponível em 0,01% do tempo devido a uma falha local e em 0,03% devido a uma falha na rede.

## 2.4 Outras Métricas

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.

## 2.5 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 3 estão as considerações usadas para selecionar um subconjunto de métricas de desempenho.

Tabela 3 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
completude	o conjunto de métricas incluídas no estudo deve ser completo. Todos os resultados importantes devem ser refletidos no 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 4 e a Figura 6 mostra gráficos hipotéticos de utilidade de três classes de métricas.

Tabela 4 Classes de métricas de desempenho de sistemas computacionais	
menor é melhor	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 ( <i>lower is better</i> )
nominal é melhor	ambos os valores altos e baixos são indesejáveis. Um valor intermediário entre os extremos é considerado o melhor. A utilização é um exemplo de uma métrica NB ( <i>nominal is best</i> )
maior é melhor	os usuários do sistema e administradores de sistemas preferem os maiores valores de tais métricas. Vazão do sistema é um exemplo de uma métrica HB ( <i>higher is better</i> )

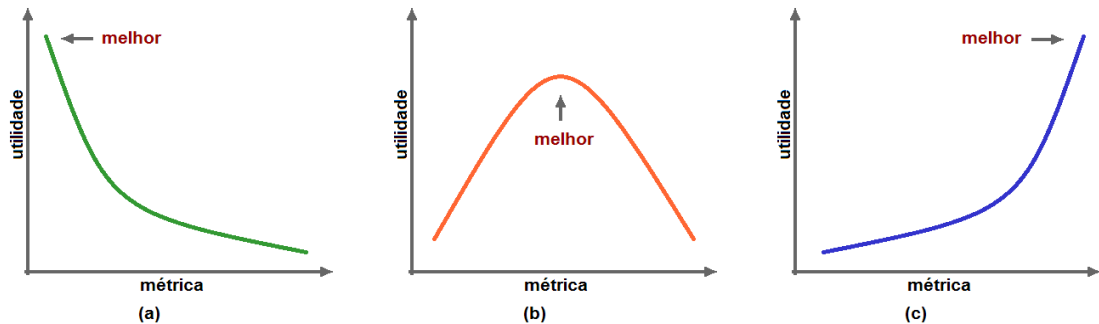


Figura 6 Tipos de métricas: a) menor é melhor (LB); b) nominal é melhor (NB); e c) maior é melhor (HB).

As métricas de desempenho também devem ser específicas, mensuráveis, viáveis e possuir eficácia, conforme é descrito na Tabela 5. Em conjunto, as métricas de desempenho devem possuir tanto características quantitativas quanto qualitativas.

Tabela 5 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

As qualidades das métricas de desempenho podem ser resumidas em uma palavra: SMART, ou seja, os requisitos devem ser específicos, mensuráveis, aceitáveis, realizáveis e eficazes (Specific, Measurable, Acceptable, Realizable and Thorough Specificity).

Exemplo 2.1 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:

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

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

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:

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

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 probabilidade de um bit estar errado deve ser inferior a  $1.0 \times 10^{-7}$
- a probabilidade de entrega de pacote com erro com indicação de erro definida deve ser inferior a 1%

- a probabilidade de entrega de pacote com erro sem indicação de erro deve ser inferior a  $1.0 \times 10^{-15}$
- a probabilidade de entrega de pacote com erro detectado no endereço de destino deve ser inferior a  $1.0 \times 10^{-18}$
- a probabilidade de entrega de pacote ser entregue mais de uma vez (duplicado) deve ser inferior a  $1.0 \times 10^{-5}$
- a probabilidade de perda de pacote na LAN, devido a todos os tipos de erros, deve ser inferior a 1%

Disponibilidade: dois modos de falha foram considerados significativos. O primeiro foi o tempo perdido devido à reinicializações 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:

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

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 6.

Exemplo 2.2 Desempenho de Redes Digitais (Network Performance)

O desempenho de redes digitais refere-se a medidas de qualidade do serviço de uma rede como visto pelo usuário. Devido à diversidade das redes, as maneiras de medir o desempenho de uma rede varia conforme sua estrutura, composição, design e funcionalidades. A modelagem e a simulação é uma alternativa à medição, que pode ser onerosa ou impraticável em algum aspecto.

As medidas de desempenho mais gerais, aplicadas à grande parte das redes, são listadas a seguir:

- **Taxa de Transferência** (throughput, vazão) - medida em bits/s, é a velocidade efetiva que dados podem ser transferidos através do canal de comunicação, é o número de bits corretos entregues por unidade de tempo
- **Taxa de Erro** - adimensional, é a razão entre o número de bits com erros e o número total de bits transferidos durante dado intervalo de tempo.



- **Latência** – medida em segundos, é o atraso entre o envio e a recepção dos dados devido ao meio, por exemplo, o sinal eletromagnético requer aproximadamente 3 milissegundos de atraso para cada 100 km de percurso devido à velocidade da luz. Há também atrasos devido ao processamento dos dados em nós a longo do percurso e aos protocolos de comunicação

Estes fatores exercem influenciam uns nos outros, dependem do meio de comunicação e das tecnologias empregadas. A latência pode ser estimada pelo comando ping, um utilitário para testar a conectividade entre equipamentos. A taxa de erro e a vazão devem ser medidas, valores úteis podem ser encontrados na literatura técnico/científica da área. Em grande medida, é o tempo de resposta que dá ao usuário a percepção da qualidade da rede, quanto menor melhor. Observe que a vazão, taxa de erros, latência e jitter estão relacionados ao tempo de resposta e, quanto maior for a vazão e menor forem os demais, menor será o tempo de resposta da rede. Como a percepção tem seus aspectos psicológicos e históricos, é difícil estabelecer limites superiores para o tempo de resposta. É aceitável longos tempos de resposta de sistemas que envolve segurança, como sistemas bancários e alguns órgãos governamentais. Em geral, os usuários ficam insatisfeitos com espera de qualquer tipo. Felizmente, a tecnologia tem avançado e a infraestrutura de comunicação tem melhorado. Há dois outros conceitos relacionados ao desempenho das redes: 1) Largura de Banda (bandwidth) - medida em bits/s, é a velocidade máxima que dados podem ser transferidos através do canal de comunicação; e 2) Jitter - medida em segundos, está relacionado à variação da latência. Fonte: Baseado na Wikipédia.

Tabela 6 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

## 2.6 Exercícios

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

- duas unidades de discos
- dois sistemas de processamento de transações
- dois algoritmos de retransmissão de pacote
- dois circuitos eletrônicos

2. O número de pacotes perdidos em dois links foi medido para quatro tamanhos de arquivo, como mostrado na Tabela 7. Qual link é o melhor?

Tabela 7 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:

- selecionar um computador pessoal
- selecionar 1.000 postos de trabalho para uma empresa
- comparar dois pacotes de planilha
- 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:

- dois computadores pessoais
- dois sistemas de banco de dado
- duas unidades de disco

## 3 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.

### 3.1 Técnicas e Ferramentas de Medição

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:

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

### 3.2 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
- podem ser incorporadas como funções internas de medição

A Tabela 8 apresenta os principais tipos de cargas de trabalho de teste que tem sido utilizadas para comparar sistemas de computação.

Tabela 8 Cargas de trabalho de teste utilizada para comparar sistemas de computação

<b>instruções mistas</b>	simulam a demanda de utilização dos recursos de um sistema por meio de um conjunto de instruções do seu processador
<b>kernel</b>	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
<b>programas sintéticos</b>	simulam a demanda de utilização dos recursos do sistema de maneira requerida pela carga, são usados no sistema como carga-piloto reproduzível
<b>benchmarks</b>	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

A carga de trabalho é a parte mais importante de qualquer projeto de avaliação de desempenho. 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.

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.

### 3.3 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 podem ser entendidos também como um middleware pois conectam o sistema monitorado e seus recursos, indicando sua utilização. A Figura 7 é o resultado do Monitor da CPU do GNU/Linux LMDE 4ste caso o recurso monitorado é a CPU e seus núcleos e o sistema monitorados são os softwares e hardwares que demandam os recursos da CPU.

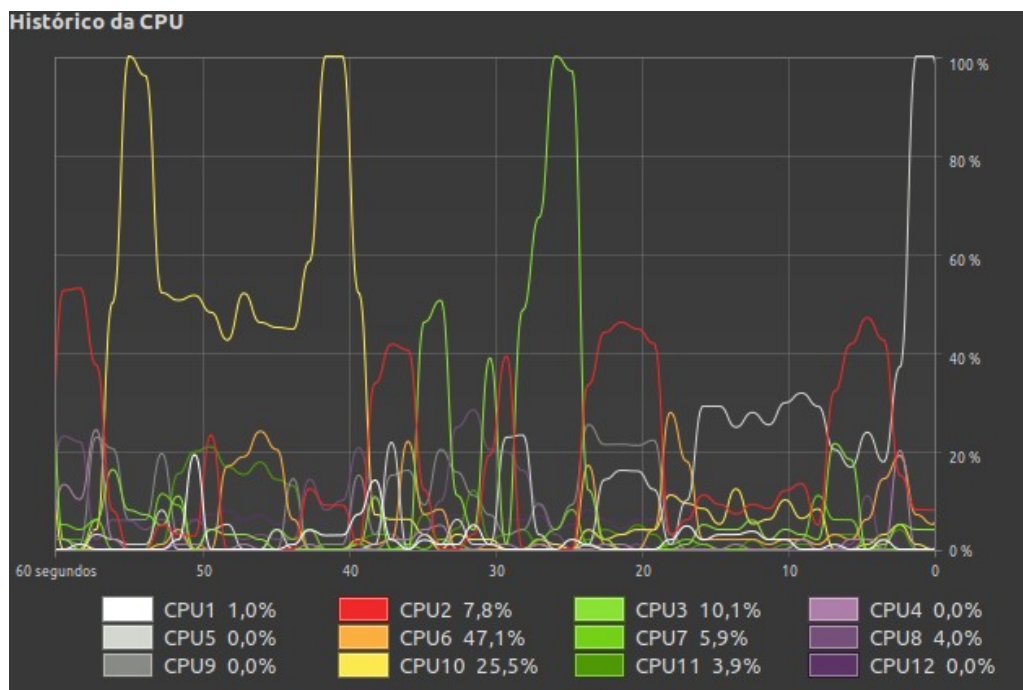


Figura 7 Monitoramento do Histórico da CPU feita pelo Linux LMDE 4.

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

Exemplo 3.1 Google Analytics

O Google Analytics (GA) é um sistema para monitoramento de tráfego em sítios (web), uma das principais funções de qualquer atividade online, indispensável para a gestão de negócios neste segmento.

O GA é gratuito e pode ser instalado em qualquer sítio, é uma ferramenta de gerenciamento de decisões na Web. As principais informações que ele fornece são:

- **Número total de visitantes** – visitante único e visitantes que retornam ao site
- **Comportamento dos visitantes** do site enquanto navegam nele
- **Origens de tráfego** – de onde estão vindo os acessos que chegam ao site
- **Metas** – quais metas traçadas foram atingidas
- **Taxa de conversão** das ações de marketing executadas

- **Marketing** – quais as campanhas de marketing que proporcionam melhor resultado
- **Mídias Sociais** – quais de suas ações em redes sociais lhe trazem maior acesso
- **Resultados de e-commerce** – como anda o desempenho da sua loja virtual

O objetivo principal do GA é informar o comportamento dos usuários ao navegar pelas diversas páginas e seções do site no qual foi instalado.

### 3.3.1 Terminologia para Monitor

Os termos relacionados a monitoramento e que são usados com frequência estão discutidos na Tabela 9.

Tabela 9 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 <i>overhead</i> (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 <i>burst</i> e sustentada. A taxa de modo <i>burst</i> 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

### 3.3.2 Classificação de Monitores

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.

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.

### 3.3.3 Monitores em Software

Monitores de software são usadas para monitorar os sistemas operacionais e software como redes e bancos de dados. 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.

### 3.3.4 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 10.



Tabela 10 Elementos dos monitores hardware de propósito geral disponíveis no mercado

<b>sondas</b>	sondas de alta impedância são utilizadas para observar os sinais nos pontos desejados no hardware do sistema
<b>contadores</b>	estes são incrementados sempre que um determinado evento ocorre
<b>elementos da lógica</b>	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
<b>comparadores</b>	estes podem ser usados para comparar valores de contadores ou sinal com valores pré-definidos
<b>mapeamento de hardware</b>	permite que histogramas de quantidades observadas possam ser calculados. É composto de vários comparadores e contadores
<b>timer</b>	usado para marcar tempo ou para desencadear uma operação de amostragem
<b>unidade de memória</b>	a maioria dos monitores têm unidades de memória para armazenar os dados

3.3.5 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 as 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.

A 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 depurado, os erros são raros. Com um monitor de hardware, 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 11.

Tabela 11 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 <sup>5</sup> por segundo é possível	taxa de amostragem limitada pelo processador e é necessário

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

3.3.6 *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.

### 3.4 Benchmarking

*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.

1. 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
2. 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
3. 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
4. Sequência de pacotes 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
5. A carga de trabalho pode ser arbitrariamente escolhida. Muitos dos programas não foram verificados para serem representativos das aplicações do mundo real
6. *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
7. *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 habilidade do tradutor do que no sistema em teste

3.4.1 Benchmarks Populares

Na área comercial, o termo *benchmark* é quase sempre usado como sinônimo de carga de trabalho. *Kernels*, programas sintéticos 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 por 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 12.

Tabela 12 Benchmarks populares

Whetstone
Este <i>kernel</i> avalia os recursos do processador, como matriz de endereçamento, aritmética de pontos fixo e flutuante. Os resultados dos <i>benchmarks</i> Whetstone são medidos em WIPS (Instruções Whetstone por Segundo). É considerado uma referência de ponto flutuante e é mais representativo de pequenas aplicações de engenharia/científicas que cabem na memória cache. Nele não há E/S.
Linpack
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. Os <i>benchmarks</i> Linpack são comparados com base na taxa de execução, medida em MFLOPS. É uma das referências mais utilizadas para medir performance de aplicações de engenharia/científicas.
Dhrystone
Contém muitas chamadas de procedimento, usado em 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. É uma medida popular de desempenho de aritmética de inteiros, mas não avalia processamento de ponto flutuante ou E/S.

3.4.2 Benchmark do SPEC

A Standard Performance Evaluation Corporation (SPEC) é uma corporação sem fins lucrativos formada pelos principais fornecedores de computador para desenvolver um conjunto padronizado de *benchmarks*.

A página [www.spec.org/cpu2017/Docs/overview.html#suites](http://www.spec.org/cpu2017/Docs/overview.html#suites) descreve a coleção de *benchmark* CPU2017 e a página [www.spec.org/cpu2017/results/](http://www.spec.org/cpu2017/results/) apresenta seus resultados. Nele pode ser encontrados *benchmark* para:

- Sistemas em nuvem
- CPU
- Plataformas gráficas
- Estações de trabalho
- Processamento paralelo
- Java Client/Server
- Servidores de mail
- Armazenamento
- Virtualização
- Servidores web

### 3.4.3 Benchmark de Roy Longbottom

A coleção de Benchmark de Roy Longbottom ([www.roylongbottom.org.uk](http://www.roylongbottom.org.uk)) inclui benchmark para arquivos, CPU, memória, disco (HD), periféricos, sistemas, gráficos, confiabilidade, dentre outros. Nele pode ser encontrados:

- um grande número de benchmarks gratuitos e programas de testes de confiabilidade;
- que são executados do Windows ou Linux, testando CPU, memória, barramentos, discos, redes, gráficos, telefones e tablets Android e Raspberry;
- empacotados em arquivos zip ou tar.gz que podem ser baixados e descomprimidos em suas próprias pastas. Nenhum processo de instalação é necessário
- com resultados são exibidos e salvos em arquivos textos;
- e tabelas de desempenho para vários Pcs, dos processadores 80486 até o Core i7

Exemplo 3.2 Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos

O Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC) é um evento anual e itinerante realizado pela Comissão Especial de Redes de Computadores e Sistemas Distribuídos, Redes de Computadores e Sistemas Distribuídos e pela Sociedade Brasileira de Computação (SBC), em parceria com o Laboratório de Redes de Computadores (LARC). Ao longo de mais de três décadas de realização, o SBRC firmou-se como o mais importante evento científico nacional em redes de computadores e sistemas distribuídos, e

um dos mais concorridos em informática. A importância do SBRC pode ser observada pelo número de submissões de trabalhos e de participantes nas suas diversas edições. O evento é composto de sessões técnicas, minicursos, painéis e debates, workshops, salão de ferramentas, palestras e tutoriais proferidos por convidados de renome internacional.

### 3.4.4 UserBenchmark

O UserBenchmark (<https://cpu.userbenchmark.com/>) atende exclusivamente aos usuários, desde 2011, e é mantido por uma equipe independente de cientistas e engenheiros. A equipe calcula e analisa benchmarks de modo a permitir aos seus usuários dimensionar seus computadores e explorar o desempenho do mundo real.

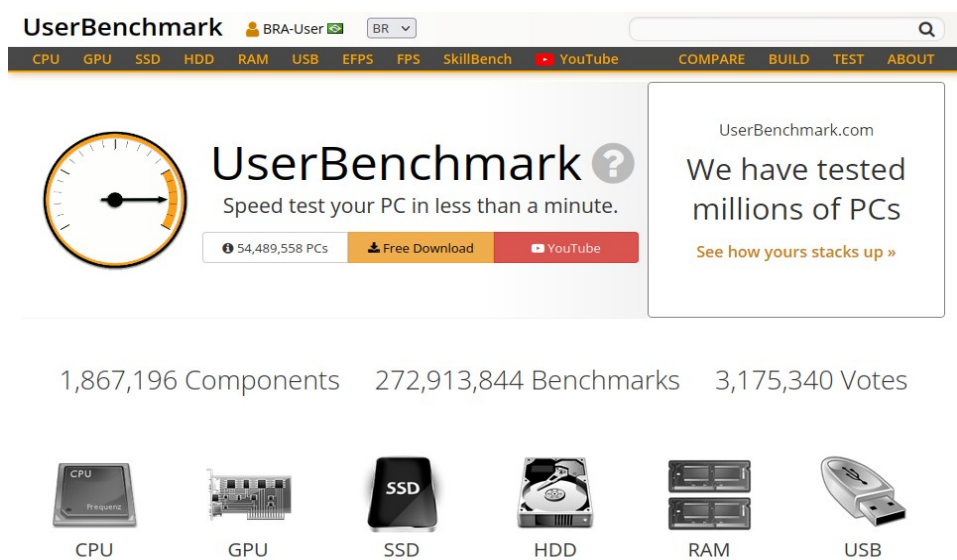


Figura 8 Imagem da parte superior do site do UserBenchmark ([cpu.userbenchmark.com](https://cpu.userbenchmark.com/)).

### 3.4.5 PassMark

A PassMark® Software Pty Ltd (<https://www.passmark.com/>) é um grupo privado de desenvolvimento de software com sede em Sydney, Austrália e uma filial na Califórnia, Estados Unidos. A empresa é especializada no desenvolvimento de soluções de benchmarking de desempenho de alta qualidade e mantêm o site [cpubenchmark.net](https://www.cpubenchmark.net/), com resultados de benchmark de CPU para mais de 1 milhão de sistemas, abrangendo mais de 3.500 tipos diferentes de CPUs.





Figura 9 Imagem da parte superior do site do PassMark (www.passmark.com).



## 4 Apresentação dos Resultados

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 experimental. 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.

## 4.2 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 10 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, por exemplo. Qual sistema é o preferido? A maioria das pessoas prefere o sistema com baixa variabilidade.

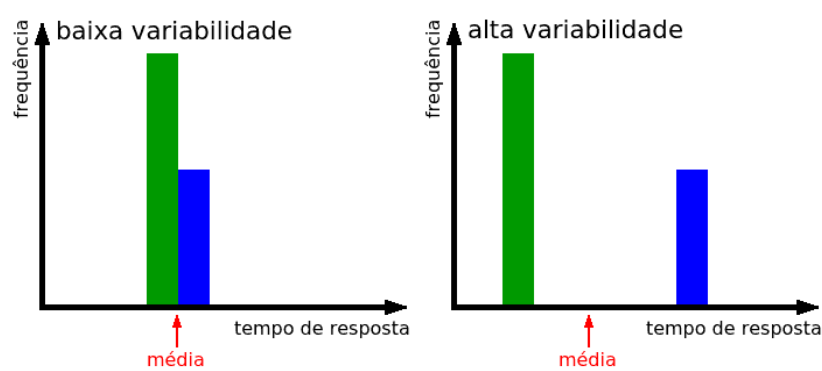


Figura 10 Histogramas de tempos de resposta de dois sistemas.

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 interquartil
- 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, o alcance 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 do alcance real.

A conclusão é que o intervalo é útil se, e somente se, existe uma razão para acreditar que a variável é limitada. O alcance dá a melhor estimativa desses limites.

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

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

$$x_m = \frac{1}{n} \sum_{i=1}^n x_i \tag{4.2}$$

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 significativamente 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 porcentagem), 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  $\alpha$ -quantil pode ser estimada pela classificação das observações e tendo a posição do  $[(n-1)\alpha + 1]$ -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,

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

4.3

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 interquartil 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 13 Sumarizando dados observados

Dados	$\{x_i\}_{i=1}^n$ amostra com n observações $x_i, i \in [1,n]$
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	
4. Mediana da amostra (ordenada)	$m_d = \begin{cases} x_{\frac{n-1}{2}} & \text{se } n \text{ ímpar} \\ \frac{1}{2} \left( x_{\frac{n}{2}} + x_{\frac{n+1}{2}} \right) & \text{se } n \text{ par} \end{cases}$
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	$COV = \frac{s}{m}$
10. Amplitude	diferença entre os valores máximo e mínimo
11. Percentis: p-percentil	$x_p = x_{[1+(n-1)p]}$ (amostra ordenada)
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  x_i - m $

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<sub>3</sub>-Q<sub>1</sub>)/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 14 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 14 Checklist para bons gráficos

<div>1. Os dois eixos de coordenadas são mostrados e rotulados?</div> <div>2. As escalas e divisões são mostradas em ambos os eixos?</div> <div>3. Há curva que pode ser removida sem reduzir as informações?</div> <div>4. Todos os símbolos no gráfico estão acompanhados de explicações textuais?</div> <div>5. As unidades de medida estão indicadas?</div> <div>6. A figura como um todo tornam as informações disponíveis para o leitor?</div> <div>7. Se o eixo vertical representa uma quantidade aleatória, são mostrados os intervalos de confiança?</div> <div>8. O título do gráfico é autoexplicativo e conciso?</div> <div>9. O gráfico claramente comunica a mensagem pretendida?</div> <div>10. A figura é referenciada e discutida no texto do relatório?</div>
--

## 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.

Embora as técnicas de regressão possam 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 Exemplo 4.3 mostra três exemplos de dados medidos e os ajustes 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. Assim, é óbvio que a qualidade do modelo deve ser medida pela distância entre os pontos observados e as linhas de modelo.

