Avaliação de desempenho da política EBS em uma arquitetura de escalonamento realimentada

Alessandro Nakamuta

Avaliação de desempenho da política EBS em uma arquitetura de escalonamento realimentada

Alessandro Nakamuta

Orientador: **Prof. Dr. Francisco José Monaco**

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

USP – São Carlos
Junho de 2012

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

Nakamuta, Alessandro
N163a Avaliação de eesempenho da política EBS em uma arquitetura de escalonamento realimentada / Alessandro Nakamuta; orientador Francisco José Monaco. -- São Carlos, 2012.
71 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2012.

1. Tempo-Real. 2. QoS. 3. Escalonador. I. Monaco, Francisco José, orient. II. Título.

Agradecimentos

Agradeço à toda minha família, especialmente meus avós, Mitiko e Hino, meus pais, Mario e Rute e minha irmã Jacqueline pelo amor e apoio na vinda à São Carlos.

À minha namorada Jucimara, pelo seu amor, compreensão e paciência nos momentos ausentes. Te amo.

Ao meu orientador e professor Monaco pela sua motivação, orientação e por ter me dado a oportunidade, sem o qual esse trabalho não seria possível.

Aos professores da UFMS e do ICMC , que ajudaram a contribuir com a minha formação, em especial Kalinka, Marcos, Paulo Sérgio, Regina, Said, Sarita, Sotoma e Turine.

Aos meus irmãos por parte de orientador Edwin, Lourenço, Michelle, Pedro Nobile, Priscila e Renê que me ajudaram com idéias e sugestões que fazem parte deste trabalho.

Aos grandes amigos de Campo Grande e da república Tereré Diogo, Gondim, Kenji, Letrícia, Mario e Patrick pelo companherismo.

Aos "primos" da república Tibilisku Bruno Guazzelli, Bruno Tardiole, Daniel, João Paulo e Paulão, por me acolherem da melhor maneira possível na reta final da minha dissertação.

Aos pessoal do LASDPC pelas conversas e descontração Bruno Faiçal, Dionísio, Douglas, Edvard, Elvis, Fausto, Júlio, Luís, Maycon, Pedro, Ricardo, Rayner e Roni.

À todos que de alguma forma participaram na realização deste trabalho.

À CNPq pelo apoio financeiro.

Resumo

Este trabalho apresenta uma avaliação do algoritmo EBS, uma política de escalonamento proposta para sistemas de tempo real flexíveis com qualidade de serviço baseado em limites superiores para tempos médios de resposta. Experimentos têm demonstrado propriedades vantajosas da política EBS em servidores *Web* com diferenciação de serviço. O objetivo do presente estudo é compreender o comportamento da *EBS* em relação à diferentes parâmetros que descrevem a carga de trabalho. Esse conhecimento é útil para obtenção de um melhor aproveitamento computacional. São apresentados experimentos e resultados que analisam a influência de cada um dos fatores considerados na qualidade do serviço oferecido. A partir desses resultados são tecidas conclusões acerca de abordagens para o dimensionamento de carga e de capacidade do servidor.

Abstract

This Master degree project has presented an evaluation of the EBS algorithm, a scheduling policy proposed for soft real-time systems with quality of service based on upper limits for average response times. Experiments have shown advantageous properties of the EBS policy on Web servers with service differentiation. The aim of this study is to understand the behavior of the EBS in relation to different parameters that describe the workload. This knowledge is useful for obtaining a better use of computing. Experiments and results are presented analyzing the influence of each factor considering the quality of service offered. From these results, conclusions are woven about approaches to the design load and server capacity.

Sumário

Agradecimentos **i**

Resumo **iii**

Abstract **v**

Lista de Siglas **xiii**

1 Introdução **1**
1.1 Contextualização 1
1.2 Motivação e Objetivos 4
1.3 Organização do Documento 5

2 Revisão **7**
2.1 Qualidade de Serviço 8
2.1.1 Arquiteturas de QoS 9
2.1.2 QoS em Nível de Aplicação 10
2.2 Sistemas de Tempo Real 13
2.2.1 Sistemas de Tempo Real 14
2.2.2 Escalonamento em Sistemas RT 16
2.3 Simulação de Sistemas 21
2.3.1 Modelos de simulação 22
2.3.2 Simulação de Filas 23
2.3.3 Linguagens de Simulação 26
2.3.4 Probabilidade e Estatística para Simulação 26
2.4 2^K Fatorial Completo 28
2.5 EBS: *Exigency Based Scheduling* 30
2.5.1 Política de escalonamento EBS 30
2.5.2 Ambiente de Simulação 32
2.5.3 Trabalhos Relacionados 33