Os modelos de regressão tentam minimizar a distância medida na vertical entre o ponto de observação e a linha do modelo (ou curva). 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.

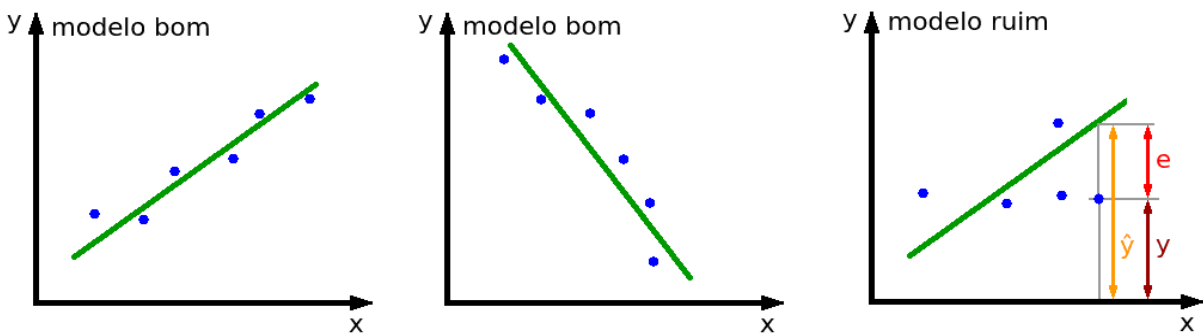


Figura 11 Histogramas de tempos de resposta de dois sistemas.

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. O Método dos Mínimos Quadrados e é

utilizado para obter o melhor modelo, utiliza o critério de escolher a linha que minimiza a soma dos quadrados dos erros.

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  $\{(x_1,y_1), \dots, (x_n,y_n)\}$ , 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 quadrado dos erros (SQE):

$$SQE = \sum_{i=1}^n e_i^2 = \sum_{i=1}^n (y_i - b_0 - b_1 x_i)^2$$

4.4

Sujeito a restrição de erro médio nulo, ou seja,  $\sum_{i=1}^n e_i = \sum_{i=1}^n (y_i - b_0 - b_1 x_i) = 0$  .

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:

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

4.5

$$b_0 = y_m - b_1 x_m$$

4.6

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,  $b_1 = (\sum x_i y_i - n x_m y_m) / (\sum x_i^2 - n x_m^2) = (3375 - 7 \times 38.71 \times 9.43) / (13855 - 7 \times 38.71^2) = 0.243$

$b_0 = y_m - b_1 x_m = 9.43 - 0.2438 \times 38.71 = -0.008$

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 12. 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.

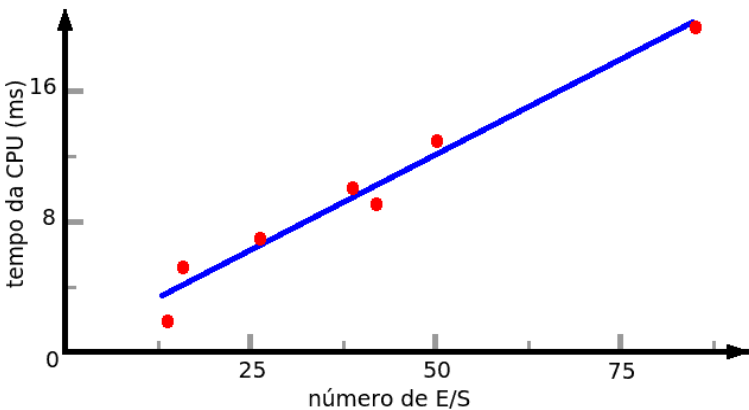


Figura 12 Dispersão de dados de tempo de E/S de disco e CPU.

Tabela 15 Cálculo dos 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	3,40430	-1,4043	1,9721
16	5	3,89180	1,1082	1,2281
27	7	6,57310	0,4269	0,1822
42	9	10,2295	-1,2295	1,5116
39	10	9,49820	0,5018	0,2518
50	13	12,1795	0,8205	0,6732
83	20	20,2235	-0,2235	0,0500
$\sum=271$	66	66,0000	0,0000	5,8690

Na Tabela 15 estão listados o tempo de CPU previsto pelo modelo, os valores medidos, erros e quadrado dos erros para cada um das sete observações. O SQE é 5,869. Este é o SQE mínimo possível. Quaisquer outros valores de  $b_0$  e  $b_1$  daria um maior SQE.

### 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 boa 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 SQR (Soma de Quadrado da Regressão) é inferior ou igual à SQE.

### 4.4.3 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:

$$(y_i - y_m) = (y_i - y_m + \hat{y}_i - \hat{y}_i) = (\hat{y}_i - y_m) + (y_i - \hat{y}_i) \tag{4.7}$$

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

$$\sum_{i=1}^n (y_i - y_m)^2 = \sum_{i=1}^n (\hat{y}_i - y_m)^2 + \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{4.8}$$

em que:

$$\begin{aligned} SQT &= \sum_{i=1}^n (y_i - y_m)^2 \text{ - Soma de Quadrado Total} \\ SQR &= \sum_{i=1}^n (\hat{y}_i - y_m)^2 \text{ - Soma de Quadrado da Regressão} \\ SQE &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 \text{ - Soma de Quadrado dos Erros (dos Resíduos)} \end{aligned}$$

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.

4.4.4 Graus de Liberdade

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

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 obtenção 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:

$$SQT = SQR + SQE$$
$$n - 1 = 1 + n - 2$$

4.9

4.10

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.

4.4.5 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 16 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		

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

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  $\chi^2$ . Assim, SQT, SQR e SQE tem distribuição  $\chi^2$  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  $\alpha$  se a razão é maior do que o  $(1-\alpha)$ -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 SQR é significativamente mais elevado do que SQE pelo cálculo da razão entre  $(SQR/gl_R) / (SQE/gl_E)$ , onde  $gl_R$  e  $gl_E$  são graus de liberdade para o SQR e SQE, respectivamente. A quantidade  $SQR/gl_R$  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 razão 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, conseqüentemente, 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 17 Cálculo dos erros dos dados de tempo de E/S 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
$\Sigma=271$	66	66,0000	0,0000	5,8690

n = 7

$y_m = 66/7 = 9,4$

Tabela 18 Tabela ANOVA para a Regressão Linear Simples

Fonte	gl	SQ	QM	F	F <sub>tabelado</sub>
Regressão (R)	1	SQR = 199,22	QMR = 199,22	$F = 199,22/1,14 = 174,8$	$F(0,95; 1; 5) = 5,99$
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	$(20,2-9,4)^2=116,6$	$(20,2-20)^2=0,0$
$\Sigma$	271	66	66	199,22	5,7
$n = 7 \quad y_m = 66/7 = 9,4$					

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.

Teorema do Limite Central: Se  $y_1, y_2, \dots, y_n$  é uma sequência de  $n$  variáveis aleatórias independentes e igualmente distribuídas com  $E(y_i) = \mu$  e  $V(y_i) = \sigma^2$  (ambos finitos),  $x = y_1 + y_2 + \dots + y_n$  e  $z_n = \frac{x - n\mu}{\sqrt{n\sigma^2}}$ , então  $\lim_{n \rightarrow \infty} z_n$  é a distribuição normal padrão.

Em teoria das probabilidades, uma das consequências imediatas do Teorema do Limite Central é que quando o tamanho da amostra aumenta, a distribuição amostral da sua média aproxima-se cada vez mais de uma distribuição normal, ou seja, evento distante da média tem menor probabilidade de ocorrer; aproximar da média aumenta a probabilidade de ocorrência.

## 5 Teoria das Filas

Segundo Fogliatti (2007), a Teoria das Filas consiste na modelagem analítica de sistemas que resultam em espera (formação de filas) e tem como objetivo determinar e avaliar quantidades<sup>2</sup>, denominadas medidas de desempenho, que expressam a produtividade desses sistemas. Ela tem como principal objetivo encontrar um ponto de equilíbrio que satisfaça seus usuários e que seja economicamente viável para o provedor; além de ser capaz de fazer previsões, tais como, dimensionamento, infraestrutura e quantidade suficiente de equipamentos para satisfazer os usuários (COSTA, 2006); constitui uma ferramenta para avaliação e de planejamento, no contexto do mercado e do negócio. A Figura 13 mostra a relação entre os atores envolvidos em um negócio na qual o ajuste entre a satisfação do usuário e o lucro do provedor envolve a Teoria das Filas. Mostra ainda a capacidade deste modelo como ferramenta de planejamento.

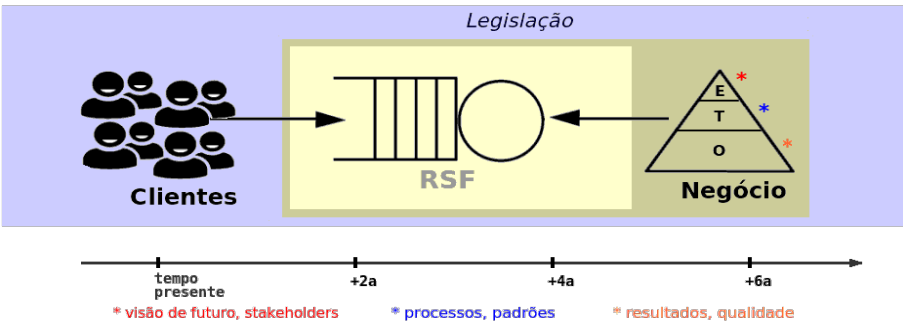


Figura 13 Relação entre os atores envolvidos em um negócio, e a capacidade de avaliação e de planejamento dos modelo baseados em SF.

A unidade básica da Teoria das Fila é o Sistema de Fila Simples (SF), também denominado Centro de Serviço (CS). Os SFs incluem os recursos do sistema, um ou mais servidores e um conjunto de usuários que demandam seus. Os elementos que definem um SF é apresentado a seguir:

2 Estas quantidades são as propriedades mensuráveis das filas são suas quantidades operacionais.

- **Servidor** – unidade que processa os serviços demandados pelos seus usuários (Figura 18 na cor verde)
- **Fila** – representa os usuários que esperam pelo serviço, incluem os que estão sendo atendidos pelos servidores (Figura 18 na cor vermelha)
- **Fila de Espera** – representa os usuários que aguardam pelo serviço (Figura 18 na cor cinza)

Adicionalmente, deve-se incluir os usuários, o provedor e o serviço:

- **Serviço** – é a unidade de trabalho do SF, é o processo executado pelo Servidor.
- **Provedor** – é o responsável pelo SF, sua configuração, operação e manutenção; estabelece as condições operacionais do lado provedor; gerencia a relação usuário/provedor.
- **Usuários** – demandantes do serviço ofertado pelo SF; estabelece as condições operacionais do lado dos usuários; gerencia a relação provedor/usuário.

A Figura 13 ilustra didaticamente um modelo de um SF, com uma única Fila e um único Servidor, nesta figura são destacadas a Fila (em vermelho), a Fila de Espera (retângulos cinza) e o Servidor (em verde) e os pacotes (retângulos cinzas); as setas indicam o fluxo dos pacotes no sistema.

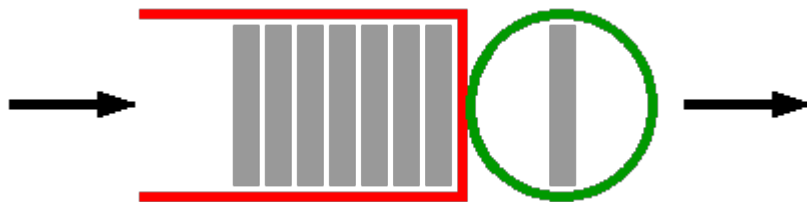


Figura 14 Modelo de um Sistema de Fila Simples, com uma (única) Fila e um (único) Servidor e a Fila de Espera, as setas indicam o fluxo dos pacotes no sistema.

## 5.1 Sistema de Filas

Uma fila ocorre sempre que a demanda pelo serviço é maior que a capacidade do sistema de prover este serviço.



Um sistema de filas pode ser definido como pacotes chegando, esperando pelo serviço (se não forem atendidos imediatamente) e saindo do sistema após terem sido atendidos. O termo pacote (p) em teoria das filas, é um termo genérico, aplicando-se não somente a seres humanos. O conceito pode abranger, por exemplo, processo esperando para ser processado por uma CPU, mensagens que chegam em um roteador e pessoas esperando no caixa do supermercado.

Rede Aberta de Filas em que todas as estações de serviço possuem servidores com capacidade fixa de processamento (CF) ou servidores que não apresentam espera em filas (infinitos servidores – IS).

## 5.2 Rede de Sistemas de Filas

Duas ou mais SFs podem ser interligadas para forma uma Rede de Sistemas de Fila Simples (RSF), permitindo que inúmeros sistemas computacionais possam ser modelados por RSFs. Em geral, RSF são modelos em que pacotes chegam, são processados pelos SFs que o constitui, e saem da rede.

Ao contrário das SF simples, 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 15. Esta figura é uma representação esquemática simplificada na qual dois Discos e uma CPU são interconectados para formar uma RSF de três SFs. Note que esta RSF é capaz de modelar servidores web de pequeno porte.

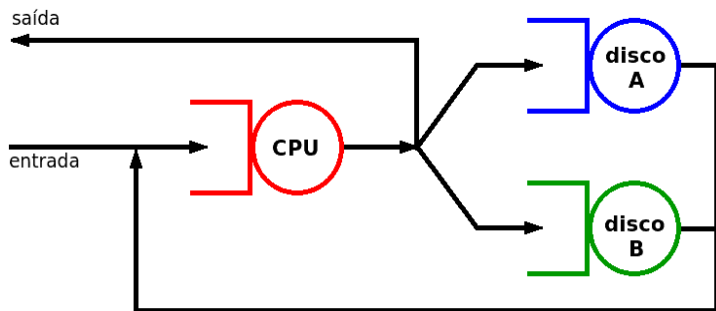


Figura 15 Rede de Sistemas de Filas abertas com pacotes entrando e saindo da rede.

Uma RSF fechada não tem chegadas externas ou partidas, o número total de pacotes na rede é constante. Pode-se representar uma RSF fechada como uma rede em que a saída está

ligada com a entrada e, desta forma, os pacotes saem e voltam imediatamente a ela, como mostrado na Figura 16, a RSF mantém a circulação dos pacotes nos seus Sfs.

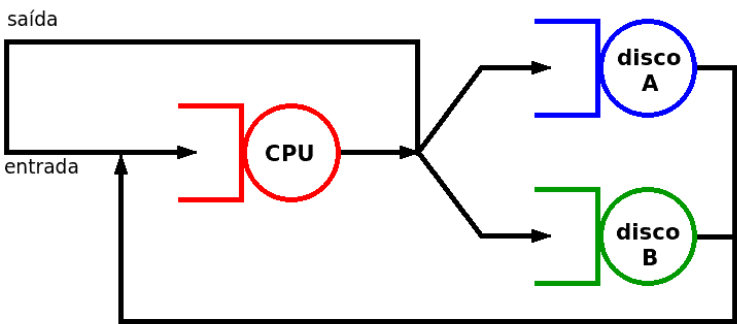


Figura 16 Rede de Sistemas de Filas fechadas em que o fluxo de pacotes é um ciclo fechado.

Também é possível ter RSF mistas que estão abertas para alguns pacotes e fechada para outros. A Figura 17 mostra um exemplo de uma RSF mista com duas classes de pacotes: lote e interativas. O sistema é fechado para pacotes interativos e aberto para pacotes em lote. Todas os pacotes de uma mesma classe têm as mesmas demandas de serviço e iguais probabilidades de transição. Os pacotes de uma classe são indistinguíveis. A análise de uma RSF mista é feita por classe de pacote.

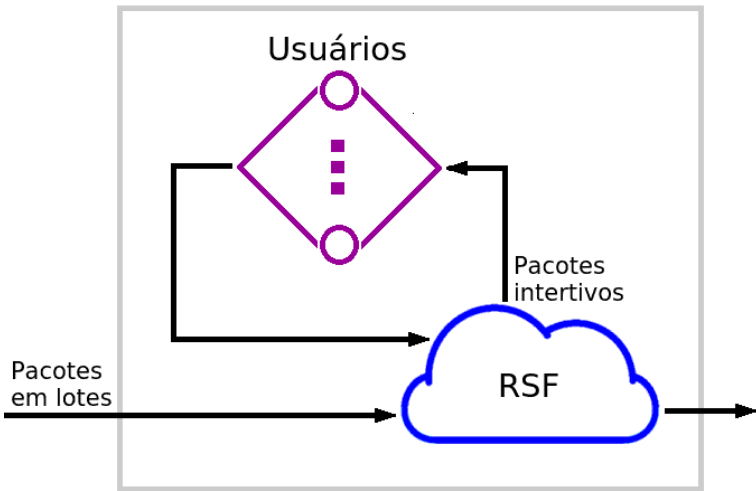

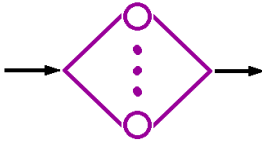


Figura 17 RSF mista, fechada para pacotes interativos e aberta para pacotes em lote.

Na modelagem de sistemas computacionais são encontrados três tipos de SF, descritos na Tabela 19.

Tabela 19 Os três tipos de SF encontrados na modelagem de sistemas computacionais	
Centro de Serviço de Capacidade Fixa (CS)	
São dispositivos cuja taxa de serviço não depende do número de pacotes no dispositivo. Por exemplo, a CPU de um sistema pode ser modelado como um centro de serviço de capacidade fixa. Dispositivos que possuem memória (buffer) auxiliar.	

Centro de Carga (CC)	
Os dispositivos cujas taxas de serviço são dependentes da carga ou do número de pacotes no dispositivo. Conexões em geral, com banda de passagem. A ligação entre dois nós de uma rede de computadores é um exemplo de um centro de serviços dependentes de carga.	
Centro de Atraso (CA)	
São dispositivos que não têm Filas e os pacotes demandam a mesma quantidade de tempo nos mesmos, independentemente do número de pacotes no Servidor. Tais dispositivos são também chamados Infinity Server (IS). Um grupo de terminais dedicados geralmente é modelado como 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.

### 5.3 Aplicações

Existem diversas aplicações da teoria das filas, que podem ser encontradas na literatura de probabilidade, dentre elas destacam-se:

- **Fluxo de tráfego** como no roteamento de pacotes em redes digitais
- **Escalonamento de tarefas** como nos sistemas opracionais
- **Prestação de serviços** como nas redes digitais e na Internet

Scherr, citado por Jain (1991), 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. Este modelo é constituído por n usuários (terminais) com um ou mais servidores. Os usuários geram pedidos (pacotes) que são atendidos pelo servidor, análogo ao reparador, Figura 18.a.

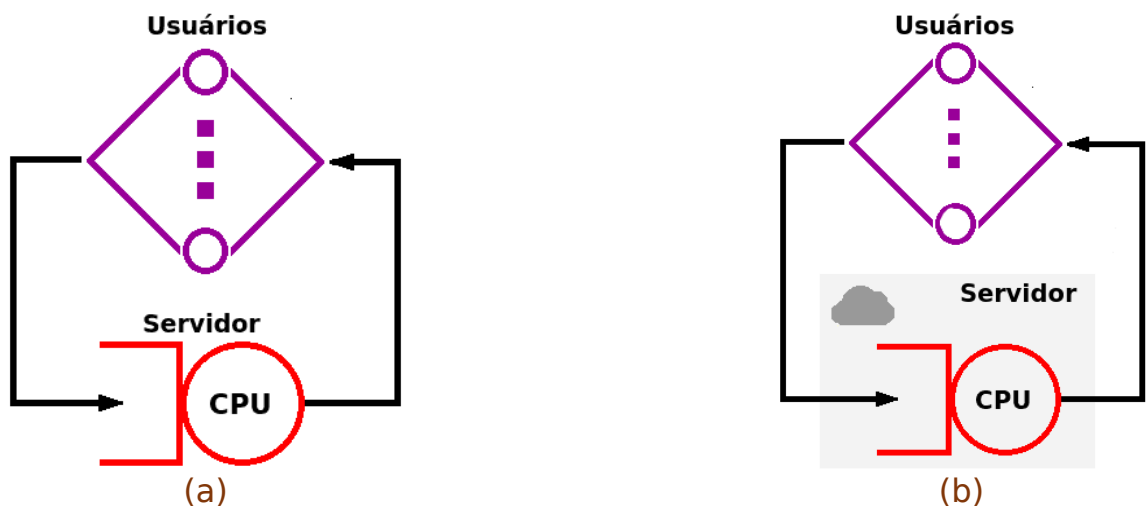


Figura 18 a) modelo de máquina reparadora como analogia para sistema computacional em nuvem; e b) servidor central como modelo de sistema de tempo compartilhado.

O modelo de servidor central foi introduzido por Buzen em 1973 (Jain, 1991) e pode ser representado também pela Figura 18.a. A CPU neste modelo é o Servidor Central que é conectado a outros dispositivos. Após o serviço dos dispositivos de E/S serem executados, os pacotes retornam para a CPU para processamento adicional e deixa-o quando a próxima E/S é requerida ou quando o pacote estiver concluída.

Estes modelos são análogos a atual computação em nuvem (cloud computing) Figura 18.b. Leia o verbete “computação em nuvem” na Wikipédia.

## 6 Leis Operacionais

A Leis Operacionais, não realmente leis e relações matemáticas simples que não necessitam de nenhuma hipótese sobre as distribuições dos tempos de serviço ou dos intervalos entre chegadas. Elas foram identificadas inicialmente por Buzen em 1976 e posteriormente estendidas por Denning e Buzen em 1978 (Jain, 1991).

A Leis Operacionais se aplicam a sistemas que constituídos por múltiplos Sistema de Filas (SF), ou seja, às Redes de Sistemas de Fila (RSF), elas fornecem soluções simples com enfoque em aplicações em sistemas reais.

### 6.1 Representando RSF

As duas maneiras úteis de representar RSFs são as visões externa e a interna. Na representação externa, a RSF é vista como um sistema compacto que recebe, processa e despacha pacotes. Em um dado tempo de observação (T), o que se pode observar neste sistema é a número de pacotes que chegam (A), o número de pacotes que saem (C) e o tempo que ele ficou ocupado (B), Figura 19. Na representação interna de uma RSF, os Sistemas de Fila (SF) que o constitui não são vistos.

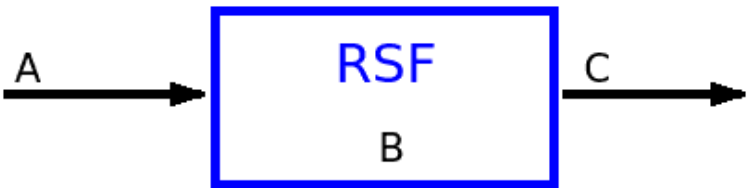


Figura 19 Representação de uma RSF por meio da visão externa.

A outra maneira, a visão externa, mostrada na Figura 20, os Sistemas de Fila (SF) que constitui a RSF são vistos. Nesta figura, a RSF possui 4 SFs, em destaque. Os pacotes que chegam na RSF são processados por (visitam) cada uma de suas SF e, ao final, saem do sistema.

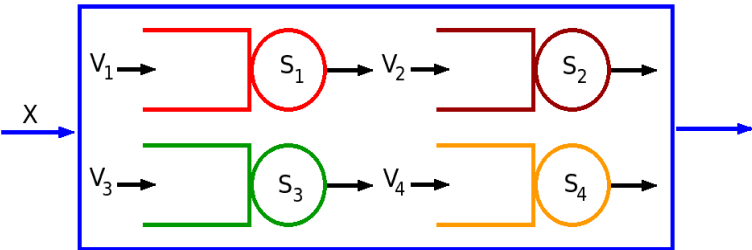


Figura 20 Representação de uma RSF e seus 4 SFs por meio da visão interna.

Uma RSF é definida pela sua taxa de chegada (X) e pelos seus SFs. Seja M o número de SFs de uma RSF, então cada SF é definido pelo número de visitas (S) que os pacotes fazem a ele e o tempo requerido (S) para processar cada pacote. Em síntese, uma RSF é definida por:

- **X** - taxa externa de chegada na RSF
- **M** - número de SFs na RSF
- **V<sub>i</sub>** - número de visitas ao SF<sub>i</sub>, 1 ≤ i ≤ M
- **S<sub>i</sub>** - tempo de serviço do SF<sub>i</sub>, 1 ≤ i ≤ M

Para fins de Engenharia, os resultados requeridos são (1 ≤ i ≤ M):

- **Q<sub>i</sub>** - número médio de pacotes no SF<sub>i</sub>
- **R<sub>i</sub>** - tempo de resposta do SF<sub>i</sub>
- **U<sub>i</sub>** - utilização do SF<sub>i</sub>
- **R** - tempo de resposta da RSF

## 6.2 Organização Interna das RSF

A aplicação das Leis não requer o conhecimento de como as SFs se organizam dentro da RSF, para fins didáticos pode-se pensar em uma associação em série destas SFs, como mostrado na Figura 21.

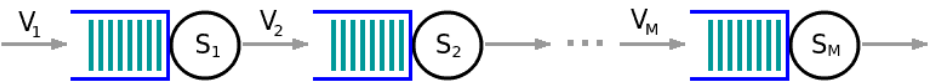


Figura 21 Associação em série de M SFs para formar uma RSF.

A associação em série, como mostrado na Figura 21, é a maneira mais simples de representara as SFs de uma RSF e suas quantidades operacionais (V e S). O leitor deve ter em mente que a RSF real pode ter qualquer outra configuração, mas isso não importa para a aplicação das Leis Operacionais.

### 6.3 Condições de Denning e Buzen

A aplicação das Leis Operacionais requer apenas as chamadas Condições de Denning e Buzen:

- 1. **Fluxo de Pacotes Balanceado (FPB)** - para cada SF, o número de chegadas é igual ao número de partidas.
- 2. **Eventos Únicos** - não podem existir eventos múltiplos simultâneos.
- 3. **Homogeneidade de Dispositivos** - a taxa de serviço para uma particular classe não depende do estado do sistema a não ser do número total de pacotes e do número de pacotes de sua classe no sistema

### 6.4 Quantidades Operacionais<sup>3</sup>

As quantidades operacionais são aquelas que podem ser medidas (diretamente) durante um período de tempo finito de observação. As grandezas que podem ser diretamente medidas (observadas) em uma RSF são o número de pacotes que chegam ( $A_i$ ), número de pacotes concluídos ( $C_i$ ), o tempo ocupado ( $B_i$ ) e o período de observação ( $T$ ) que é definido pelo observador. A Tabela 20 relaciona as quantidades operacionais das RSF, sua unidade, descrição e relação matemática.

Tabela 20 Quantidades Operacionais das RSF, unidade, descrição e relação matemática

Quantidade	Unidade	Descrição	Relação
------------	---------	-----------	---------

<sup>3</sup> A palavra operacional significa que a quantidade é observável e portanto pode ser medida, não é uma quantidade estimada por alguma expressão matemática derivada do sistema.

<b>T</b>	[s]	Duração da observação	
<b>A<sub>i</sub></b>	[p]	Número de chegadas ( <i>arrivals</i> )	
<b>C<sub>i</sub></b>	[p]	Número de términos ( <i>completions</i> )	
<b>B<sub>i</sub></b>	[s]	Tempo ocupado ( <i>busy time</i> )	
<b>λ<sub>i</sub></b>	[p/s]	Taxa de chegada	$\lambda_i = \frac{A_i}{T}$
<b>X<sub>i</sub></b>	[p/s]	Vazão ( <i>throughput</i> )	$X_i = \frac{C_i}{T}$
<b>U<sub>i</sub></b>	[ ]	Utilização	$U_i = \frac{B_i}{T}$
<b>S<sub>i</sub></b>	[s/p]	Tempo médio de serviço	$S_i = \frac{B_i}{C_i}$

Essas quantidades são variáveis que podem mudar de um período de observação para outro. Entretanto, existem relações envolvendo estas quantidades, aqui denominadas Leis Operacionais, que se mantém em várias observações, definindo as propriedades das RSF e permitindo avaliar seu desempenho.

Sistema Estáveis

A abordagem matemática de RSF pela Teoria das Filas exige que exista estabilidade no fluxo de chegada e no processo de atendimento, ou seja, os valores de λ e μ se mantêm constantes no tempo. Sistemas que funcionam ininterruptamente tem-se, geralmente, uma situação estável.

Outra exigência para que o processo seja estável é que os Servidores sejam capazes de atender ao fluxo de chegada. No caso de um Sistema de Fila Simples (Fila+Servidor), tem-se que μ > λ (a capacidade de atendimento é maior que o ritmo de chegada) e, caso isto não ocorra, resulta que o tamanho da fila aumentará indefinidamente.

Em sistemas estáveis, todas as características aleatórias das filas se mantêm estáveis o tempo todo, significando que oscilam em torno de um valor médio como, por exemplo, o tamanho médio, o tempo médio de espera, o tempo médio de atendimento

6.5 Lei da Utilização

Dado um número de partidas C<sub>i</sub>, um tempo de ocupação B<sub>i</sub>, de um sistema de fila SF<sub>i</sub> durante um intervalo de observação T, a Equação 6.2 é válida e é denominada **Lei da Utilização**.



$U_i = \frac{B_i}{T} = \frac{C_i}{T} \times \frac{B_i}{C_i}$	6.1
$U_i = X_i S_i \text{ (Lei da Utilização)}$	6.2

Exemplo 6.1 Considere um roteador em que pacotes chegam a uma taxa de 600 pps e o roteador leva em média 1 ms para encaminhá-los. Qual a utilização do sistema?

O Centro de Serviço é um roteador, aqui denotado por r.  
 $X_r$  = taxa de saída do roteador  $r$  = taxa de chegada do roteador  $r = 600$  p/s  
 $S_r$  = tempo médio de serviço do roteador  $r = 0,001$  s/p  
 $U_r = X_r S_r = 600 \times 0,001 = 0,60$   
A utilização do sistema é igual a 60%, este resultado é válido para qualquer processo (distribuição de probabilidade) de chegada ou de atendimento.

## 6.6 Lei do Fluxo

A Lei do Fluxo correlaciona a vazão global da RSF com as vazões de seus SFs. Em uma rede aberta de filas, o número de pacotes partindo da RSF na unidade de tempo define a sua vazão. Numa RSF fechada, a taxa com que se cicla no sistema define a sua vazão. Se num dado período de observação T, o número de pacotes que entram é igual ao número de pacotes que saem da RSF, expresso pela Equação 6.3, pode-se afirmar que esta RSF satisfaz a hipótese de fluxo de pacotes balanceado. Se o intervalo de observação é grande,  $C_i$  tende a se aproximar de  $A_i$ .

$A_i = C_i$	6.3
-------------	-----

Supondo que no tempo T:

- cada pacote faça  $V_i$  visitas ao  $SF_i$
- que  $C_i$  pacotes entram ou saem do  $SF_i$
- que  $C_0$  pacotes entram ou saem da RSF

Se o fluxo de pacotes da RSF é balanceado então  $C_0$ ,  $C_i$  e  $V_i$  estão relacionados pela Equação 6.4.

$C_i=C_0 V_i \text{ ou } V_i=\frac{C_i}{C_0}$	6.4
---	-----

Em que  $V_i$  representa a taxa de visitas ao  $SF_i$ ;  $C_0$  é número de pacotes que entra/saem da RSF (fluxo balanceado); e  $C_i$  é número de pacotes que entra/saem do  $SF_i$  (fluxo balanceado). A Figura 22 ilustra a aplicação da Equação 6.4 nos  $SF_i$  de uma RSF com fluxo de pacotes balanceado.

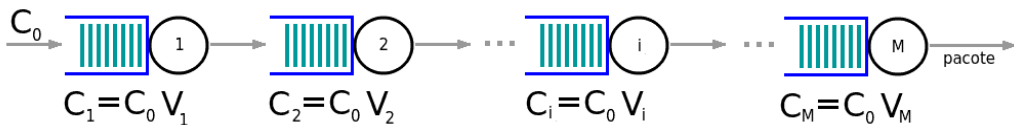


Figura 22 Relação entre  $C_0$ ,  $C_i$  e  $V_i$  dos  $SF_i$  de uma RSF com fluxo de pacotes balanceado.

A vazão global ( $X$ ) da RSF durante este período de observação é dada pela Equação 6.5, já a vazão do  $SF_i$  é dada pela Equação 6.6, denominada **Lei do Fluxo**.

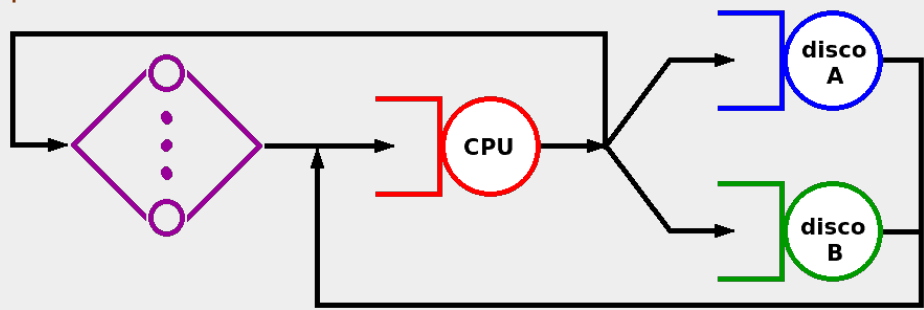
$X=\frac{C_0}{T}$	6.5
$X_i=\frac{C_i}{T} \rightarrow X_i=\frac{C_i}{T} \times \frac{C_0}{C_0} \rightarrow X_i=\frac{C_i}{C_0} \times \frac{C_0}{T} \rightarrow X_i=V_i X$	6.6
$X_i=X V_i$ <b>Lei do Fluxo</b>	6.7

Combinando a **Lei da Utilização** (Equação 6.2) com a **Lei do Fluxo** (Equação 6.6) obtém-se as Equações 6.8 e 6.9.

$U_i=X_i S_i \rightarrow U_i=X V_i S_i \rightarrow U_i=X D_i$	6.8
$D_i=V_i S_i$	6.9

Em que  $D_i$ , Equação 6.9, é chamado de **Demanda Total sobre  $SF_i$** . O  $SF_i$  que possuir o maior  $D_i$  será o **gargalo da RSF**.

Exemplo 6.2 Em um sistema de tempo compartilhado, figura abaixo, 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 E/S no Disco A e 100 pedidos E/S 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 E/S por segundo. Qual é o rendimento do sistema e a utilizações dos dispositivos.



- Dados:
- $D_{CPU} = 5\text{ s}$
  - $V_A = 80$
  - $V_B = 100$
  - $Z = 18\text{ s}$
  - $S_A = 0,050\text{ s}$
  - $S_B = 0,030\text{ s}$
  - $N = 17$
  - $X_A = 15,70\text{ p/s}$

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,

$$\begin{aligned} D_{CPU} &= 5\text{ s} \\ D_A &= S_A V_A = 0,050 \times 80 = 4\text{ s} \\ D_B &= S_B V_B = 0,030 \times 100 = 3\text{ s} \end{aligned}$$

Usando a Lei de Fluxo Forçado, as vazões são:

$$\begin{aligned} X &= X_A/V_A = 15,75/80 = 0,1963\text{ p/} \\ X_{CPU} &= X V_{CPU} = 0,1963 \times 181 = 35,48\text{ p/s} \\ X_B &= X V_B = 0,1963 \times 100 = 19,6\text{ p/s} \end{aligned}$$

Usando a Lei de Utilização, as utilizações dos dispositivos são:

$$\begin{aligned} U_{CPU} &= X D_{CPU} = 0,1963 \times 5 = 98,0\% \\ U_A &= X D_A = 0,1963 \times 4 = 78,4\% \\ U_B &= X D_B = 0,1963 \times 3 = 58,8\% \end{aligned}$$

## 6.7 Lei de Little

Desenvolvida por John Little no início dos anos 60, a Lei de Little relaciona o número de pacotes em um SF ( $Q_i$ ) com o tempo médio despendido nele ( $R_i$ ):

Número médio de pacotes = Taxa de chegada x Tempo médio de resposta (Equação 6.10)

$Q_i = \lambda_i R_i$	6.10
-----------------------	------

A Lei de Little se aplica sempre que o número de chegadas é igual ao número de saídas (RSF com fluxo de pacotes balanceado). Em RSF com fluxo de pacotes balanceado, tem-se que  $\lambda_i = X_i$  e a Equação 6.10 equivale à Equação 6.11, onde  $X_i$  é a vazão da  $SF_i$ .

$Q_i = X_i R_i$	6.11
-----------------	------

A Lei de Little tem outras expressões matemáticas, a da Equação 6.10 se aplica a este contexto.

Exemplo 6.3 O comprimento médio da fila no sistema de computador do Exemplo 6.2 foram observadas como sendo 8,88, 3,19 e 1,40 pacotes na CPU, no Disco A e no Disco B, respectivamente. Quais foram os tempos de resposta destes dispositivos?

No Exemplo 6.2, as vazões dos dispositivos foram determinados como sendo:

$X_{CPU} = 35,4 \text{ p/s}$   
 $X_A = 15,7 \text{ p/s}$   
 $X_B = 19,6 \text{ p/s}$

A nova informação dada neste exemplo é:

$Q_{CPU} = 8,8 \text{ p}$   
 $Q_A = 3,1 \text{ p}$   
 $Q_B = 1,40 \text{ p}$

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

$R_{CPU} = Q_{CPU}/X = 8,88/35,48 = 0,250 \text{ s}$ $R_A = Q_A /X_A = 3,19/15,70 = 0,203 \text{ s}$ $R_B = Q_B /X_B = 1,40/19,6 = 0,071 \text{ s}$
---

Exemplo 6.4 Um servidor de arquivos foi monitorado durante 30 minutos e o número observado de operações de I/O foi 10.800 e que o número médio de pedidos ativos foram três. Qual o tempo médio de resposta por pedido no servidor?

O Centro de Serviço é um servidor, aqui denotado por r.

T = período de observação = 30 m = 1800 s  
 $\lambda_r$  = taxa de chegada no servidor r = 600 p/s  
 $R_r$  = tempo médio de resposta do servidor r = 0,001 s  
 $Q_r = \lambda_r R_r = 600 \times 0,001 = 0,60$

O tamanho da fila do sistema é igual a1 pacote.

## 6.8 Lei do Tempo de Resposta

Em uma RSF, usuários enviam pacotes que são processados por ela e os resultados são devolvidos aos usuários. A **Lei de Little** pode ser aplicada para qualquer SF<sub>i</sub> de uma RSF, desde que ela possua fluxo de pacotes balanceado. A Equação 6.12 é obtida aplicando-se a Lei de Little a RSF.

$Q = X R$	6.12
-----------	------

Conhecendo-se o número de pacotes em cada um dos SF<sub>i</sub> da RSF, obtém-se a Equação 6.13 aplicando a Lei de Little aos seus M SF.

$Q=Q_1+Q_2+\cdots+Q_M=\sum_{i=1}^M Q_i$	6.13
---	------

Substituindo as Equações 6.12 e 6.11 na Equação 6.13, obtém-se a Equação 6.14.

$X R= X_1 R_1+X_2 R_2+X_3 R_3+\cdots+X_M R_M$	6.14
---	------

Aplicando a **Lei do Fluxo** (Equação 6.6) a Equação 6.14, obtém-se a Equação 6.15 que, simplificada é a Equação 6.16, a expressão matemática da **Lei do Tempo de Resposta**.

$X R= X V_1 R_1+X V_2 R_2+X V_3 R_3+\cdots+X V_M R_M$	6.15
$R=V_1 R_1+V_2 R_2+V_3 R_3+\cdots+V_M R_M \rightarrow R=\sum_{i=1}^M V_i R_i \text{ (Lei do Tempo de Resposta)}$	6.16

Exemplo 6.5 Para o sistema de tempo compartilhado do Exemplo 6.2, pode-se calcular o tempo de resposta, utilizando a Lei do Tempo de Resposta Iterativa como se segue:

Dos dados do exemplo anterior, tem-se que:

$$\begin{aligned} V_A &= 80 \\ V_B &= 100 \\ V_{CPU} &= 181 \\ R_{CPU} &= 0,250 \text{ s} \\ R_A &= 0,203 \text{ s} \\ R_B &= 0,071 \text{ s} \end{aligned}$$

$$R = V_{CPU} R_{CPU} + V_A R_A + V_B R_B = 181 \times 0,250 + 80 \times 0,203 + 100 \times 0,071 = 68,59 \text{ s}$$

## 6.9 Análise de Gargalo

Em uma RSF, o **SF gargalo** é aquele que possui a maior demanda de serviço  $D$  e, conseqüentemente, a maior utilização  $U$ . Se o SF gargalo é a  $SF_b$  então  $D_b = D_{max}$ , em que  $D_{max}$  é o valor máximo de  $\{D_1, D_2, ..., D_M\}$ .

A vazão e o tempo de resposta da RSFo é limitado pelo SF gargalo, e seus valores obtidos resolvendo as Inequações 6.17 e 6.18, respectivamente. Estas inequações são denominadas limites assintóticos de  $X$  e  $R$ .

$X(N) \leq \min\left(\frac{1}{D_{max}}, \frac{N}{D+Z}\right) \text{ (limite assintótico de X)}$	6.17
$R(N) \geq \max\left(D, N D_{max} - Z\right) \text{ (limite assintótico de R)}$	6.18

Em que  $D = \sum_{i=1}^M D_i$ , é a soma da demanda de serviço de todos os  $SF_i$ , exceto os terminais dos usuários.

O ponto de interseção das duas retas limites é chamado “joelho” do sistema e é dado pela Equação 6.19.

$N^* = \frac{D+Z}{D_{max}}$	6.19
-----------------------------	------

Em que,  $N^*$  é o número de usuários no “joelho” do sistema. Se o número de usuários no sistema for maior que  $N^*$ , pode-se concluir há formação de Fila ( $Q > 0$ ) em algum SF da RSF.

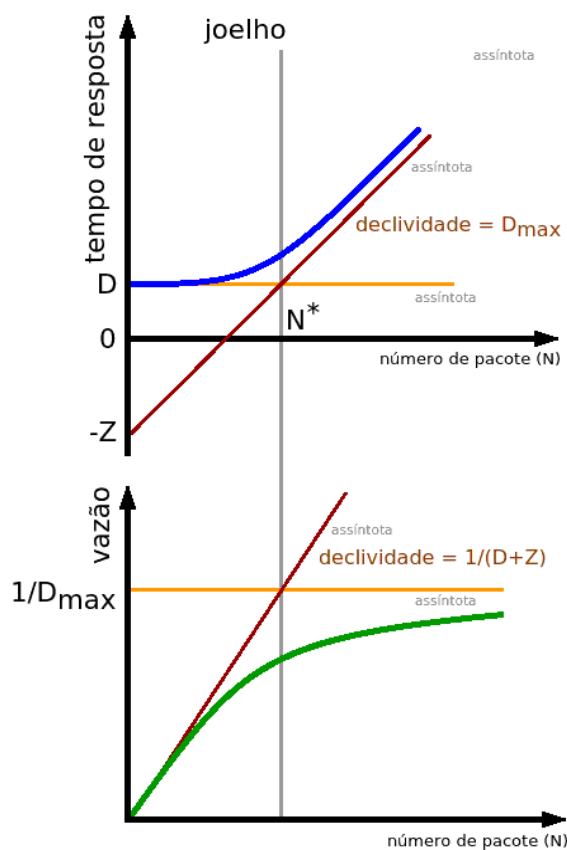


Figura 23 Limites assintóticos do tempo de resposta (curva azul) e da vazão (curva verde); e o joelho da RSF.

Na Figura 23 pode-se observar a variação do tempo de resposta (curva azul) e da vazão (curva verde) em função de  $N$  (número de pacotes na RS). A medida que  $N$  aumenta, o tempo de resposta aumenta proporcionalmente a  $D_{\max}$ . Com o aumento de a vazão tende ao valor de  $1/D_{\max}$ . O valor de  $N^*$ , o joelho da RSF, é definido a partir de duas assíntotas de cada gráfico.

Exemplo 6.6 Quantos usuários podem ser suportados pelo sistema de tempo compartilhado do Exemplo 6.2, se o tempo de resposta for mantida abaixo de 100 s?