3 Resultados **37**
3.1 Metodologia de Desenvolvimento 38
3.2 Resultados 40
3.2.1 Análise da influência 40

3.2.2 Análise da influência fator a fator 43
3.3 Notas finais 54

4 Conclusão **57**
4.1 Contribuições 58
4.2 Trabalhos Futuros 58

A Tabelas **61**

Lista de Figuras

2.1 Ilustração das Restrições Temporais. 15
2.2 Caracterização das tarefas: (a) periódica; (b) esporádica e (c) aperiodica. 16
2.3 Exemplo da Utilização do Algoritmo de Escalonamento RM 19
2.4 Exemplo da Utilização do Algoritmo de Escalonamento DM 20
2.5 Exemplo da Utilização do Algoritmo de Escalonamento EDF 21
2.6 Elementos de uma fila 24
2.7 Representação de centros de serviços 24
2.8 Modelos de Redes de Filas: (a) Aberto, (b) Fechado e (c) Misto. 25

3.1 Modelo de servidor sequencial 38
3.2 Gráfico da influência dos fatores. 42
3.3 Gráfico da influência dos fatores em um ambiente com maior taxa de utilização . . 43
3.4 Influência no tempo de resposta com a variação da taxa de utilização 45
3.5 Influência na satisfação dos usuários com a variação da taxa de utilização 46
3.6 Influência na dispersão da satisfação dos usuários com a variação da taxa de utilização 47
3.7 Influência no tempo de resposta com a variação do número de contratos 48
3.8 Influência na satisfação dos usuários com a variação do número de contratos 49
3.9 Influência na satisfação dos usuários com a variação do número de contratos 49
3.10 Influência no tempo de resposta com a variação da média dos contratos 50
3.11 Influência na satisfação dos usuários com a variação da média dos contratos 51
3.12 Influência na variação da satisfação dos usuários com a variação da média dos contratos 51
3.13 Influência no tempo de resposta com a variação da dispersão dos contratos 52
3.14 Influência na satisfação dos usuários com a variação da dispersão dos contratos . . 53
3.15 Influência na variação da satisfação com a variação da dispersão dos contratos . . . 53

Lista de Tabelas

2.1 Tabela com exemplo para o algoritmo RM 18
2.2 Tabela com exemplo para o algoritmo DM 19

3.1 Experimentos 41
3.2 Experimentos 43
3.3 Experimentos com mudança da taxa de utilização 44
3.4 Experimentos com mudança no número de contratos 48
3.5 Experimentos com mudança na média dos contratos 50
3.6 Experimentos com mudança na dispersão dos contratos 52
3.7 Influências positivas e negativas 53
3.8 Variação das influências no tempo médio de resposta 54
3.9 Variação das influências na satisfação dos usuários 54
3.10 Variação das influências na dispersão das satisfações 54

A.1 Resultados 61
A.2 Influência (2^k fatorial completo) 62
A.3 Resultados 62
A.4 Influência (2^k fatorial completo) 63
A.5 Influência do número de contratos com variação da taxa de utilização 63
A.6 Influência da média dos contratos com variação da taxa de utilização 63
A.7 Influência da dispersão dos contratos com variação da taxa de utilização 63
A.8 Influência da taxa de utilização com variação do número de contratos 64
A.9 Influência da média dos contratos com variação do número de contratos 64
A.10 Influência da dispersão dos contratos com variação do número de contratos 64
A.11 Influência da taxa de utilização com variação do número de contratos 64
A.12 Influência do número de contratos com variação do número de contratos 64
A.13 Influência da dispersão dos contratos com variação do número de contratos 65
A.14 Influência da taxa de utilização com variação da dispersão dos contratos 65
A.15 Influência do número de contratos com variação da dispersão dos contratos 65
A.16 Influência do número de contratos com variação da dispersão dos contratos 65

Lista de Siglas

DiffServ - Differentiated services

DM - Deadline Monotonic

DMR - Deadline Miss Ratio

DS Field - Differentiated Service Field

EBS - Exigency Based Scheduler

EDF - Earliest Deadline First

FCFS - First Come First Server

FIFO - First In First Out

IETF - Internet Engineering Task Force

IntServ - Integrated services

ISO - International Organization for Standardization

ISP - Internet Service Provider

IP - Internet Protocol

LCFS - Last Come First Server

PRIAdap - Prioridades Adaptativo

OSI - Open System Interconnection

QoS - Quality of Service

RM - Rate Monotonic

RSVAdap - Reserva Adaptativa de Recursos

RSVP - Resource ReSerVation Protocol

RT - Real-Time

SFD - Short Flow Differentiating

SJF - Shortest Job First

SLA - Service Level Agreement

SOA - Service-Oriented Architecture

SWDS - Servidor Web com Diferenciação de Serviços

TOS - Type of Service

WFQ - Weighted Fair Queuing

CAPÍTULO

1

Introdução

1.1 Contextualização

À medida que sistemas computacionais vão sendo integrados nos mais diversos serviços orientados ao usuário, amplia-se a gama de requisitos operacionais relevantes ao seu desempenho e confiabilidade. Requisitos temporais de responsividade, dentre esses, são importantes em sistemas interativos, como por exemplo, telemedicina, sistemas de aviação, comércio eletrônico, voz sobre IP (Internet Protocol), jogos online, IPTV[1], entre outros. Essas aplicações demandam abordagens de análise e síntese pertinentes ao domínio dos sistemas de tempo real (RT(Real-Time)).