Usando os limites assintóticos sobre o tempo de resposta, obtém-se:  
 $R(N) \geq \max \{12, 5N-18\}$   
O tempo de resposta será maior do que 100 se  $5N-18 \geq 100 \rightarrow N \geq 23,6$ .

**Conclusão:** o sistema pode suportar até 23 usuários com o tempo de resposta inferior a 100 s.

### 6.10 Lei do Tempo de Resposta Interativo

Em uma RSF interativa, usuários enviam pacotes que são processados por ela e os resultados são devolvidos de volta aos usuários. Entre cada pedido de um usuário, há um tempo ocioso  $Z$ .



Se o tempo de resposta da RSF interativa é R, então o tempo entre a entra e a saída de um pacote nesta rede é (R+Z). Logo, cada usuário envia/recebe  $\frac{T}{(R+Z)}$  pacotes da RSF interativa em um intervalo de tempo T.

Se a RSF interativa possuir N usuários, então sua vazão X é dada pela Equação 6.21, a **Lei do Tempo de Resposta Interativo**.

$X = \frac{\text{numero de pacotes na RSF}}{\text{tempo total}} = \frac{\text{numero de usuarios} \times \text{numero de pacotes por usuario}}{\text{tempo total}}$	6.20
$X = \frac{N \left[ \frac{T}{(R+Z)} \right]}{T} \rightarrow R = \frac{N}{X} - Z \text{ (Lei do Tempo de Resposta Interativo)}$	6.21

Desta forma, completa-se a discussão das Leis Operacionais, um resumo destas leis é apresentado na Tabela 21.

Tabela 21 Resumo das Leis Operacionais e suas variáveis

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 do Tempo de Resposta	$R = \sum_{i=1}^M R_i V_i$	
Lei do Tempo de Resposta Interativo	$R = \frac{N}{X} - Z$	
Limites Assintóticos	$X(N) \leq \min \left\{ \frac{1}{D_{max}}, \frac{N}{D+Z} \right\}$	
	$R(N) \geq \max \left\{ D, \frac{N}{D_{max}} - Z \right\}$	
Variáveis		
D	soma das demandas de serviço nos SFs, $\sum D_i$	[s/p]
$D_i$	demanda total de serviço por pacotes do $SF_i$	[s/p]
$D_{max}$	demanda de serviço no SF gargalo, $\max \{ D_i \}$	[s/p]
M	número de SFs da RSF	[]
N	número de pacotes na RSF	[p]
$Q_i$	número de pacotes no $SF_i$	[p]
R	tempo de resposta da RSF	[s]
$R_i$	tempo de resposta por pacotes do $SF_i$	[s]

$S_i$	tempo de serviço por pacotes do $SF_i$	[s/p]
$U_i$	utilização do $SF_i$	[ ]
$V_i$	visitas ou número de pacotes executadas pelo $SF_i$	[ ]
$X$	vazão ou rendimento da RSF	[p/s]
$X_i$	vazão ou rendimento do $SF_i$	[p/s]
$Z$	tempo aleatório ou latência (hardware)	[s]

## 7 Análise de Valor Médio (AVM)

Modelos Abertos de Filas são usados para representar sistemas de processamento, tais como Software e Hardware. 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 uma taxa média de chegada  $\lambda$ .

Assume-se que todos os dispositivos do sistema 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 CS com capacidade fixa (SF) em uma rede de filas aberta, o tempo de resposta é dada pela Equação 7.1.

$R_i = S_i(1 + Q_i)$	7.1
----------------------	-----

Uma maneira fácil de entender essa relação é a de considerar um pacote marcado que flui através do sistema. Ao chegar em um SF, este pacote vê outros  $Q_i$  pacotes à frente (incluindo a que está no Servidor) e espera o tempo  $Q_i \times S_i$ . Incluindo o tempo para ele mesmo, este pacote deverá esperar  $S_i(1 + Q_i)$ .

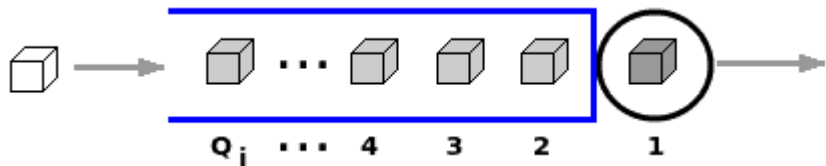


Figura 24 Sequência de  $Q_i$  pacotes vista pelo pacote que chega em um SF.

A Equação 7.1 não é uma Lei Operacional. Ela assume que o serviço é um Processo de Poisson (sem memória), uma suposição que não é testável operacionalmente. Para calcular o

tempo de resposta do pacote marcado sem a suposição “sem memória” é preciso saber quanto tempo o pacote que está no Servidor já usou.

A equação anterior, combinado com 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 Pacotes, temos que o rendimento do sistema é igual à taxa de chegada, expresso pela Equação 7.2. A taxa de transferência do SF<sub>i</sub>, utilizando a Lei do Fluxo Forçado é dada pela Equação 7.3. A utilização do SF<sub>i</sub>, utilizando a Lei de Utilização é expressa pela Equação 7.4. O tamanho da fila do SF<sub>i</sub>, usando a lei de Little, é calculado pela Equação 7.5.

$X=\lambda$	7.2
$X_i=XV_i$	7.3
$U_i=X_iS_i=XV_iS_i=XD_i$	7.4
$Q_i=X_iR_i=X_iS_i(1+Q_i)=U_i(1+Q_i)\rightarrow Q_i=\frac{U_i}{(1-U_i)}$	7.5

Substituindo Q<sub>i</sub>, da Equação 7.5, na Equação 7.1, o tempo de resposta do SF<sub>i</sub> é dado pela Equação 7.6.

$Q_i=\frac{U_i}{1-U_i}\rightarrow R_i=S_i(1+Q_i)\rightarrow R_i=\frac{S_i}{1-U_i}$	7.6
--	-----

Em IS 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 pacotes recebidas pelo Servidor, pois não há espera. Assim, o tempo de resposta e o comprimento de fila para IS são calculados pelas Equações 7.7 e 7.8, respectivamente.

$R_i=S_i$	7.7
$Q_i=R_iX_i=S_iXV_i=XD_i=U_i$	7.8

Note-se que a utilização do IS representa a média do número de pacotes 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 22.

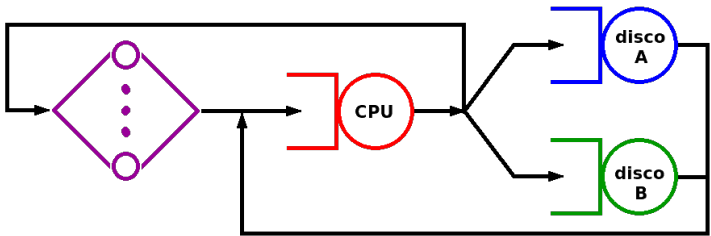
Tabela 22 Quadro da análise de RSF abertas

Entradas		
X	taxa de chegada externa, Vazão do sistema	[p/s]

$S_i$	tempo de serviço por visita ao $SF_i$	[s/p]
$V_i$	número de visitas ao $SF_i$	[]
$M$	número de SF e IS(não incluindo terminais)	[]
Saídas		
$Q_i$	média do número de pacotes no $SF_i$	[p]
$R_i$	tempo de resposta do $SF_i$	[s]
$U_i$	utilização do $SF_i$	[]
$R$	tempo de resposta do sistema	[s]
$N$	número médio de pacotes no sistema	[p]
demanda de serviços do $SF_i$	$D_i=S_i V_i$	[s/p]
utilização do $SF_i$	$U_i= X D_i$	[]
vazão do $SF_i$	$X_i= X V_i$	[p/s]
tempo de resposta do $SF_i$	$R_i=\frac{S_i}{(1-U_i)}$ (CS) $R_i=S_i$ (IS)	[s]
comprimento da fila do $SF_i$	$Q_i=\frac{U_i}{(1-U_i)}$ (CS) $Q_i=U_i$ (IS)	[p]
tempo de resposta do sistema	$R=\sum_{i=1}^M R_i V_i$	[s]
número de pacotes no sistema	$N=\sum_{i=1}^M Q_i$	[p]

Exemplo 7.1 A figura abaixo 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 s
Número de pacotes	= 10.800
Tempo de CPU ocupada	= 1.728 s
Tempo de Disco A ocupado	= 1.512 s
Tempo de Disco B ocupado	= 2.592 s
Número de visitas ao Disco A	= 75.600
Número de visitas ao Disco B	= 86.400



Vazão da RSF:  $X = 10.800/3.600 = 3$  pacotes por segundo

Visitas aos SFs:

$V_A = 75.600/10.800 = 7$  visitas por pacote ao disco A

$V_B = 86.400/10.800 = 8$  visitas por pacote ao disco B

$V_{CPU} = 1 + 7 + 8 = 16$  visitas por pacote a CPU

Demandas dos SFs por pacote:

$D_{CPU} = 1.728/10.800 = 0,16 \text{ s de CPU por pacote}$   
 $D_A = 1.512/10.800 = 0,14 \text{ s do Disco A por pacote}$   
 $D_B = 2.592/10.800 = 0,24 \text{ s do Disco B por pacote}$

Taxa de serviço dos SFs:

$S_{CPU} = 0,16/16 = 0,01 \text{ s}$   
 $S_A = 0,14/7 = 0,02 \text{ s}$   
 $S_B = 0,24/8 = 0,03 \text{ s}$

Utilização dos Sfs:

$U_{CPU} = X D_{CPU} = 3 \times 0,16 = 0,48$   
 $U_A = X D_A = 3 \times 0,14 = 0,42$   
 $U_B = X D_B = 3 \times 0,24 = 0,72$

Os tempos de resposta dos SFs:

$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,01/(1-0,48) = 0,0192 \text{ s}$   
 $R_A = S_A/(1-U_A) = 0,02/(1-0,42) = 0,0345 \text{ s}$   
 $R_B = S_B/(1-U_B) = 0,03/(1-0,72) = 0,107 \text{ s}$

Tempo de resposta do servidor  $R = \sum R_i V_i = 16 \times 0,0192 + 7 \times 0,0345 + 8 \times 0,107$ , ou seja,  
 $R = 1,406 \text{ s}.$

O modelo pode ser usado para responder a algumas das perguntas típicas. Por exemplo, podemos querer quantificar o impacto das seguintes alterações:

1. Aumentar o número de clientes para 8

Assumindo que o pacote médio possuam as mesmas características, a taxa de chegada de aumenta por um fator de 8/6.

$X = 3 (8/6) = 4 \text{ pacotes/s}$

$U_{CPU} = X D_{CPU} = 4 \times 0,16 = 0,64$   
 $U_A = X D_A = 4 \times 0,14 = 0,56$   
 $U_B = X D_B = 4 \times 0,24 = 0,96$

$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,01 / (1 - 0,64) = 0,0278 \text{ s}$   
 $R_A = S_A/(1-U_A) = 0,02 / (1 - 0,56) = 0,0455 \text{ s}$   
 $R_B = S_B/(1-U_B) = 0,03 / (1 - 0,96) = 0,75 \text{ s}$

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

**Conclusão:** se o número de pacotes aumenta de 6 para 8, o tempo de resposta do servidor degradar-se pelo fator  $6,76/1,406 = 4,8$

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 E/S) de 10%

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

$$V_B = 0,5 \times 8 = 4 \text{ (cache com 50\% de acerto)}$$
$$S_{CPU} = 1,3 \times 0,01 = 0,013 \rightarrow D_{CPU} = 0,208 \text{ s (sobrecarga de 30\%)}$$
$$S_B = 1,1 \times 0,03 = 0,033 \rightarrow D_B = 4 \times 0,033 = 0,132 \text{ s (aumento de 10\%)}$$

A análise do sistema modificado é a seguinte:

$$U_{CPU} = X D_{CPU} = 3 \times 0,208 = 0,624$$
$$U_A = X D_A = 3 \times 0,14 = 0,42$$
$$U_B = X D_B = 3 \times 0,132 = 0,396$$

$$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,013/(1-0,624) = 0,0346 \text{ s}$$
$$R_A = S_A/(1-U_A) = 0,02/(1-0,42) = 0,0345 \text{ s}$$
$$R_B = S_B/(1-U_B) = 0,033/(1-0,396) = 0,0546 \text{ s}$$

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

**Conclusão:** ao usar um cache para o Disco B, o tempo de resposta do servidor melhorar em  $(1,406-1,013)/1,406 = 28\%$

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

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

$$V_B = 0$$
$$V_A = 7 + 8 = 15$$

$$D_{CPU} = 0,16 \text{ s (como antes)}$$
$$D_A = 15 \times 0,02 = 0,3 \text{ s}$$

$$U_{CPU} = X D_{CPU} = 3 \times 0,16 = 0,48$$
$$U_A = X D_A = 3 \times 0,3 = 0,90$$

$$R_{CPU} = S_{CPU}/(1-U_{CPU}) = 0,01/(1-0,48) = 0,0192 \text{ s}$$
$$R_A = S_A/(1-U_A) = 0,02/(1-0,90) = 0,2 \text{ s}$$

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

**Conclusão:** se substituir os dois discos por um lado, o tempo de resposta do servidor degrada-se pelo fator  $3,31/1,406 = 2,35$

## 7.1 Algoritmo da Análise do Valor Médio

A Análise do Valor Médio - Mean Value Analysis- (AVM) é uma técnica que estende os resultados das Leis Operacionais para a análise de RSF. A AVM, como o nome indica, calcula o desempenho médio, ela não permite calcular variância. 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 pacotes, Reiser e Lavenberg mostraram em 1980, que o tempo de resposta do dispositivo de ordem i (SF<sub>i</sub>) é dado pela Equação 7.9.

$R_i(N)=S_i[1+Q_i(N-1)]$	7.9
--------------------------	-----

Aqui, Q<sub>i</sub>(N-1) é o comprimento médio da fila do SF<sub>i</sub>, com N-1 pacotes na RSF. A partir de N = 0 o desempenho para qualquer número de pacotes pode ser calculado de forma iterativa, como mostramos a seguir. Tendo em conta os tempos de resposta dos SF individuais, o tempo de resposta da RSF, utilizando a Lei do Tempo de Resposta Geral é dado pela Equação 7.10. A vazão da RSF, usando a Lei do Tempo de Resposta Iterativo, é dado pela Equação 7.11. As vazões dos SF medido, em termos de pacotes por segundo, são calculados pela Equação 7.12. Os comprimentos das filas nos SF são expressos, pela Lei de Little, pela Equação 7.13. Estas equações assumem que todos os SF são centros de capacidade fixa enfileirados.

$R(N)=\sum_{i=1}^M V_i R_i(N)$	7.10
$X(N)=\frac{N}{R(N)+Z}$	7.11
$X_i(N)=X(N) V_i$	7.12
$Q_i(N)=X_i(N) R_i(N)=X(N) V_i R_i(N)$	7.13

Se um SF é um centro de atraso (IS), 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 pacotes que o servidor recebe. Assim, a equação de tempo de resposta para os centros de atraso é simplesmente Equação 7.14.

$R_i(N)=S_i$	7.14
--------------	------

As Equações 7.11 e 7.12 para a vazão de dispositivos e comprimentos de fila se aplicam a centros de atraso. As Equações 7.11, 7.12, 7.13 e o 7.14 definem uma iteração do AVM. O procedimento é inicializado para N = 0 usuários e Q<sub>i</sub>(0) = 0. O procedimento completo é descrito na Tabela 23.



Tabela 23 Algoritmo do AVM

Entradas		
N	número de pacotes	[p]
Z	tempo aleatório ou latência (hardware)	[s]
M	número de SF (não incluindo terminais)	[ ]
$S_i$	tempo de serviço por visita ao $SF_i$	[s/p]
$V_i$	número de visitas ao $SF_i$	[ ]
Saídas		
X	rendimento da RSF	[p/s]
$Q_i$	número médio de pacotes no $SF_i$	[p]
$R_i$	tempo de resposta do $SF_i$	[s]
$U_i$	utilização do $SF_i$	[ ]
R	tempo de resposta da RSF	[s]
Valores iniciais	para $i = 1$ até $M$ faça início $Q_i = 0$ fim	
Iteração	para $n = 1$ até $N$ faça início $R = 0$ para $i = 1$ até $M$ faça início $R_i = S_i(1 + Q_i)$ ou $R_i = S_i$ $R += R_i V_i$ fim $X = n/(R+Z)$ para $i = 1$ até $M$ faça início $X_i = X V_i$ $Q_i = X V_i R_i$ $U_i = X V_i S_i$ fim fim	
	fim	

O Exemplo 7.2 é uma aplicação da AVM usando a rede de filas do Exemplo 7.1.

Exemplo 7.2 Considere o modelo de rede de filas do Exemplo 7.1, cada usuário faz 10 pedidos do E/S ao Disco A e cinco pedidos E/S 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:

$S_A = 0,3, V_A = 10 \rightarrow D_A = 3 \text{ p/s}$   
 $S_B = 0,2, V_B = 5 \rightarrow D_B = 1 \text{ p/s}$   
 $D_{CPU} = 2, V_{CPU} = V_A + V_B + 1 = 16 \rightarrow S_{CPU} = 0,125 \text{ pacote/s}$   
 $Z = 4 \text{ s}$   
 $N = 20$

Inicialização
Pacote N = 0 Comprimentos da Fila nos SFs: $Q_{CPU} = 0, Q_A = 0, Q_B = 0$
Iteração 1
Pacote N = 1  Tempos de resposta do SF:  $R_{CPU} = S_{CPU} (1+Q_{CPU}) = 0,125 (1+0) = 0,125 \text{ s}$ $R_A = S_A (1+Q_A) = 0,3(1+ 0) = 0,3 \text{ s}$ $R_B = S_B (1+Q_B) = 0,2 (1+ 0) = 0,2 \text{ s}$  Tempo de resposta da RSF:  $R = R_{CPU} V_{CPU}+R_A V_A+R_B V_B = 0,125\times16+0,3\times10+0,2\times5 = 6 \text{ s}$  Vazão da RSF:  $X = N/(R+Z) = 1/(6+4) = 0,1 \text{ pacote/s}$  Comprimentos de fila do dispositivo:  $Q_{CPU} = X R_{CPU} V_{CPU} = 0,1 \times 0,125 \times 16 = 0,2$ $Q_A = X R_A V_A = 0,1 \times 0,3 \times 10 = 0,3$ $Q_B = X R_B V_B = 0,1 \times 0,2 \times 5 = 0,1$
Iteração 2
Número de usuários: N = 2  Tempos de resposta do SF:  $R_{CPU} = S_{CPU} (1+Q_{CPU}) = 0,125 (1 + 0,2) = 0,15 \text{ s}$ $R_A = S_A (1 + Q_A) = 0,3 (1 + 0,3) = 0,39 \text{ s}$ $R_B = S_B (1 + Q_B) = 0,2 (1 + 0,1) = 0,22 \text{ s}$  Tempo de resposta da RSF:  $R = R_{CPU} V_{CPU}+R_A V_A+R_B V_B = 0,15\times16+0,39\times10+0,22\times5 = 7,4 \text{ s}$  Vazão da RSF:  $X = N/(R+Z) = 2/(7,4+4) = 0,175 \text{ pacote/s}$  Comprimentos de fila do SF:  $Q_{CPU} = X R_{CPU} V_{CPU} = 0,175 \times 0,15 \times 16 = 0,421$ $Q_A = X R_A V_A = 0,175 \times 0,39 \times 10 = 0,684$ $Q_B = X R_B V_B = 0,175 \times 0,22 \times 5 = 0,193$

As iterações podem continuar para valores mais elevados de N. Eles podem ser implementados usando qualquer pacote de software de planilha eletrônica. Os tempos de resposta, vazão e comprimentos de fila e utilização no final destas iterações estão listados na Tabela 24.

Tabela 24 Resultados da MVA para o Exemplo 3.9

N (p)	Tempo de Resposta (s)				Vazão do Sistema (p/s)	Comprimento da Fila (p)			Utilização		
	CPU	Disco A	Disco B	Sistema		CPU	Disco A	Disco B	CPU	Disco A	Disco B
1	0,0100	0,0100	0,0100	0,050	0,247	0,007	0,002	0,002	0,007	0,002	0,002
1	0,0100	0,0300	0,0400	0,690	0,213	0,034	0,045	0,068	0,033	0,043	0,064
2	0,0103	0,0313	0,0427	0,727	0,423	0,070	0,093	0,145	0,065	0,085	0,126
3	0,0107	0,0328	0,0458	0,767	0,629	0,108	0,144	0,231	0,097	0,126	0,187
4	0,0111	0,0343	0,0492	0,811	0,831	0,147	0,200	0,327	0,128	0,167	0,247
5	0,0115	0,0360	0,0531	0,860	1,030	0,189	0,259	0,437	0,159	0,206	0,304
6	0,0119	0,0378	0,0575	0,914	1,220	0,232	0,323	0,561	0,188	0,244	0,360
7	0,0123	0,0397	0,0625	0,975	1,410	0,277	0,391	0,703	0,217	0,281	0,413
8	0,0128	0,0417	0,0681	1,040	1,590	0,324	0,463	0,865	0,245	0,317	0,464
9	0,0132	0,0439	0,0746	1,120	1,760	0,373	0,541	1,050	0,272	0,351	0,512
10	0,0137	0,0462	0,0820	1,200	1,920	0,422	0,622	1,260	0,297	0,384	0,558
11	0,0142	0,0487	0,0905	1,290	2,080	0,473	0,708	1,500	0,321	0,415	0,601
12	0,0147	0,0512	0,1000	1,400	2,220	0,524	0,798	1,780	0,344	0,444	0,641
13	0,0152	0,0539	0,1110	1,510	2,360	0,575	0,890	2,100	0,365	0,471	0,677
14	0,0158	0,0567	0,1240	1,640	2,480	0,625	0,985	2,460	0,385	0,496	0,711
15	0,0163	0,0596	0,1380	1,780	2,590	0,674	1,080	2,870	0,403	0,519	0,742
16	0,0167	0,0624	0,1550	1,940	2,690	0,721	1,180	3,340	0,419	0,541	0,769
17	0,0172	0,0653	0,1730	2,120	2,780	0,765	1,270	3,850	0,433	0,559	0,794
18	0,0176	0,0681	0,1940	2,310	2,850	0,805	1,360	4,430	0,446	0,576	0,816
19	0,0181	0,0708	0,2170	2,520	2,910	0,842	1,440	5,060	0,457	0,591	0,835
20	0,0184	0,0733	0,2420	2,750	2,960	0,873	1,520	5,750	0,466	0,603	0,852
21	0,0187	0,0756	0,2700	2,990	3,000	0,901	1,590	6,490	0,474	0,614	0,866
22	0,019	0,0777	0,3000	3,240	3,040	0,923	1,650	7,280	0,480	0,623	0,879
23	0,0192	0,0796	0,3310	3,510	3,060	0,942	1,70	8,110	0,485	0,63	0,890
24	0,0194	0,0811	0,3640	3,790	3,080	0,957	1,750	8,980	0,489	0,636	0,900
25	0,0196	0,0825	0,3990	4,080	3,090	0,968	1,790	9,870	0,492	0,641	0,908
26	0,0197	0,0836	0,4350	4,380	3,10	0,977	1,820	10,800	0,494	0,645	0,915
27	0,0198	0,0845	0,4720	4,680	3,110	0,984	1,840	11,700	0,496	0,648	0,922
28	0,0198	0,0852	0,5100	4,990	3,110	0,989	1,860	12,700	0,497	0,65	0,927
29	0,0199	0,0857	0,5480	5,300	3,120	0,992	1,870	13,700	0,498	0,652	0,932
30	0,0199	0,0861	0,5870	5,610	3,120	0,995	1,880	14,600	0,499	0,653	0,936

Note que o AVM 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 Pacote. Além disso, a análise, como apresentado aqui, assume que todos os SF ou são centros de capacidade fixa de serviços ou centros de atraso. Em ambos os casos, assumimos os tempos de serviço é exponencialmente distribuído. O Algoritmo 7.1 implementa o algoritmo da AVM em C++, conforme descrito na Tabela 23.

Algoritmo 7.1 Software AVM em C++

//-----  
#include <string>  
#include <vector>

```
#include <algorithm>
#include <fstream>
#include <sstream>
//-----
using namespace std;
//-----
void Salvar( string fn, string str ){
    ofstream fs;
    fs.open(fn);
    replace(str.begin(), str.end(),'.',' ');
    fs << str;
    fs.close();
}
string AVM( const int M, const int N, double *S, int *V, double Z ){
    double R[M], Q[M], U[M], X[M], D[M], Rs, Xs;
    stringstream htm;
    htm << fixed;
    htm.precision(2);
    htm << "Simulacao AVM<hr>"
        << "<table border='1'><tr><th>n<th>R<th>X";
    for( int m = 0; m < M; m++ )
        htm << "<th style='background-color:LightGray'>R<sub>" << m << "</sub>"
            << "<th>Q<sub>" << m << "</sub>"
            << "<th>U<sub>" << m << "</sub>"
            << "<th>X<sub>" << m << "</sub>";
    for( int m = 0; m < M; m++ ) { D[m] = S[m]*V[m]; Q[m] = 0; }
    for( int n = 0; n < N; n++ ){
        Rs = 0.0;
        for( int m = 0; m < M; m++ ){
            R[m] = S[m] * (1.0 + Q[m]);
            Rs += R[m] * V[m];
        }
        Xs = n/(Rs+Z);
        for( int m = 0; m < M; m++ ){
            X[m] = Xs * V[m];
            Q[m] = Xs * V[m] * R[m];
            U[m] = Xs * S[m] * V[m];
        }
        htm << "<tr><td>" << n << "<td>" << Rs << "<td>" << Xs;
        for( int m = 0; m < M; m++ )
            htm << "<td style='background-color:Gainsboro'>" << R[m]
                << "<td>" << Q[m]
                << "<td>" << U[m]
                << "<td>" << X[m];
    }
    htm << "</table><br><br><br>";
    return htm.str();
}
int main( void ){
    const int M    = 3,
            N      = 40;
    int      V[M] = { 1, 1, V[0]+ V[1] + 1 };
    double   S[M] = { 0.03, 0.04, 0.01 },
            Z     = 4.0;
    Salvar("AVM-out.html",AVM(M,N,S,V,Z));
    return 0;
}
```

A Tabela 25 lista alguns resultados do Algoritmo 7.1.

Tabela 25 Resultados da MVA para o Exemplo 3.9

n	R	X	R <sub>0</sub>	Q <sub>0</sub>	U <sub>0</sub>	X <sub>0</sub>	R <sub>1</sub>	Q <sub>1</sub>	U <sub>1</sub>	X <sub>1</sub>	R <sub>2</sub>	Q <sub>2</sub>	U <sub>2</sub>	X <sub>2</sub>
(p)	(s)	(p/s)	(s)	(p)		(p/s)	(s)	(p)		(p/s)	(s)	(p)		

0	0,16	0,00	0,03	0,00	0,00	0,00	0,04	0,00	0,00	0,00	0,03	0,00	0,00	0,00
1	0,16	0,24	0,03	0,01	0,01	0,24	0,04	0,01	0,01	0,24	0,03	0,02	0,02	0,72
2	0,16	0,48	0,03	0,01	0,01	0,48	0,04	0,02	0,02	0,48	0,03	0,04	0,04	1,44
3	0,17	0,72	0,03	0,02	0,02	0,72	0,04	0,03	0,03	0,72	0,03	0,07	0,06	2,16
4	0,17	0,96	0,03	0,03	0,03	0,96	0,04	0,04	0,04	0,96	0,03	0,09	0,09	2,88
5	0,17	1,20	0,03	0,04	0,04	1,20	0,04	0,05	0,05	1,20	0,03	0,12	0,11	3,60
6	0,17	1,44	0,03	0,04	0,04	1,44	0,04	0,06	0,06	1,44	0,03	0,14	0,13	4,31
7	0,18	1,68	0,03	0,05	0,05	1,68	0,04	0,07	0,07	1,68	0,03	0,17	0,15	5,03
8	0,18	1,91	0,03	0,06	0,06	1,91	0,04	0,08	0,08	1,91	0,04	0,20	0,17	5,74
9	0,18	2,15	0,03	0,07	0,06	2,15	0,04	0,09	0,09	2,15	0,04	0,23	0,19	6,45
10	0,19	2,39	0,03	0,08	0,07	2,39	0,04	0,10	0,10	2,39	0,04	0,26	0,21	7,17
11	0,19	2,63	0,03	0,08	0,08	2,63	0,04	0,12	0,11	2,63	0,04	0,30	0,24	7,88
12	0,19	2,86	0,03	0,09	0,09	2,86	0,04	0,13	0,11	2,86	0,04	0,33	0,26	8,58
13	0,20	3,10	0,03	0,10	0,09	3,10	0,05	0,14	0,12	3,10	0,04	0,37	0,28	9,29
14	0,20	3,33	0,03	0,11	0,10	3,33	0,05	0,15	0,13	3,33	0,04	0,41	0,30	9,99
15	0,21	3,57	0,03	0,12	0,11	3,57	0,05	0,16	0,14	3,57	0,04	0,45	0,32	10,70
16	0,21	3,80	0,03	0,13	0,11	3,80	0,05	0,18	0,15	3,80	0,04	0,50	0,34	11,40
17	0,22	4,03	0,03	0,14	0,12	4,03	0,05	0,19	0,16	4,03	0,04	0,54	0,36	12,10
18	0,22	4,26	0,03	0,15	0,13	4,26	0,05	0,20	0,17	4,26	0,05	0,59	0,38	12,79
19	0,23	4,50	0,03	0,15	0,13	4,50	0,05	0,22	0,18	4,50	0,05	0,64	0,40	13,49
20	0,23	4,73	0,03	0,16	0,14	4,73	0,05	0,23	0,19	4,73	0,05	0,70	0,43	14,18
21	0,24	4,96	0,03	0,17	0,15	4,96	0,05	0,24	0,20	4,96	0,05	0,76	0,45	14,87
22	0,24	5,18	0,04	0,18	0,16	5,18	0,05	0,26	0,21	5,18	0,05	0,82	0,47	15,55

7.2 Análise do Valor Médio Aproximada

O algoritmo de valor médio é iterativo e para se calcular o desempenho da RSF com N pacotes deve-se conhecer o desempenho do sistema com N-1 pacotes. Para simplificar este processo e evitar ter que fazer os cálculos sequencialmente, utiliza-se uma aproximação que baseia-se no fato de que se aumentarmos o número de pacotes na RSF, o seu número aumentará proporcionalmente em cada SF, ou seja, a relação  $Q_i(N)/N$  tende a ser uma constante, Equações 7.15 e 7.16, o que permite escrever a Equação 7.17.

$\frac{Q_i(N)}{N}=a_i \text{ constante para } \forall N$	7.15
$\frac{Q_i(N-1)}{N-1}=\frac{Q_i(N)}{N}=\frac{Q_i(N+1)}{N+1}=\dots$	7.16
$Q_i(N-1)=\frac{N-1}{N}Q_i(N)$	7.17

Substituindo o valor de  $Q_i(N-1)$  da Equação 7.17 na Equação 7.9, obtém-se a Equação 7.18, que é equação que dá origem ao AVM Aproximado. As Equações 7.18 a 7.22 é o modelo matemático do AVM aproximado.

$R_i(N)=S_i\left[1+\frac{N-1}{N}Q_i(N)\right]$	7.18
$R_i(N)=S_i \text{ (IS)}$	7.19
$R(N)=\sum_{i=1}^M V_i R_i(N)$	7.20
$X(N)=\frac{N}{Z+R(N)}$	7.21
$Q_i(N)=X(N)V_iR_i(N)$	7.22

Observe que  $Q_i(N)$  é recalculado ao final de cada iteração. Quando a diferença entre  $Q_i(N)$  no começo e no final da iteração for menor que um certo  $\epsilon$ , o algoritmo deve parar. O valor inicial de  $Q_i(N)$  não influencia o valor final, ele interfere somente no número de iterações do algoritmo. Na Tabela 26 está escrito o algoritmo da AVM Aproximada (Algoritmo de Schweitzer).

Tabela 26 Algoritmo Aproximado do AVM (Algoritmo de Schweitzer)

Valores iniciais	<pre>ε = 0.001 f = (N-1)/N ΔQ = 10*ε para i = 1 até M faça início     Q<sub>i</sub> = N/M fim</pre>
Iteração	<pre>enquanto ΔQ &gt; ε faça início     R = 0     para i = 1 até M faça início         R<sub>i</sub> = S<sub>i</sub>(1 + f*Q<sub>i</sub>) ou R<sub>i</sub> = S<sub>i</sub>         R += R<sub>i</sub> V<sub>i</sub>     fim     X = N/(R+Z)     ΔQ = 0     para i = 1 até M faça início         se ΔQ &lt;  Q<sub>i</sub>-X*V<sub>i</sub>*R<sub>i</sub>  então ΔQ =  Q<sub>i</sub>-X*V<sub>i</sub>*R<sub>i</sub>      fim     para i = 1 até M faça início         X<sub>i</sub> = X V<sub>i</sub>         Q<sub>i</sub> = X V<sub>i</sub> R<sub>i</sub>         U<sub>i</sub> = X V<sub>i</sub> S<sub>i</sub>     fim fim</pre>

O Exemplo 7.3 é uma aplicação da AVM Aproximada usando a rede de filas da Figura 3.5.

Exemplo 7.3 O sistema de rede de filas da Figura 3.5 será analisado com o método AVM Aproximado considerando que a RSF possui 20 pacotes. O critério de parada do algoritmo é quando a máxima alteração no número de usuários for menor que 0,01.
Parâmetros do sistema:
N = 20 pacotes S <sub>A</sub> = 0,3 s → V <sub>A</sub> = 10 → D <sub>A</sub> = 3 S <sub>B</sub> = 0,2 s → V <sub>B</sub> = 5 → D <sub>B</sub> = 1 D <sub>CPU</sub> = 2 → V <sub>CPU</sub> = V <sub>A</sub> + V <sub>B</sub> + 1 = 16 → S <sub>CPU</sub> = 0,125 s Z = 4 s ε = 0,01
Inicialização
Para inicializar as filas assume-se que todos os 20 pacotes estão igualmente distribuídos nas 3 Filas do sistema: Q <sub>CPU</sub> = Q <sub>A</sub> = Q <sub>B</sub> = 20/3 = 6,67 pacotes
Iteração 1
Pacote N = 1
Tempos de resposta do SF:
R <sub>CPU</sub> = S <sub>CPU</sub> [1+ (N-1)N Q <sub>CPU</sub> ] = 0,125 (1+019/20*6,67) = 0,92 s R <sub>A</sub> = S <sub>A</sub> [1+(N-1)NQ <sub>A</sub> ] = 0,3(1+ 0,95*6,67) = 2,20 s R <sub>B</sub> = S <sub>B</sub> [1+(N-1)NQ <sub>B</sub> ] = 0,2(1+ 0,95*6,67) = 1,47 s
Tempo de resposta da RSF:
R = R <sub>CPU</sub> V <sub>CPU</sub> +R <sub>A</sub> V <sub>A</sub> +R <sub>B</sub> V <sub>B</sub> = 0,92×16+2,20×10+1,47×5 = 44 s
Vazão da RSF:
X = N/(R+Z) = 1/(44+4) = 0,42 pacote/s
Comprimentos de fila do dispositivo:
Q <sub>CPU</sub> = X R <sub>CPU</sub> V <sub>CPU</sub> = 0,42 × 0,92 × 16 = 6,11 Q <sub>A</sub> = X R <sub>A</sub> V <sub>A</sub> = 0,42 × 2,20 × 10 = 9,17 Q <sub>B</sub> = X R <sub>B</sub> V <sub>B</sub> = 0,42 × 1,47 × 5 = 3,06
Variação máxima no número de pacotes: ΔQ = max{  Q <sub>CPU</sub> -X R <sub>CPU</sub> V <sub>CPU</sub>  ;  Q <sub>A</sub> -X R <sub>A</sub> V <sub>A</sub>  ;  Q <sub>B</sub> -X R <sub>B</sub> V <sub>B</sub>   } ΔQ = max{  6,67-0,42×0,92×16 ;  6,67-0,42×2,20×10 ;  6,67-0,42×1,47×5  } ΔQ = max{  0,56;2,5;3,61 } = 3,61 > ε = 0,01
E assim sucessivamente

O Algoritmo 7.2 implementa em C++o software da Tabela 26 com a solução do Exemplo 7.3.

Algoritmo 7.2 Software AVM Aproximado em C++ (Algoritmo de Schweitzer)
//----- #include <string>

```

#include <vector>
#include <cmath>
#include <algorithm>
#include <fstream>
#include <sstream>
//-----
using namespace std;
//-----
void Salvar( string fn, string str ){
    ofstream fs;
    fs.open(fn);
    replace(str.begin(), str.end(), '.', ',');
    fs << str;
    fs.close();
}

string AVMA( double erro, const int M, const int N, double *S, int *V, double Z ){
    int n = 0;
    double R[M], Q[M], U[M], X[M], Rs, Xs, f = (N-1.0)/N, dQ = 1+erro;
    stringstream htm;
    htm << fixed;
    htm.precision(2);
    htm << "Simulacao AVM-A<hr>"
        << "<table border='1'><tr><th>n<th>R<th>X";
    for( int m = 0; m < M; m++ )
        htm << "<th style='background-color:LightGray'>R<sub>" << m << "</sub>"
            << "<th>Q<sub>" << m << "</sub>"
            << "<th>U<sub>" << m << "</sub>"
            << "<th>X<sub>" << m << "</sub>";
    for( int m = 0; m < M; m++ ){ Q[m] = N/M; }
    while( dQ > erro ){
        Rs = 0.0;
        for( int m = 0; m < M; m++ ){
            R[m] = S[m] * (1.0 + f*Q[m]);
            Rs += R[m] * V[m];
        }
        Xs = N/(Rs+Z);
        dQ = 0.0; // dQ maximo
        for( int m = 0; m < M; m++ ){
            double tmp = fabs(Q[m]-Xs*V[m]*R[m]);
            if( dQ < tmp ) dQ = tmp;
        }
        for( int m = 0; m < M; m++ ){
            X[m] = Xs * V[m];
            Q[m] = Xs * V[m] * R[m];
            U[m] = Xs * S[m] * V[m];
        }
        htm << "<tr><td>" << n++ << "<td>" << Rs << "<td>" << Xs;
        for( int m = 0; m < M; m++ )
            htm << "<td style='background-color:Gainsboro'>" << R[m]
                << "<td>" << Q[m]
                << "<td>" << U[m]
                << "<td>" << X[m];
    }
    htm << "</table><br><br><br>";
    return htm.str();
}

int main( void ){
    const int M    = 3,
              N    = 20;
    int      V[M] = { 10, 5, 16 };
    double   S[M] = { 0.3, 0.2, 0.125 },
              Z    = 4.0,
              erro  = 0.01;
    Salvar("AVMA-out.html",AVMA(erro,M,N,S,V,Z));
    return 0;
}

```



A Tabela 27 lista os resultados da aplicação do Exemplo 7.3 no Algoritmo 7.2.

Tabela 27 Resultados do Algoritmo AVM Aproximado

n	R	X	R <sub>0</sub>	Q <sub>0</sub>	U <sub>0</sub>	X <sub>0</sub>	R <sub>1</sub>	Q <sub>1</sub>	U <sub>1</sub>	X <sub>1</sub>	R <sub>2</sub>	Q <sub>2</sub>	U <sub>2</sub>	X <sub>2</sub>
(p)	(s)	(p/s)	(s)	(p)		(p/s)	(s)	(p)		(p/s)	(s)	(p)		(p/s)
0	40,20	0,45	2,01	9,10	1,36	4,52	1,34	3,03	0,45	2,26	0,84	6,06	0,90	7,24
1	46,32	0,40	2,89	11,49	1,19	3,97	0,78	1,54	0,40	1,99	0,85	5,37	0,79	6,36
2	50,43	0,37	3,58	13,14	1,10	3,67	0,49	0,91	0,37	1,84	0,76	4,49	0,73	5,88
3	52,83	0,35	4,04	14,23	1,06	3,52	0,37	0,65	0,35	1,76	0,66	3,70	0,70	5,63
4	54,23	0,34	4,36	14,96	1,03	3,43	0,32	0,56	0,34	1,72	0,56	3,10	0,69	5,50
5	55,08	0,34	4,56	15,45	1,02	3,39	0,31	0,52	0,34	1,69	0,49	2,67	0,68	5,42
6	55,62	0,34	4,70	15,78	1,01	3,35	0,30	0,50	0,34	1,68	0,44	2,38	0,67	5,37
7	55,97	0,33	4,80	16,00	1,00	3,34	0,30	0,49	0,33	1,67	0,41	2,17	0,67	5,34
8	56,20	0,33	4,86	16,15	1,00	3,32	0,29	0,49	0,33	1,66	0,38	2,04	0,66	5,32
9	56,35	0,33	4,90	16,25	0,99	3,31	0,29	0,48	0,33	1,66	0,37	1,94	0,66	5,30
10	56,45	0,33	4,93	16,31	0,99	3,31	0,29	0,48	0,33	1,65	0,36	1,88	0,66	5,29
11	56,52	0,33	4,95	16,35	0,99	3,30	0,29	0,48	0,33	1,65	0,35	1,84	0,66	5,29
12	56,56	0,33	4,96	16,38	0,99	3,30	0,29	0,48	0,33	1,65	0,34	1,82	0,66	5,28
13	56,59	0,33	4,97	16,40	0,99	3,30	0,29	0,48	0,33	1,65	0,34	1,80	0,66	5,28
14	56,61	0,33	4,97	16,41	0,99	3,30	0,29	0,48	0,33	1,65	0,34	1,79	0,66	5,28
15	56,63	0,33	4,98	16,42	0,99	3,30	0,29	0,48	0,33	1,65	0,34	1,78	0,66	5,28

## 8 Projeto e Análise de Experimentos

Um dos problemas mais comuns do engenheiro é determinar a influência de uma ou mais variáveis sobre uma outra variável de interesse. Por exemplo, como a performance de um computador é influenciada pela memória RAM e pelo barramento. No linguajar estatístico, diz-se que o engenheiro está interessado em descobrir como a resposta (performance do computador) depende dos fatores memória RAM e barramento. Este projeto de computador é um caso particular da situação mostrada esquematicamente na Figura 25. Nesta figura um conjunto de fatores  $F = \{ f_1, f_2, \dots, f_m \}$ , atuando sobre o sistema projetado, produz um outro conjunto de respostas  $R = \{ r_1, r_2, \dots, r_n \}$ .



Figura 25 Um sistema projetado pode ser representado por uma função (em princípio desconhecida) ligando os fatores (variáveis de entrada) às respostas (variáveis de saída).

Veja que projetar um novo sistema, como mostrado na Figura 25, é similar a visão geral de programa de computador, Figura 26, bastante conhecida.

Entrada → Processamento → Saída

Figura 26 Visão geral de programa de computador

O sistema projetado atua como uma função – desconhecida, em princípio – que transforma variáveis de entrada (os fatores) e em saídas observáveis (respostas). O objetivo do

engenheiro que faz projetos é descobrir essa função ou obter uma aproximação satisfatória para ela. Com esse conhecimento, o engenheiro poderá entender melhor a natureza do sistema projetado, e assim escolher melhores seus componentes.

O projeto de um equipamento é similar a um experimento, daqui pra frente pense em experimento como projeto.

Exemplo 8.1 Estabelecer os fatores F e as respostas R de projetos de Sistemas de Fila Simples M/M/1. Indicar uma função que transforma F em R.

As entradas (fatores) um Sistema de Fila Simples (SF) é o conjunto  $F = \{\mu, \lambda, N\}$  e suas saídas mais importantes são  $R = \{U, n_f, tsf\}$ , mostradas esquematicamente na Figura 27.

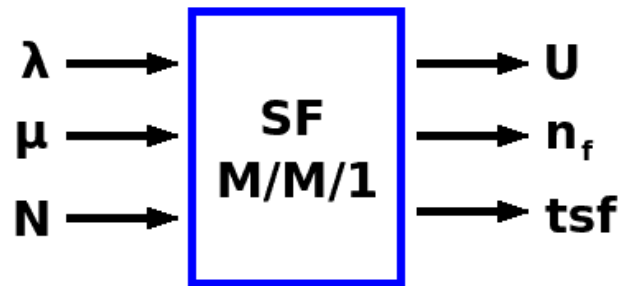


Figura 27 Entradas e saídas de um Sistema de Fila Simples. Uma função que transforma F em R é o Modelo Computacional M/M/1.

O primeiro passo de um experimento é decidir quais são os fatores e as respostas de interesse. Os fatores, em geral, são os componentes dos equipamentos, estas variáveis o projetista tem condições de controlar. Dependendo do projeto, as respostas de interesse estão relacionadas ao desempenho do equipamento.

Exemplo 8.2 Planejamento fatorial para avaliar o efeito do fator Memória RAM em quatro níveis, 4 GB, 8 GB, 12 GB e 16 GB, e do fator Taxa de Gravação em SSD em três níveis, 120 MB/s, 300 MB/s e 450 MB/s, na utilização de uma CPU por meio de uma carga padrão.

O sistema é uma CPU, o fator é o conjunto  $F = \{ \text{Memória RAM (em 4 níveis), Taxa de Gravação SSD ( em 3 níveis) } \}$  e o resultado é o conjunto  $R = \{U\}$ , mostradas esquematicamente na Figura 28.

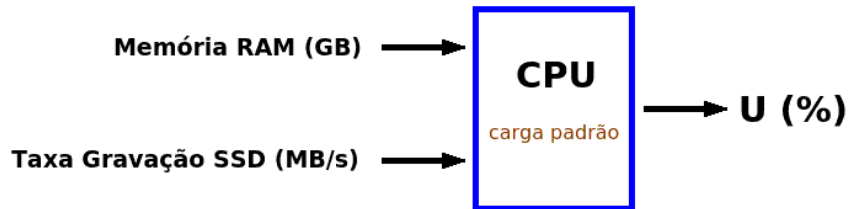


Figura 28 Experimento para avaliar o efeito dos fatores Memória RAM e Taxa de Gravação em SSD na utilização de uma CPU

Uma vez especificados os fatores e seus níveis, bem como a resposta, pode-se planejar o experimento fatorial.

Para fazer o planejamento fatorial deve-se combinar todas as possíveis combinações dos

níveis dos fatores. Neste caso, temos um fator com 4 níveis e o outro com 3 níveis, c, são necessários  $4 \times 3 = 12$  medições ou ensaios diferentes. Este é o chamado Planejamento Fatorial  $4 \times 3$  e seu planejamento está na Tabela 28.

Tabela 28 Planejamento Fatorial  $4 \times 3$  do experimento Efeito da Memória RAM e da Taxa de Gravação em SSD na utilização de uma CPU sob uma carga padrão.

Ensaio	Nível do Fator Memória RAM (GB)	Nível do Fator Taxa de Gravação SSD (MB/s)	Resposta U(%)
1	4 GB	120	$U_1$
2	4 GB	300	$U_2$
3	4 GB	450	$U_3$
4	8 GB	120	$U_4$
5	8 GB	300	$U_5$
6	8 GB	450	$U_6$
7	12 GB	120	$U_7$
8	12 GB	300	$U_8$
9	12 GB	450	$U_9$
10	16 GB	120	$U_{10}$
11	16 GB	300	$U_{11}$
12	16 GB	450	$U_{12}$

O Engenheiro deve realizar o experimento, executar os pacotes de cada ensaio, preencher a tabela acima, analisar os resultados obtidos, manter suas memórias e cálculo, emitir sua conclusão. Seu relato deve ser suficiente para que outro engenheiro seja capaz de refazer o trabalho realizado.

A Tabela 29 é o Gabarito do Planejamento Fatorial  $4 \times 3$ .

Tabela 29 Gabarito do Planejamento Fatorial  $4 \times 3$

Ensaio	Nível do Fator 1 (NF1)	Nível do Fator 2 (NF2)	Resposta (R)
1	1	1	R11
2	1	2	R12
3	1	3	R13
4	2	1	R21
5	2	2	R22
6	2	3	R23
7	3	1	R31
8	3	2	R32
9	3	3	R33
10	4	1	R41
11	4	2	R42
12	4	3	R43

Para k fatores, isto é, k variáveis controladas pelo experimentador, um planejamento completo de dois níveis requer a realização de  $2 \times 2 \times \dots \times 2 = 2^k$  ensaios diferentes, sendo chamado Planejamento Fatorial  $2^k$ .

Exemplo 8.3 Experimento Fatorial 2 × 2 do experimento Efeito da Memória RAM (4 e 12 GB) e da Taxa de Gravação SSD (120 e 300 MB/s) na utilização de uma CPU sob uma carga padrão, com duas replicações

Na Tabela 30 estão os resultados dos quatro ensaios do experimento Efeito da Memória RAM (4 e 12 GB) e da Taxa de Gravação SSD (120 e 300 MB/s) na utilização de uma CPU sob uma carga padrão. Os ensaios foram feitos em duplicata, produzindo oito respostas (Utilização) no total, com isso, pode-se estimar o erro experimental de cada ensaio.

Tabela 30 Resultado do Planejamento Fatorial 2 × 2 do experimento Efeito da Memória RAM (4 e 12 GB) e da Taxa de Gravação SSD (120 e 300 MB/s) na utilização de uma CPU sob uma carga padrão

Ensaio	Memória RAM (GB)	Taxa de Gravação SSD (MB/s)	Utilização(%)		Média (%)
			1	2	
1	4	120	57	61	59
2	4	300	92	88	90
3	12	120	55	53	54
4	12	300	66	70	68

O Engenheiro deve realizar o experimento, executar as pacotes de cada ensaio, preencher a tabela acima, analisar os resultados obtidos, manter suas memórias e cálculo, emitir sua conclusão. Seu relato deve ser suficiente para que outro engenheiro seja capaz de refazer o trabalho realizado.

Os planejamentos fatoriais 2×2 são úteis em investigações preliminares, quando o objetivo é saber se determinados fatores têm ou não influência sobre a resposta, sem estar preocupado ainda em descrever muito rigorosamente essa possível influência. São planejamentos muito simples de executar, que depois podem ser ampliados para formar um planejamento mais sofisticado, para conhecer melhor a relação entre a resposta e os fatores importantes.

## 8.1 Experimento com Um Fator

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<sup>k</sup>, o número de níveis pode ser superior a 2.

O Exemplo 8.4 apresenta um exemplo de experimento com um fator com 4 níveis.

Exemplo 8.4 Uma Engenheira está interessada em investigar a relação entre a configuração da memória RAM DDR4 e a velocidade de gravação em SSD.

O objetivo de um experimento é modelar a relação entre memória RAM DDR4 e a taxa de

transferência para SSD, e especificar a memória RAM DDR4 para fins de projeto. Neste caso o **Fator** é memória RAM DDR4 (GB) e a **Resposta** é a taxa de transferência para SSD (MB/s)

As questões de interesse são: Uma variação na memória RAM afeta a velocidade média de gravação na SSD? Há um nível ótimo de memória RAM? É importante poder responder estas questões de forma objetiva.

Foi feito um experimento para avaliar quatro tamanhos de RAM DDR4: 4 GB, 8 GB, e 12 GB e 16 GB, e decidiu medir 5 vezes cada avaliação do tamanho de memória utilizado. Assim, tem-se um experimento a um fator com  $a = 4$  níveis (4 GB, 8 GB, 12 GB, 16 GB) e  $n = 5$  replicações realizadas em ordem aleatória.

Os dados de gravação em SSD (MB/s) variando o tamanho da Memória RAM, com 5 replicações obtidos estão apresentados na Tabela 31.

Tabela 31 Dados de gravação em SSD (MB/s) variando o tamanho da Memória RAM, com 5 replicações

Memória RAM DDR4 (nível)	Taxa de Transferência para SSD (MB/s)				
	1	2	3	4	5
1	279	288	278	291	272
2	301	292	289	300	293
3	310	307	303	300	299
4	330	332	329	335	323

O diagrama de dispersão da memória RAM DDR4 versus taxa de transferência para a SSD e boxplot da memória RAM DDR4 versus taxa de transferência para a SSD, da Figura 29 permite uma inspeção visual do comportamento dos dados experimentais.

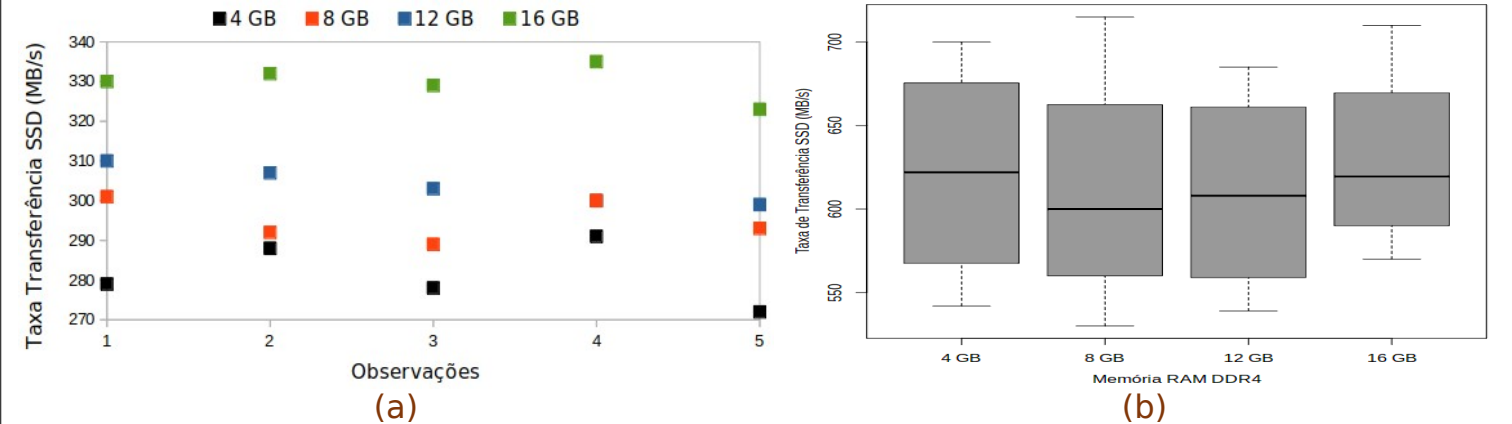


Figura 29 a) Diagrama de dispersão da memória RAM DDR4 versus taxa de transferência para a SSD; b) Boxplot da memória RAM DDR4 versus taxa de transferência para a SSD.

A técnica a ser usada neste caso é a Análise de Variância (ANOVA), amplamente usada nas mais variadas áreas, particularmente em experimentos industriais.

A Tabela 32 contém os dados típicos para o experimento de um fator. Nesta tabela pode-se observar que existem **a** níveis do fator (tratamentos), para cada nível são realizadas **n** observações (replicações) de forma aleatória. Num total de  $N = a \times n$  observações. Temos

ainda o total e a media das observações de cada nível do fator. Nesta tabela,  $y_{ij}$  é o valor observado do nível  $i$  e replicação  $j$ ,  $y_{i\cdot}$  é a soma dos valores observados do nível  $i$  e  $\bar{y}_{i\cdot}$  é a média aritmética dos valores observados do nível  $i$ ,  $i \in [1,a]$  e  $j \in [1,n]$ .

Tabela 32 Dados típicos para o experimento de um fator<sup>4</sup>

Tratamento (nível)	Observações							Totais	Média
	1	2	3	...	j	...	n		
1	$y_{11}$	$y_{12}$	$y_{13}$	...	$y_{1j}$	...	$y_{1n}$	$y_{1\cdot}$	$\bar{y}_{1\cdot}$
2	$y_{21}$	$y_{22}$	$y_{23}$	...	$y_{2j}$	...	$y_{2n}$	$y_{2\cdot}$	$\bar{y}_{2\cdot}$
3	$y_{31}$	$y_{32}$	$y_{33}$	...	$y_{3j}$	...	$y_{3n}$	$y_{3\cdot}$	$\bar{y}_{3\cdot}$
...				...	...	...	...	...	...
i	$y_{i1}$	$y_{i2}$	$y_{i3}$	...	$y_{ij}$	...	$y_{in}$	$y_{i\cdot}$	$\bar{y}_{i\cdot}$
...	...	...	...	...	...	...	...	...	...
a	$y_{a1}$	$y_{a2}$	$y_{a3}$	...	$y_{aj}$	...	$y_{an}$	$y_{a\cdot}$	$\bar{y}_{a\cdot}$

Sendo que:

$y_{i\cdot} = \sum_{j=1}^n y_{ij}, \quad i \in [1,a]$	8.1
$\bar{y}_{i\cdot} = \frac{\sum_{j=1}^n y_{ij}}{n} = \frac{y_{i\cdot}}{n}, \quad i \in [1,a]$	8.2

8.1.1 Modelo

Para fins de Engenharia, será adotado o Modelo de Efeitos da Análise de Variância<sup>5</sup> (ANOVA), expresso pela Equação 8.3. Os efeitos são calculados de modo que eles somam zero, ou seja,  $\sum \tau_i = 0$  .

4 Observe que nessa tabela as amostras têm tamanhos iguais, essa restrição não é necessária.  
5 O nome “Análise de Variância” deve-se ao particionamento da variabilidade total da variável resposta (decomposição da soma de quadrados total) em componentes que são consistentes com o modelo adotado para o experimento.

$y_{ij}=\mu+\tau_i+\varepsilon_{ij}, \quad i \in [1,a] \text{ e } j \in [1,n]$	8.3
--	-----

em que:

- $y_{ij}$  é a j-ésima observação do i-ésimo nível de tratamento
- $\mu$  é a média dos valores de todos níveis e de todos tratamentos
- $\tau_i$  é o efeito do i-ésimo nível do fator ou do i-ésimo tratamento
- $\varepsilon_{ij}$  é o erro experimental aleatório do i-ésimo nível e da j-ésima observação

O valor  $\mu$ , a média dos valores de todos níveis e de todos tratamentos, é estimada pela Equação 8.4. Vale ressaltar que  $\bar{y}_{..} = \mu$ , Equação 8.5.

$\mu = \frac{\sum_{i=1}^a \sum_{j=1}^n y_{ij}}{N}$	8.4
$\bar{y}_{..} = \frac{\sum_{i=1}^a \sum_{j=1}^n y_{ij}}{N}$	8.5

O erro experimental  $\varepsilon_{ij}$  pode incorporar outras fontes de variação no experimento, incluindo erros de medidas, variabilidades que surgem de fatores não controláveis, diferenças entre as unidades experimentais às quais os tratamentos são aplicados e ruídos do próprio processo.

Em geral, assume-se que o erro é normalmente distribuído com média 0 ( $\sum \sum \varepsilon_{ij} = 0$ ), além disso, também assume-se que  $\text{Var}(\varepsilon_{ij}) = \sigma^2$ , tal que as variâncias são homogêneas para os diferentes níveis de tratamento.

A Equação 8.3 é um modelo linear, isto é, a variável resposta  $y_{ij}$  é uma função linear dos parâmetros do modelo, é chamado de modelo de análise de variância a um fator.

Além disso a ordem na qual os tratamentos são designados às unidades experimentais é aleatória de modo que o experimento aqui é aleatorizado.

Objetivo: testar de forma apropriadas hipóteses sobre as médias de tratamento e estimá-las. Para fazer inferência sobre os parâmetros do modelo são necessárias suposições sobre a distribuição dos dados. Em ANOVA supomos que os erros são independentes com variância constante  $\sigma^2$  e normalmente distribuídos:  $\varepsilon_{ij} \sim \text{NID}(0, \sigma^2)$ . Isto implica que  $y_{ij} \sim N(\mu+\tau_i, \sigma^2)$ , para  $\mu_i = \mu+\tau_i$ .

Na Engenharia, os componentes dos projetos são selecionados de catálogos e/ou *datasheets*, ou seja, os Fatores não são aleatórios e, assim, é necessário considerá-los fixos, determinísticos.



No modelo ANOVA, quando os tratamentos são escolhidos pelo experimentador, como é o caso, testa-se hipóteses sobre as médias de tratamento e as conclusões são aplicáveis somente aos níveis considerados na análise, não podendo ser estendidas para tratamentos similares que não foram explicitamente considerados. Pode-se também estimar  $\mu$ ,  $\tau_i$  e  $\sigma^2$ . Este é chamado modelo de efeitos fixos.

A Análise de Variância para o Modelo de Efeitos Fixos decompõe a Soma de Quadrados Total (SQT) em Soma de Quadrados dos Tratamentos (SQTrat) e Soma de Quadrados dos Resíduos (erros), Equação 8.6.

$SQT = SQTrat + SQRes$	8.6
------------------------	-----

O número de graus de liberdade total é  $a \times n - 1$ , o número de graus de liberdade devido a tratamento é  $a - 1$  e, o número de graus de liberdade dos resíduos é  $a(n - 1)$ , como mostrado na Equação 8.7.

$a \times n - 1 = a - 1 + a(n - 1).$	8.7
--------------------------------------	-----

As hipóteses de interesse aqui são:

- $H_0: \mu_1 = \mu_2 = \dots = \mu_a$  versus
- $H_1$ : pelo menos uma média difere das demais.

As Somas de quadrados não podem ser diretamente comparadas para testar a hipótese de igualdade das médias. Mas, os quadrados médios, definidos pela razão entre a soma de quadrados e seus respectivos graus de liberdade, podem ser comparados para testar essa hipótese.

$QMTrat = \frac{SQTrat}{(a - 1)}$	8.8
$QMRes = \frac{SQRes}{a(n - 1)}$	8.9

Se as médias de tratamentos são iguais ( $\mu_1 = \mu_2 = \dots = \mu_a$ ), os valores esperados dos quadrados médios de tratamento e de resíduos são iguais.

Se as médias de tratamento diferem, o valor esperado do quadrado médio de tratamento será maior que o valor esperado do quadrado médio do resíduo.

A Tabela 33 é o gabarito de como montar a Tabela do Modelo de Efeitos da ANOVA a um fator, as Equações 8.10 a 8.16 referem-se aos cálculos indicados nesta tabela.

Tabela 33 Tabela do Modelo de Efeitos da ANOVA a um fator

Fonte de Variação	gl	SQ	QM	F <sub>0</sub>
Tratamento	a-1	SQTrat	QMTrat	QMTrat/QMRes
Erro	a(n-1)	SQRes	QMRes	
Total	a×n-1	SQT		

Em que:

$\bar{y}_{..} = \frac{\sum_{i=1}^a \sum_{j=1}^n y_{ij}}{a \times n}$	8.10
$SQT = \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - \bar{y}_{..})^2$	8.11
$SQTrat = n \sum_{i=1}^a (\bar{y}_{i.} - \bar{y}_{..})^2$	8.12
$SQRes = SQT - SQTrat$	8.13
$QMTrat = \frac{SQTrat}{(a - 1)}$	8.14
$QMRes = \frac{SQRes}{a(n - 1)}$	8.15
$F_0 = \frac{QMTrat}{QMRes}$	8.16

Sob H<sub>0</sub>, F<sub>0</sub> tem distribuição F com a–1 e a(n–1) graus de liberdade.  
Ao nível de significância α, rejeitaremos H<sub>0</sub> se F<sub>0</sub> ≥ F[1–α; a–1; a(n–1)].  
F[1–α; a–1; a(n–1)] representa o quantil acumulado de 1–α da distribuição F[a–1; a(n–1)].

Exemplo 8.5 Verifique se as médias dos tratamentos do Exemplo 8.4, o experimento Memória × SSD são iguais ao nível de significância de 5%.

Esta questão refere-se à hipótese H<sub>0</sub> e, sendo assim, a solução é fazer a Tabela Modelo da ANOVA deste experimento, calcular F<sub>0</sub> e verificar se F<sub>0</sub> < F[1–α; a–1; a(n–1)] ao nível de significância α = 5%.

Os dados de gravação em SSD (MB/s) variando o tamanho da Memória RAM, com 5 replicações obtidos estão repetidos na Tabela 34, m que **a** = 4, **n** = 5 e **N** = 20.

Tabela 34 Dados de gravação em SSD (MB/s) variando o tamanho da Memória RAM, com 5 replicações

Memória RAM DDR4 (nível)	Taxa de Transferência para SSD (MB/s)					Totais (MB/s)	Média (MB/s)
	1	2	3	4	5		
1	279	288	278	291	272	1408	282
2	301	292	289	300	293	1475	295
3	310	307	303	300	299	1519	304
4	330	332	329	335	323	1649	330

A Tabela 35 é a Tabela do Modelo de Efeitos da ANOVA a um fator do experimento Memória × SSD, preenchido a partir dos dados da Tabela 34, ao gabarito da Tabela 33 e suas equações auxiliares.

Tabela 35 Tabela do Modelo de Efeitos da ANOVA a um fator do experimento Memória × SSD

Fonte de Variação	gl	SQ	QM	F <sub>0</sub>
Tratamento	3	6200	2067	64,0
Erro	16	517	32	
Total	19	6717		

Com o auxílio de planilha, foram calculados (sequencialmente):

$\bar{y}_{..} = 303$   
 $SQT = 6717$   
 $SQ_{Trat} = 6200$   
 $SQ_{Res} = 517$   
 $QM_{Trat} = 2067$   
 $QM_{Res} = 32$   
 $F_0 = 64,0$

Neste caso  $F[1-\alpha; a-1; a(n-1)]$  e  $F(95\%; 3; 16) = 3,24$  (valor tabelado).  
Como  $F_0 = 64,0 \geq F(0,95; 3; 16) = 3,24$ , rejeita  $H_0$  a nível de significância de 5%.

**Conclusão:** As médias dos tratamentos do experimento Memória × SSD (Exemplo 8.4) não são todas iguais ao nível de confiança de 95%.

Exemplo 8.6 Verifique se as médias dos tratamentos do Exemplo 8.4, o experimento Memória × SSD são iguais ao nível de significância de 5%, utilizando o software R.

Prepara-se o arquivo de dados, neste caso seu nome é file.csv, em seguida executa-se o comando listado a seguir e interpreta-se os resultados.

file.csv	Comandos R para o Modelo de Efeitos da ANOVA a um fator e seus resultados
F;R	Comandos R:
1;279	
2;301	dados <- read.csv2("/home/leal/file.csv", sep=";")
3;310	anova <- aov(dados\$R ~ as.factor(dados\$F))
4;330	summary(anova)
1;288	
2;292	Resultados:
3;307	Df Sum Sq Mean Sq F value Pr(>F)
4;332	as.factor(dados\$F) 3 6200 2066.7 63.98 3.96e-09 ***
1;278	Residuals 16 517 32.3
2;289	---
3;303	Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
4;329	
1;291	R version 3.5.2 Platform: x86_64-pc-linux-gnu (64-bit)
2;300	
3;300	
4;335	
1;272	
2;293	
3;299	
4;323	

F(95%; 3; 16) = 3,24 (valor tabelado).

Como  $F_0 = 64,0 \geq F(0,95; 3; 16) = 3,24$ , rejeita  $H_0$  a nível de significância de 5%.

**Conclusão:** As médias dos tratamentos do experimento Memória × SSD (Exemplo 8.4) não são todas iguais ao nível de confiança de 95%.

8.1.2 Cálculo dos Efeitos

Os dados medidos em um experimento de um fator consiste de **a** níveis do fator e **n** observações para cada um destes níveis, com **N = a × n** dados, que são dispostos em uma matriz **a × n**, mostrado anteriormente na Tabela 32.

Substituindo os valores  $y_{ij}$  na Equação 8.3, que é o Modelo de Efeitos da Análise de Variância, obtém-se  $a \times n$  equações, expressas pela Equação 8.17.

$\sum_{i=1}^a \sum_{j=1}^n y_{ij} = a \times n \mu + n \sum_{i=1}^a \tau_i + \sum_{i=1}^a \sum_{j=1}^n \varepsilon_{ij}$	8.17
--	------

Como a soma dos efeitos é zero, ou seja,  $\sum \tau_i = 0$  ; e assumindo que o erro é normalmente distribuído com média 0, isto é  $\sum \varepsilon_{ij} = 0$ , obtém-se a Equação 8.18

$\sum_{i=1}^a \sum_{j=1}^n y_{ij} = a \times n \mu + 0 + 0$	8.18
---	------

Logo, o parâmetro  $\mu$  do modelo é dado pela Equação 8.19.

$\mu = \frac{\sum_{i=1}^a \sum_{j=1}^n y_{ij}}{a \times n}$	8.19
---	------

A quantidade do lado direito é a chamada média geral de todas as  $a \times n$  respostas, e é denotada por  $\bar{y}_{..}$ , 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 linhas e coluna, que são obtidos por uma média de respostas pertencentes a uma determinada linha ou coluna.

A média da i-ésima linha, denotado por  $\bar{y}_{i.}$ , é calculada pela Equação 8.20.

$\bar{y}_{i.} = \frac{\sum_{j=1}^n y_{ij}}{n}$	8.20
--	------

Substituindo a Equação 8.3 na Equação 8.20, obtém-se a Equação 8.21.

$$\bar{y}_{i.} = \frac{\sum_{j=1}^a (\mu + \tau_i + \varepsilon_{ij})}{n} = \frac{n\mu + n\tau_i + \sum_{j=1}^a e_{ij}}{n} = \mu + \tau_i$$

8.21

Aqui assumimos que a soma dos termos de erro das **a** observações de cada nível é zero. O parâmetro  $\tau_i$  pode ser estimado pela Equação 8.22.

$$\tau_i = \bar{y}_{i.} - \mu = \bar{y}_{i.} - \bar{y}_{..}$$

8.22

Exemplo 8.7 Estudo para avaliar a influência do código como carga de trabalho de processadores.

Foi utilizado um código para avaliar sua influência como carga de trabalho nos processadores A, B e C. O código foi executado em cada um dos processadores e mediu-se o tempo de processamento. Foram realizadas cinco repetições com cada processador. O fator neste experimento é o processador, ele tem três níveis: A, B e C. A resposta é o tempo de processamento (ms) da carga de trabalho. Foram feitas 5 réplicas. Os dados medidos estão mostrados na Tabela 36.

Tabela 36 Dados do Estudo Processador versus Carga de Trabalho

Fator	Tempo de Processamento (ms)				
	1	2	3	4	5
A	144	120	176	288	144
B	101	144	211	288	72
C	130	180	141	374	302

A análise é mostrada na Tabela 37. Foi calculada a média dos três níveis do fator ( $\bar{y}_{1.}$ ,  $\bar{y}_{2.}$  e  $\bar{y}_{3.}$ ) e a média de todos os dados ( $\bar{y}_{..}$ ). Em seguida foi usada a Equação 8.22 para calcular os efeitos de cada fator:  $\tau_1 = -13,3$  ms;  $\tau_2 = -24,5$  ms e  $\tau_3 = 37,7$  ms.

Tabela 37 Análise do Estudo Comparativo do Tamanho do Código

Fator	Tempo de Processamento (ms)					Média (ms)	Efeito (ms)
	1	2	3	4	5		
A	144	120	176	288	144	$\bar{y}_{1.} = 174,4$	$\tau_1 = \bar{y}_{1.} - \bar{y}_{..} = -13,3$
B	101	144	211	288	72	$\bar{y}_{2.} = 163,2$	$\tau_2 = \bar{y}_{2.} - \bar{y}_{..} = -24,5$
C	130	180	141	374	302	$\bar{y}_{3.} = 225,4$	$\tau_3 = \bar{y}_{3.} - \bar{y}_{..} = 37,7$
						$\bar{y}_{..} = 187,7$	

Os resultados são interpretados da seguinte forma. Um processador médio requer 187,7 ms para processar a carga de trabalho. O processador A requer 13,3 ms a menos do que o processador médio, B requer 24,5 ms a menos do que o processador médio, e C requer 37,7 ms a mais do que o processador médio.

**Conclusão:** Os processadores requerem diferentes tempos para processar a carga de trabalho. O processador B é o que usa menos tempo e o processador C é o que usa mais tempo.

8.1.3 Estimativa dos Erros Experimentais

Uma vez que os parâmetros do modelo foram calculados, pode-se estimar a resposta para cada um dos níveis.

A diferença entre a resposta estimada e a medida é o erro experimental. A soma dos erros experimentais das **a** × **n** observações é zero pois os parâmetros  $\mu$  e  $\tau_i$  foram calculados assumindo que a soma dos erros de cada linha é zero. A variância dos erros pode ser estimada a partir da Soma do Quadrado dos Erros (SQE)

$$SQE = \sum_{i=1}^a \sum_{j=1}^n \epsilon_{ij}^2$$

8.23

Cada um dos valores de  $\epsilon_{ij}$  é obtido resolvendo a matriz de Equação 8.24 utilizando os valores de  $y_{ij}$ ,  $\mu$  e  $\tau_i$  previamente estimados.

$$\begin{bmatrix} y_{11} & y_{12} & \dots & y_{1j} & \dots & y_{1n} \\ y_{21} & y_{22} & \dots & y_{2j} & \dots & y_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y_{i1} & y_{i2} & \dots & y_{ij} & \dots & y_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ y_{a1} & y_{a2} & \dots & y_{aj} & \dots & y_{an} \end{bmatrix} = \begin{bmatrix} \mu & \mu & \dots & \mu & \dots & \mu \\ \mu & \mu & \dots & \mu & \dots & \mu \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mu & \mu & \dots & \mu & \dots & \mu \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \mu & \mu & \dots & \mu & \dots & \mu \end{bmatrix} + \begin{bmatrix} \tau_1 & \tau_1 & \dots & \tau_1 & \dots & \tau_1 \\ \tau_2 & \tau_2 & \dots & \tau_2 & \dots & \tau_2 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \tau_i & \tau_i & \dots & \tau_i & \dots & \tau_i \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \tau_a & \tau_a & \dots & \tau_a & \dots & \tau_a \end{bmatrix} + \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \dots & \epsilon_{1j} & \dots & \epsilon_{1n} \\ \epsilon_{21} & \epsilon_{22} & \dots & \epsilon_{2j} & \dots & \epsilon_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \epsilon_{i1} & \epsilon_{i2} & \dots & \epsilon_{ij} & \dots & \epsilon_{in} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \epsilon_{a1} & \epsilon_{a2} & \dots & \epsilon_{aj} & \dots & \epsilon_{an} \end{bmatrix}$$

8.24

Exemplo 8.8 Calcular SQE do Experimento Processador versus Carga de Trabalho, partir dos dados do estudo para avaliar a influência do código como carga de trabalho de processadores.

A partir dos dados da Tabela 37 e do gabarito da Equação 8.24, foi montado a 8.25 e resolvida.

$$\begin{bmatrix} 144 & 120 & 176 & 288 & 144 \\ 101 & 144 & 211 & 288 & 72 \\ 130 & 180 & 141 & 374 & 302 \end{bmatrix} = \begin{bmatrix} 187,7 & 187,7 & 187,7 & 187,7 & 187,7 \\ 187,7 & 187,7 & 187,7 & 187,7 & 187,7 \\ 187,7 & 187,7 & 187,7 & 187,7 & 187,7 \end{bmatrix} + \begin{bmatrix} -13,3 & -13,3 & -13,3 & -13,3 & -13,3 \\ -24,5 & -24,5 & -24,5 & -24,5 & -24,5 \\ +37,7 & +37,7 & +37,7 & +37,7 & +37,7 \end{bmatrix} + \epsilon$$

8.25

$$\varepsilon = \begin{bmatrix} -30,4 & -54,4 & 1,6 & 113,6 & -30,4 \\ -63,2 & -20,2 & 46,8 & 123,8 & -92,2 \\ -97,4 & -47,4 & -86,4 & 146,6 & 74,6 \end{bmatrix}$$

O valor de SQE =  $= (-30,4)^2 + (-54,4)^2 + \dots + (74,6)^2 = 94390,2 \text{ m}^2\text{s}^2$ .

8.1.4 Atribuição da Variação

A variação da resposta de um experimento pode ser atribuída ao fator e aos erros. Portanto é necessário separar a variação do fator da variação dos erros. Para isso, eleva-se ao quadrado ambos os lados da Equação 8.3, equação do será adotado o Modelo de Efeitos da Análise de Variância.

$$y_{ij}^2 = (\mu + \tau_i + e_{ij})^2 = \mu^2 + \tau_i^2 + e_{ij}^2 + 2\mu\tau_i + 2\mu e_{ij} + 2\tau_i e_{ij}$$

8.26

Adicionando os **a × n** termos correspondentes das equações, obtém-se a Equação 8.27.

$$\sum_{i=1}^a \sum_{j=1}^n y_{ij}^2 = \sum_{i=1}^a \sum_{j=1}^n \mu^2 + \sum_{i=1}^a \sum_{j=1}^n \tau_i^2 + \sum_{i=1}^a \sum_{j=1}^n e_{ij}^2 + \text{termos de produtos cruzados}$$

8.27

Os termos de produtos cruzados são iguais a zero pois  $\sum \tau_i = 0$  e  $\sum \sum e_{ij} = 0$ .  
A equação anterior, expressa em termos de somas de quadrados, pode ser escrita como a Equação 8.28.

$$SQY = SQO + SQA + SQE$$

8.28

Onde SQY é a soma dos quadrados de y, SQO é a soma dos quadrados das médias, SQA é a soma dos quadrados dos efeitos, e SQE é a soma do quadrado dos erros. Note-se que SQO e SQA podem ser calculadas pelas Equações 8.29 e 8.30, respectivamente.



$SQO = \sum_{i=1}^a \sum_{j=1}^n \mu^2 = n \times a \mu^2$	8.29
$SQA = \sum_{i=1}^a \sum_{j=1}^n \tau_i^2 = n \sum_{i=1}^a \tau_i^2$	8.30

Assim, SQE pode ser calculada a partir SQY sem calcular os erros individuais. A variação total de y (SQT) é expressa pela Equação 8.31.

$SQT = \sum_{i=1}^a \sum_{j=1}^n (y_{ij} - \bar{y})^2 = \sum_{i=1}^a \sum_{j=1}^n y_{ij}^2 - n \times a \bar{y}^2 = SQY - SQO = SQA + SQE$	8.31
--	------

A variação total pode ser dividida em duas partes, SQA e SQE, que representam a parte que é explicada pelos efeitos e a parte que é explicada pelos erros, Equação 8.32. A variação dos resultados podem ser expressas como frações da variação total. Quanto maior for a fração SQA/SQT melhor será o modelo.

$\frac{SQA}{SQT} + \frac{SQE}{SQT} = 1$	8.32
---	------

Exemplo 8.9 Estudo para avaliar a influência do código como carga de trabalho de processadores.

A partir dos dados da Tabela 37, pode-se calcular:  
  
 $SQY = 1.442 + 1.202 + \dots + 3.022 = 633.639,0$   
 $SQO = 3 \times 5 \times (187,7)^2 = 528.469,0$   
 $SQA = 5[(-13,3)^2 + (-24,5)^2 + (37,6)^2] = 10.992,0$   
 $SQT = SQY - SQO = 633.639,0 - 528.281,7 = 105.170,0$   
 $SQE = SQT - SQA = 105.357,3 - 10.992,1 = 94.178,0$   
 $SQA/SQT = 100 \times 10.992/105.170 = 10,4\%.$   
 $SQE/SQT = 100 \times 94.178/105.170 = 89,6\%.$   
Porcentagem de variação explicada pelos processadores é igual a 10,5%.  
Porcentagem de variação explicada pelos erros experimentais é igual a 89,5%.  
  
**Conclusão:** Os processadores explicação apenas aproximadamente 11% da variação dos resultados, indicando que é necessário incluir outros fatores no experimento.

### 8.1.5 Análise da Variação

A atribuição da variação a diferentes fatores é uma abordagem muito útil na prática. Dessa forma, qualquer fator que explicar com elevado percentual de variação pode ser considerado importante. Ser importante não é o mesmo que ser significativo em termos estatísticos.

Para determinar se um fator tem um efeito estatístico significativo sobre a resposta, compara-se a sua contribuição para a variação dos erros. Se a variação inexplicada (devido a erros) é elevada, 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 - SQY, SQO, SQA e SQE. Cada uma das somas dos quadrados tem um grau de liberdade associado que corresponde ao número de valores independentes necessários para seu cálculo. Os graus de liberdade das parcelas da equação  $SQY = SQO + SQA + SQE$  (Equação 8.3), são descritos a a seguir:

- A soma SQY consiste de uma soma de  $n \times a$  termos, todos os quais podem ocorrer de forma independente, portanto, tem  $n \times a$  graus de liberdade
- A soma SQO consiste em um único termo  $\mu^2$  que é repetida  $n \times a$  vezes, como ocorre um único valor de  $\mu$ , SQO tem 1 (um) grau de liberdade
- A soma SQA é a soma dos termos  $\tau_i^2$  mas apenas  $a-1$  delas são independentes pois a soma de  $\tau_i$  é zero, logo SQA tem um  $a-1$  graus de liberdade.
- A soma SQE consiste de  $n \times a$  termos de erro, dos quais apenas  $a(n-1)$  podem ocorrer de forma independente. Como a soma os erros das  $n$  replicações de cada nível do fator deve ser zero, somente  $n-1$  erros em cada um das replicações são independentes.

Desta forma, pode-se escrever a equação dos graus de liberdade dos dois lados da Equação 8.3.

$n \times a = 1 + (a - 1) + a(n - 1)$	8.33
---------------------------------------	------

O Teste F pode ser usado para verificar se SQA é significativamente maior do que SQE. Assumindo que os erros são normalmente distribuídos, SQE e SQA têm distribuições Qui-quadrado. A relação  $(SQA/gIA)/(SQE/gIE)$ , onde  $gIA = a-1$  e  $gIE = a(n-1)$ , tem uma distribuição F com numerador  $gIA$  e  $gIE$  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, SQA é considerado significativamente maior que SQE. A quantidade  $SQA/glA$  é chamado de quadrado médio de de A (QMA). De forma similar,  $SQE/gIE$  é chamado de quadrado médio dos erros (QME). Se a relação calculado  $QMA/QME$  é maior do que o valor de quantis da variável F tabelado, assume-se que o fator explica uma fração significativa da variação. Um arranjo tabular conveniente para realizar o teste F é mostrado na Tabela 38.

Tabela 38 Tabela ANOVA para comparar SQA com SQE

Componente	Soma de Quadrados	Fração da Variação	Graus de Liberdade	Quadrado da Média	F Calculado	F Tabelado
y	$\sum y_{ij}^2$		$a \times n$		QMA/QME	$F[1-\alpha; a-1; a(n-1)]$
$\bar{y}..$	$SQO = a \times n \mu^2$		1			
$y-\bar{y}..$	$SQT=SQY-SQO$		$a \times n-1$			
A	$SQA = n \sum \tau_i^2$	$SQA/SQT$	$a-1$	$QMA = SQA/(a-1)$		
e	$SQE=SQT-SQA$	$SQE/SQT$	$a(n-1)$	$QME = SQE/[a(n-1)]$		

$s_e = QME^{0,5}$

Exemplo 8.10 Tabela ANOVA para a avaliar a influência do código como carga de trabalho de processadores está preenchida na Tabela 39. 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 ANOVA para a avaliar a influência do código como carga de trabalho de processadores está preenchida na Tabela 39, como o valor de F calculado é menor que o de Ftabelado conclui-se que a diferença observada nos resultados é devido, principalmente, a erros experimentais e não à diferenças significativa entre os processadores.

Tabela 39 Tabela ANOVA para a avaliar a influência do código como carga de trabalho de processadores está preenchida na .

Componente	Soma de Quadrados	Fração da Variação	Graus de Liberdade	Quadrado da Média	F Calculado	F Tabelado
y	633.639		25			
$\bar{y}..$	528.469		1			
$y-\bar{y}..$	105.170	1,0	24			
A	10992	0,105	2	5.496	0,7	2,8
e	94.178	0,895	12	7.848		

$s_e = QME^{0,5} = 7.848^{0,5} = 88,6$

**Conclusão:** a diferença observada nos valores observados é devido a erros experimentais e não à diferenças significativas entre os processadores.

## 8.2 Experimento Fatorial 2<sup>2</sup>

Um projeto experimental 2<sup>2</sup> é um caso especial de um projeto 2<sup>k</sup> 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 8.11** 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 40.

Tabela 40 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 memória

$x_A = +1$ , se 16 GB de memória

$x_B = -1$ , 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:

$$15 = q_0 - q_A - q_B + q_{AB}$$

$$45 = q_0 + q_A - q_B - q_{AB}$$

$$25 = q_0 - q_A + q_B + q_{AB}$$

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

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.

### 8.2.1 Cálculo dos Efeitos

Em geral, todo o projeto  $2^2$  pode ser analisado usando o método do Exemplo 8.11. 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 41.

Tabela 41 Análise do delineamento  $2^2$

Tratamento	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:

$y_1 = q_0 - q_A - q_B + q_{AB}$

$y_2 = q_0 + q_A - q_B - q_{AB}$

$y_3 = q_0 - q_A + q_B - q_{AB}$

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

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.

8.2.2 Método da Tabela de Sinais

Para um projeto de 2<sup>2</sup>, os efeitos podem ser facilmente calculado através da preparação de uma matriz de sinal 4x4, como mostrado na Tabela 42.

Tabela 42 Método da Tabela de Sinais para o Cálculo dos Efeitos de Projetos 2<sup>2</sup>

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.

8.2.3 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:  $s_e^2 = \sum (y_i - \bar{y})^2 / (2^2 - 1)$

Aqui,  $\bar{y}$  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:  $SQT = \sum (y_i - \bar{y})^2$

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

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

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  $SQT$  que é explicada pelo fator  $A$ . Chama-se a soma dos quadrados devido a  $A$  e é denotado como  $SQA$ . Da mesma forma,  $SQB$  é  $2^2 q_B^2$  e  $SQAB$  (devido à interação  $AB$ ) é  $2^2 q_{AB}^2$ . Assim,  $SQT = SQA + SQB + SQAB$

Estas partes podem ser expressas como uma fração, por exemplo, Fração da variação explicada por  $A = SQA/SQT$ , 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  $SQT$  segue agora.

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

$$y_i = q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{AiBi}, i \in [1, 2^2] \tag{8.34}$$

As colunas  $x_A$ ,  $x_B$  e  $x_A x_B$  da matriz de design na Tabela 42 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:

$$\bar{y} = \frac{1}{4} \sum y_i = \frac{1}{4} \sum (q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{AiBi}) \tag{8.35}$$

$$\bar{y} = \frac{1}{4} \sum q_0 + \frac{1}{4} \sum q_A x_{Ai} + \frac{1}{4} \sum q_B x_{Bi} + \frac{1}{4} \sum q_{AB} x_{AiBi} \tag{8.36}$$

Variação Total:

$$SQT = \sum (y_i - \bar{y})^2 = \sum (q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{AiBi} - \bar{y})^2 \tag{8.37}$$

$$SQT = \sum (q_A x_{Ai})^2 + \sum (q_B x_{Bi})^2 + \sum (q_{AB} x_{AiBi})^2 + \text{termos do produto} \tag{8.38}$$

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

8.39

$$SQT=4 q_A^2+4 q_B^2+4 q_{AB}^2$$

8.40

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



Exemplo 8.12 No caso do estudo da memória cache

$$\bar{y} = 1/4(15+55+25+75) = 40$$
$$SQT = \sum (y_i - \bar{y})^2 = \sum (25^2 + 15^2 + 15^2 + 35^2) = 2100 \text{ ou}$$
$$SQT = 4 \times 20^2 + 4 \times 10^2 + 4 \times 5^2 = 2100$$

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 8.13 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 buffer
3. Comutação de circuitos foi usado em vez de comutação de pacote
4. Arbitragem aleatória foi usada em vez de round robin
5. *Interleaving* infinito de memória foi usada de modo que não havia contenção no banco de memória.

Um planejamento 2<sup>2</sup> fatorial foi utilizado. As atribuições dos símbolos é mostrado na Tabela 43. 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 44. Os efeitos, calculado usando o método de tabela de sinais, são apresentados na Tabela 45. A tabela também contém porcentagem de variação explicada.

Tabela 43 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 44 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 45 Efeitos médios para o Estudo da Rede de Interconexão

Parâmetros	Estimativa da Média			Variação Explicada (%)		
	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:

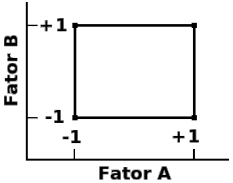
- A vazão média é de 0,5725. O rendimento é em grande parte afetada pelo padrão de referência, que faz uma diferença de  $\pm 0,1257$  e, portanto, explica 77% da sua variação. O tipo de rede contribui 0,0595 para a vazão. A Rede Omega é muito superior à média, e a Rede Crossbar é muito inferior à média. 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.

8.3 Experimento Fatorial 2<sup>k</sup>

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  $2^k$ , foi útil começar com o caso simples de apenas dois fatores ( $k = 2$ ). Este caso especial foi apresentado nas seções anteriores. Após o desenvolvimento deste caso, serão generalizados os conceitos para um número maior de fatores.

Os elementos conceituais de um experimento fatorial completo com somente 2 fatores:

Resposta	Variável R	
Fatores	Fator A (níveis A1, A2 ou níveis -1, +1) Fator B (níveis B1, B2 ou níveis -1, +1)	
Tratamentos	T1: A1 × B1 → R1 T2: A1 × B2 → R2 T3: A2 × B1 → R3 T4: A2 × B2 → R4	
Unidade Experimental	Estrutura para obtenção do resultado de cada tratamento	
Réplicas	Repetições dos tratamentos, preservando as condições operacionais	

Tratamento	Fator A	Fator A	Fator B	Fator B	Resposta
T1	A1	+1	B1	+1	R1
T2	A1	+1	B2	-1	R2
T3	A2	-1	B1	+1	R3
T4	A2	-1	B2	-1	R4

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 8.14 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 46. O projeto  $2^3$  e do desempenho medido em MIPS é mostrada na Tabela 47.

Tabela 46 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 47 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 48. 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 48 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 SQT pode ser calculado por meio dos efeitos da seguinte forma:

$SQT = 2^3(q_A^2 + q_B^2 + q_C^2 + q_{AB}^2 + q_{AC}^2 + q_{BC}^2 + q_{ABC}^2)$

$SQT = 8 (10^2 + 5^2 + 20^2 + 5^2 + 2^2 + 3^2 + 1^2)$

$SQT = 800 + 200 + 3200 + 200 + 32 + 72 + 8 = 4512$

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.

### 8.4 Exercícios

1. Demonstre que  $\sum (y_i - \bar{y})^2 = \sum (\hat{y}_i - \bar{y})^2 - \sum (y_i - \hat{y}_i)^2$
2. Quantos experimentos são necessárias para analisar o desempenho de um sistema que depende dos seguintes fatores:
  - CPU tipo: Intel, AMD, Apple
  - Tipo de sistema operacional: Windows, Linux, Android
  - Tipo de unidade de disco: Disco Rígido (HD), Disco Sólido (SSD)
3. Analise o desenho  $2^3$  mostrado na Tabela 49.
  - Quantifique os efeitos principais e as interações
  - Quantifique as porcentagens de variação explicada
  - Classifique as variáveis na ordem decrescente de importância

Tabela 49 Um projeto  $2^3$

Fator	A <sub>1</sub>		A <sub>2</sub>	
	C <sub>1</sub>	C <sub>2</sub>	C <sub>1</sub>	C <sub>2</sub>
B <sub>1</sub>	100	15	120	10
B <sub>2</sub>	40	30	20	50

4. Analise o desenho  $2^4$  mostrado na Tabela 50.
- 
- Quantifique os efeitos principais e as interações
  - Quantifique as porcentagens de variação explicada
  - Classifique as variáveis na ordem decrescente de importância

Tabela 50 Um projeto  $2^4$

Fator	A <sub>1</sub>				A <sub>2</sub>			
	C <sub>1</sub>		C <sub>2</sub>		C <sub>1</sub>		C <sub>2</sub>	
	D <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>2</sub>
B <sub>1</sub>	95	30	15	35	100	60	10	50
B <sub>2</sub>	50	10	35	20	20	30	60	40

5. Analise o desenho  $2^5$  mostrado na Tabela 51.
- 
- Quantifique os efeitos principais e as interações
  - Quantifique as porcentagens de variação explicada
  - Classifique as variáveis na ordem decrescente de importância

Tabela 51 Um projeto  $2^5$

Fator		A <sub>1</sub>				A <sub>2</sub>			
		C <sub>1</sub>		C <sub>2</sub>		C <sub>1</sub>		C <sub>2</sub>	
		D <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>2</sub>
B <sub>1</sub>	E1	104	32	16	32	93	56	10	49
	E2	154	30	15	35	100	60	10	50
B <sub>2</sub>	E1	48	9	35	19	20	27	58	43
	E2	102	24	22	29	71	48	26	47

## 9 Referências

Cisco. Cisco Annual Internet Report (2018-2023) White Paper. Disponível em <<https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>>. Acesso em: 20 Dez 2020.

Costa (2006)

Daemonio. Gerando combinações sem repetição pela contagem binária em C. Disponível em: <<https://daemoniolabs.wordpress.com/2011/02/17/gerando-combinacoes-sem-repeticao-pela-contagem-binaria-em-c/>>. Acesso em 18 Jun 2020.

Pakrashi, Arjun. All Combinations Without Repetitions. Disponível em: <<https://phoxis.org/2009/10/13/allcombgen/>>. Acesso em: 18 Jun 2020.

Fogliatti (2007)

Jain, Raj. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. Wiley- Interscience, New York, NY, 1991.

Tecnoblog. Tráfego de interconexão mundial vai triplicar para 8.200 terabits por segundo em 2021. Disponível em <<https://tecnoblog.net/260519/trafego-interconexao-mundial-2021/>>. Acesso em: 20 Dez 2020.

## 10Apendice

### 10.1 Programa C++ para calcular os efeitos e a variação em experimentos 2<sup>k</sup> fatoriais

O método combinar, do programa baixo, foi adaptado de Pakrashi(2020).

```
// -----
#include <algorithm>
#include <iostream>
#include <sstream>
#include <fstream>
#include <iomanip>
#include <string>
#include <vector>
#include <cmath>
// -----
using namespace std;
// -----
class clE2kF{
public:
    void processar( string, vector<double> );
private:
    int    k, col, lin;
    double SQT;
    string str;
    vector<double> y, T, q, V;
    vector<string> titulo;
    vector<vector<string>> tabela;

    void calcular( void );
    void combinar( void );
    void erro      ( void );
    void html      ( void );
    void listar    ( void );
    void ordenar   ( void );
    void salvar    ( string, string );
    void sinais    ( void );
};

void clE2kF::calcular( void ){
    for( int c = 0; c < col; c++ ){
        T[c] = 0.0;
        for( int l = 0; l < lin; l++ ){
            T[c] += stod(tabela[l][c]) * y[l];
        }
        q[c] = T[c]/lin;
    }
}
```



```

        SQT += lin*q[c]*q[c];
    }
    for( int c = 0; c < col; c++ ){
        V[c] = 100.0*lin*q[c]*q[c]/SQT;
    }
}

void clE2kF::combinar( void ){
    // Adaptado de: Pakrashi, Arjun. All Combinations Without Repetitions.
    // Disponível em: <https://phoxis.org/2009/10/13/allcombgen/>.
    // Acesso em: 18 Jun 2020.
    int l = 1;
    unsigned int m, x = ~(1 << k);
    while( l < lin ){
        int s = 0, c = 0;
        m = 1;
        string lista = "";
        while( m & x ){
            if( l & m ){
                lista += str[s];
                c++;
            }
            s++;
            m <<= 1;
        }
        l++;
        titulo.push_back(lista);
    }
}

void clE2kF::erro( void ){
    if( k < 2 ){
        cout << endl
            << "Erro" << endl
            << "-----" << endl
            << "0 número de fatores é insuficiente." << endl
            << "-----" << endl;
        exit(1);
    }
    if( y.size() != lin ){
        cout << endl
            << "Erro" << endl
            << "-----" << endl
            << "0 número de valores de y é inconsistente." << endl
            << "-----" << endl;
        exit(2);
    }
}

void clE2kF::html( void ){
    double TI = 0.0, SV = 0.0;;
    stringstream htm;
    htm << setprecision(2) << fixed
        << "<html>"
        << "<style>body { background-color: #F5FFFA}"
        << "    table{ background-color: #F0F8FF}</style>"
        << "<body>"
        << "<h3>Tabela de Sinais de Experimento 2<sup>" << k << "</sup> Fatorial</h3>"
        << "<table border='1' cellpadding='4' cellspacing='1'>";

    htm << "<tr><th>E<th>I";
    for( int c = 0; c < col; c++ )
        htm << "<th>" << titulo[c];
    htm << "<th>y";

    for( int l = 0; l < lin; l++ ){
        htm << "<tr><th>" << l+1 << "<td>" << 1;
        for( int c = 0; c < col; c++ ){
            htm << "<td>" << tabela[l][c];
        }
        htm << "<td>" << y[l] << endl;
    }

    for( int l = 0; l < lin; l++ ) TI += y[l];

    htm << "<tr><th>T<td>" << TI;
    for( int c = 0; c < col; c++ ) htm << "<td>" << T[c];
    htm << "<td>";

```

```

    htm << "<tr><th rowspan='2'>q<th>q<sub>0</sub>";
    for( int c = 0; c < col; c++ ) htm << "<th>q<sub>" << titulo[c] << "</sub>";
    htm << "<td>";

    htm << "<tr><td>" << TI/lin;
    for( int c = 0; c < col; c++ ) htm << "<td>" << q[c];
    htm << "<td>";

    htm << "<tr><th>V<td>";
    for( int c = 0; c < col; c++ ){
        htm << "<td>" << V[c];
        SV += V[c];
    }
    htm << "<td>" << SV
        << "</table><br>"
        << "SQT=" << SQT
        << "<br>"
        << "Legenda: T - Total, q - Efeito e V - Variação(%)<br>";
    salvar( "E2kF.out.html", htm.str() );
}

void clE2kF::listar( void ){
    cout << endl << endl
        << "-----" << endl
        << "    n | Combinações" << endl
        << "-----" << endl;
    for( int l = 0; l < titulo.size(); l++ ){
        cout.width(4);
        cout << l+1 << " | " << titulo[l] << endl;
    }
    cout << "-----" << endl;
}

void clE2kF::ordenar( void ){
    for( int l = 0; l < titulo.size(); l++ ){
        for( int c = l+1; c < titulo.size(); c++ ){
            if( titulo[l].size() > titulo[c].size() ){
                string s = titulo[l];
                titulo[l] = titulo[c];
                titulo[c] = s;
            }
        }
    }
    for( int l = 0; l < titulo.size(); l++ ){
        for( int c = l+1; c < titulo.size(); c++ ){
            if( titulo[l].size() < titulo[c].size() ) continue;
            if( titulo[l].compare(titulo[c]) > 0 ){
                string s = titulo[l];
                titulo[l] = titulo[c];
                titulo[c] = s;
            }
        }
    }
}

void clE2kF::processar( string str, vector<double> y ){
    this->str = str;
    this->y = y;
    k = str.size();
    lin = pow(2,k);
    col = lin-1;
    erro();
    titulo.clear();
    tabela.resize(lin,vector<string>(col));
    T.resize(col);
    q.resize(col);
    V.resize(col);
    SQT = 0.0;
    combinar();
    ordenar();
    listar();
    sinais();
    calcular();
    html();
}

void clE2kF::salvar( string fn, string str ){
    ofstream fo;
    fo.open(fn);
    replace(str.begin(), str.end(), '.', ',');
    fo << str;
}

```

```

    fo.close();
}
void clE2kF::sinais( void ){
    for( int l = 0; l < lin; l++ ){
        for( int c = 0; c < k; c++ ){
            int r = l / int(pow(2,k-c-1));
            if( r % 2 == 0 ) tabela[l][c] = "+1";
            else tabela[l][c] = "-1";
        }
        for( int c = k; c < col; c++ ){
            int s = 1;
            for( int p = 0; p < k; p++ ){
                if( titulo[c].find(str[p]) != string::npos ){
                    int r = l / int(pow(2,k-p-1));
                    if( r % 2 != 0 ) s *= -1;
                }
            }
            tabela[l][c] = s > 0 ? "+1" : "-1";
        }
    }
}
}
int main( void ){
    clE2kF Tabela;
    vector<double> y = { 1.0, 1.0, 1.0, 1.0, 1.8, 1.9, 1.7, 1.8, 4.8, 3.7, 5.0, 3.4, 4.8, 4.8, 8.5,
4.4 };
    Tabela.processar( "ABCD", y );
    return 0;
}

```

## 10.2 Programa html para calcular os efeitos e a variação em experimentos $2^k$ fatoriais

```

<!doctype html>
<html lang="pt-br">
<head>
    <meta charset="utf-8">
    <title>Efeitos e Variação em Experimentos Fatoriais</title>
    <meta name="description" content="Tabela de Sinais de Experimentos Fatoriais">
    <meta name="keywords" content="desempenho, estatística, sistema, computador">
    <meta name="author" content="Brauli G Leal -
http://www.univasf.edu.br/~brauli.leal">
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
</head>
<script>
function clTSE2kF(str,y){
    let k = str.length, titulo = [], tabela = [], T = [], q = [], V = [];
    let lin = Math.pow(2,k), col = lin-1, SQT = 0.0;

    this.calcular = function(){
        for( let c = 0; c < col; c++ ) T[c] = q[c] = V[c] = 0.0;

        for( let c = 0; c < col; c++ ){
            let s = 0.0;
            for( let l = 0; l < lin; l++ ){
                s += parseInt(tabela[l][c]) * y[l];
            }
            T[c] = s;
            q[c] = s/lin;
            SQT += lin*q[c]*q[c];
        }
        for( let c = 0; c < col; c++ ){
            V[c] = 100.0*lin*q[c]*q[c]/SQT;
        }
    }
    this.combinar = function(){
        let x, m, n, i, j;
    }
}

```

```

    x = ~(1 << k);
    n = 1;
    while( x & n ){
        let lista = "";
        m = 1;
        i = j = 0;
        while( m & m ){
            if( n & m ){
                lista += str[j] ;
                i++ ;
            }
            j++;
            m = m << 1 ;
        }
        titulo.push(lista);
        n++;
    }
}
// Gerando combinações sem repetição pela contagem binária em C
// https://daemoniolabs.wordpress.com/2011/02/17/gerando-combinacoes-sem-repeticao-pela-
// contagem-binaria-em-c/
// 18 jun 2020
}
this.html = function(){
    let htm, TI = 0.0;
    htm = "<br>"
        + "Efeitos(E) e Variação(V) de Experimento 2<sup>" + k + "</sup> Fatorial"
        + "<table border='1' cellpadding='4' cellspacing='1'>";
    htm += "<tr><th>E<td>I";
    for( let c = 0; c < titulo.length; c++ )
        htm += "<th>" + titulo[c];
    htm += "<th>y";
    for( let l = 0; l < tabela.length; l++ ){
        htm += "<tr><th>" + (l+1) + "<td>1";
        for( let c = 0; c < tabela[l].length; c++ ){
            htm += "<td>" + tabela[l][c];
        }
        htm += "<td>" + y[l];
    }

    for( let l = 0; l < lin; l++ ) TI += y[l];

    htm += "<tr><th>T<td>" + TI.toFixed(2);
    for( let c = 0; c < col; c++ ) htm += "<td>" + T[c].toFixed(2);
    htm += "<td>";

    htm += "<tr><th rowspan='2'>q<td>q<sub>0</sub>";
    for( let c = 0; c < col; c++ ) htm += "<td>q<sub>" + titulo[c] + "</sub>";
    htm += "<td>";

    htm += "<tr><td>" + (TI/lin).toFixed(2);
    for( let c = 0; c < col; c++ ) htm += "<td>" + q[c].toFixed(2);
    htm += "<td>";

    htm += "<tr><th>V<td>";
    for( let c = 0; c < col; c++ ) htm += "<td>" + V[c].toFixed(2);
    htm += "<td>"
        + "</table><br>";
    + "SQT=" + SQT.toFixed(2);
    + "<br><br>";
    + "T - Total, q - Efeito e V - Variação(<br>";
    return htm.replaceAll('.', ',');
}
this.ordenar = function(){
    for( let l = 0; l < titulo.length; l++ ){
        for( let c = l+1; c < titulo.length; c++ ){
            if( titulo[l].length > titulo[c].length ){
                let str = titulo[l];
                titulo[l] = titulo[c];
                titulo[c] = str;
            }
        }
    }
}

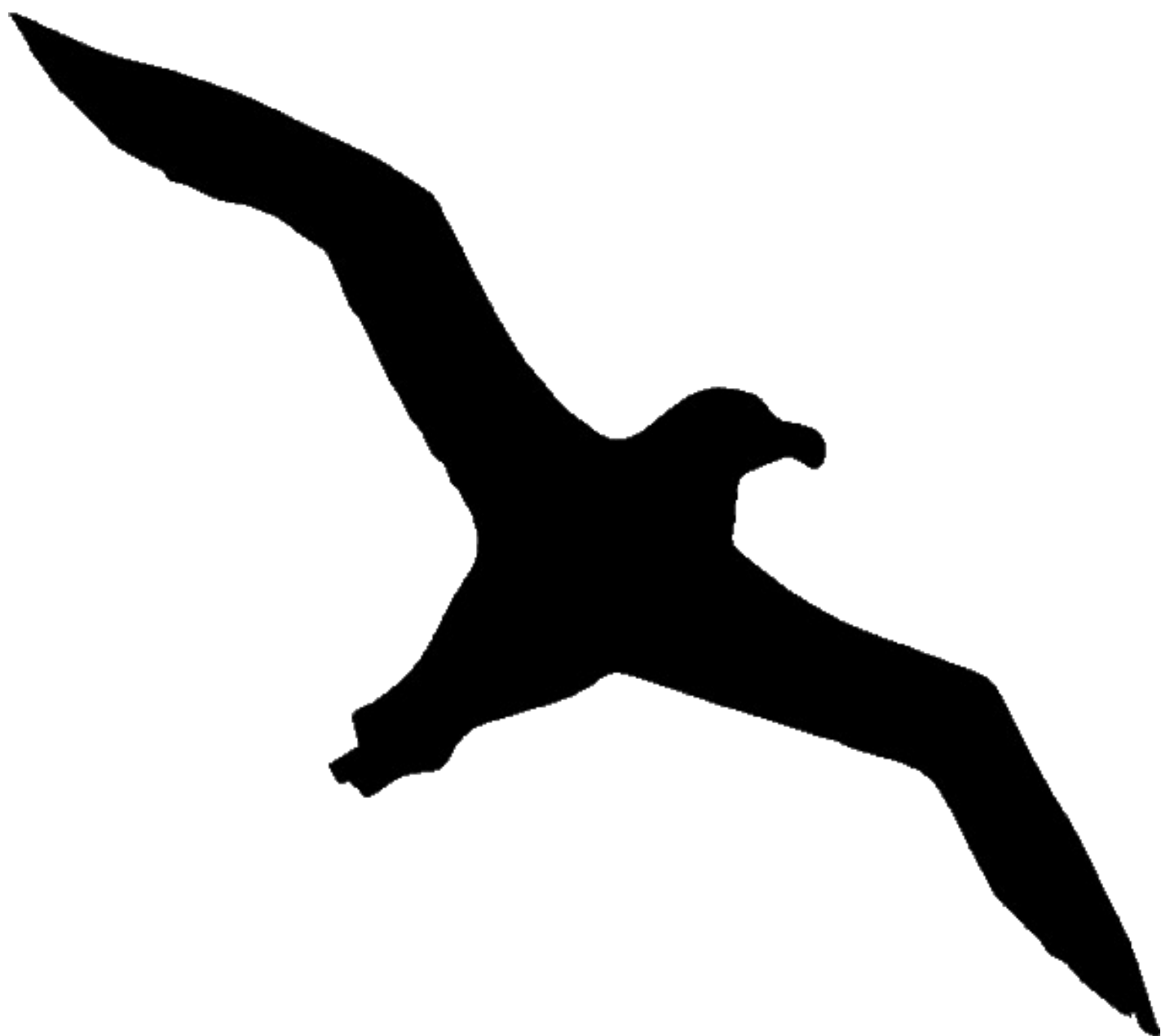
```

```

        }
        else if( (titulo[l].length === titulo[c].length) && (titulo[l] > titulo[c]) )
    {
        let str = titulo[l];
        titulo[l] = titulo[c];
        titulo[c] = str;
    }
    }
}
this.processar = function(){
    this.combinar();
    this.ordenar();
    this.sinais();
    this.calcular();
    return this.html();
}
this.sinais = function(){
    for( let l = 0; l < lin; l++ ) tabela[l] = new Array(col);
    for( let l = 0; l < lin; l++ ){
        for( let c = 0; c < k; c++ ){
            let r = l / Math.pow(2,k-c-1);
            if( parseInt(r) % 2 === 0 ) tabela[l][c] = "+1";
            else tabela[l][c] = "-1";
        }
        for( let c = k; c < col; c++ ){
            let s = 1;
            for( let p = 0; p < k; p++ ){
                if( titulo[c].indexOf(str[p]) >= 0 ){
                    let r = l / Math.pow(2,k-p-1);
                    if( parseInt(r) % 2 !== 0 ) s *= -1;
                }
            }
            tabela[l][c] = s > 0 ? "+1" : "-1";
        }
    }
}
}
}
function processar(){
    let str = document.getElementById('str').value,
        y = JSON.parse(document.getElementById('y').value);
    y = JSON.parse("[34.5,33.2,37.1,35.8,31.6,30.8,33.3,32.5]");
    TS = new clTSE2kF(str,y);
    document.getElementById("tabela").innerHTML = TS.processar();
}
function exemplo1(){
    let str = "ABC",
        y = JSON.parse( "[ 1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.6, 8.8 ]" );
    document.getElementById('str').value = str;
    TS = new clTSE2kF(str,y);
    document.getElementById("tabela").innerHTML = TS.processar();
}
function exemplo2(){
    let str = "ABCD",
        y = JSON.parse( "[ 1.0, 1.0, 1.0, 1.0, 1.8, 1.9, 1.7, 1.8, 4.8, 3.7, 5.0, 3.4, 4.8, 4.8, 8.5, 4.4 ]" );
    document.getElementById('str').value = str;
    document.getElementById('y').value = "[ 1.0, 1.0, 1.0, 1.0, 1.8, 1.9, 1.7, 1.8, 4.8, 3.7, 5.0, 3.4, 4.8, 4.8, 8.5, 4.4 ]";
    TS = new clTSE2kF(str,y);
    document.getElementById("tabela").innerHTML = TS.processar();
}
window.onload = exemplo2;
</script>
</head>
<body>
    <div>
        <h3>Tabela de Sinais</h3>

```

```
<hr>
  <input type="text" id="str" name="str" value="ABC" size="15"/>
  y <input type="text" id="y" name="y" value="[ 1.1, 2.2, 3.3, 4.4, 5.5, 6.6, 7.6, 8.8
]" size="60"/>
  <input type="button" value=">>" onclick="processar()"/>
  <hr>
</div>
<div id="tabela"></div>
</body>
</html>
```



Fonte: <https://image.flaticon.com/icons/png/512/47/47206.png>

Os albatrozes são aves marinhas de grandes dimensões que se movem de forma muito eficiente no ar, cobrindo grandes distâncias com pouco esforço. São extremamente adaptadas à vida em alto mar, retornam à terra somente para se reproduzirem.