Sistemas RT são aplicáveis quando eventos devem ser tratados com tempo de resposta máximos pré-definidos, chamado de prazo de resposta. O não cumprimento desse prazo é chamado de falta. É importante ressaltar que esse requisito não está ligado à velocidade, e sim ao cumprimento dos prazos estabelecidos. Em aplicações de tempo real existe a premissa que o sistema precisa se sincronizar com o ambiente; caso contrário, a aplicação não é de tempo real, e é chamado de aplicação de tempo virtual.

Os sistemas RT são classificados conforme a gravidade da consequência a que uma falta pode acarretar. Sistemas onde é intolerável a ocorrência de faltas são chamados de *Hard-RT*, ou sistemas de tempo real rígidos. Nesses sistemas a violação de um único prazo pode danificar equipamentos, machucar pessoas, destruir dados, dessincronizar sistemas, ou qualquer outra consequência de natureza não recuperável. Sistemas onde é tolerável a ocorrência de faltas são chamados de tempo real flexível. Nesses sistemas as violações podem ocorrer em número e frequência, dependendo

[1] ou TVIP é um sistema através da qual o serviço de televisão digital é disponibilizado usando métodos de arquitetura e rede IP.

da aplicação. A violação dos prazos degrada o sistema, mas de forma recuperável ou aceitável. Existem duas vertentes nesse tipo de sistema, o *Soft-RT* e *Firm-RT*. Sistemas *Soft-RT* podem ou não utilizar os dados atrasados, já nos sistemas *Firm-RT* não há possibilidade ou necessidade de utilização desses dados.

Em sistemas *Soft-RT* e *Firm-RT*, a tentativa de cumprimento de requisito de tempo real pode ser associado ao conceito de qualidade de serviço (QoS (Quality of Service)). QoS se refere à capacidade dos elementos de um sistema prover garantias acerca de determinados parâmetros associados à percepção da qualidade de um serviço oferecido. Exige-se que tais parâmetros permaneçam dentro de limites bem definidos. No campo da telecomunicação e redes de computadores, são importantes quatro parâmetros: confiabilidade, retardo, flutuação e largura de banda (Tanenbaum, 2003).

Grande parte das contribuições oriundas da área de redes de comunicação de dados para provisão de QoS é desenvolvida no nível de rede, tendo como referência o modelo OSI(Open System Interconnection.). Os projetos para Serviços Integrados (*IntServ* (Integrated services)) e Serviços Diferenciados (*DiffServ* (Differentiated services)) fazem parte dessas contribuições. Porém, para que a QoS seja melhor implementada, é conveniente que ela esteja presente em todos os níveis. QoS no nível de aplicação se mostra interessante para evitar descartes indistintos dos servidores *Web* que poderiam prejudicar a atuação da QoS no nível de rede (Teixeira et al., 2005).

Em relação aos serviços com requisitos de tempo real flexível (sistemas *Soft-RT* ou *Firm-RT*), a métrica convencional para avaliação da QoS é a taxa de faltas (DMR (Deadline Miss Ratio)). Na sua forma mais simples, requisitos de qualidade podem ser declarados como limites superiores para o DMR, o que é suficiente para quantificar a confiabilidade do sistema em relação à taxa de faltas do serviço, bem como a eficiência de transferência efetiva com respeito à repetição de requisições (por exemplo, retransmissão de pacotes em um mecanismo de comunicação confiável). Isto, porém, não mede a distribuição dos atrasos nos tempos de resposta, em outras palavras, só indica que ocorreram atrasos, mas não indica o quão grave foram os atrasos.

Uma métrica alternativa que relaciona em uma única medida os tempos de serviço com sua frequência de ocorrência é o tempo médio de resposta (ART (Average Response Time)) (Monaco et al., 2009). Como um parâmetro do contrato de serviço (SLA (Service Level Agreement)), o tempo médio de resposta pode ser significativo em muitas circunstâncias em que as restrições sobre as métricas de desempenho global são relevantes. Este é o caso, por exemplo, de um sistema *soft-RT* com um *buffer*[2] finito, onde pode não ser necessário que o serviço atenda a fila de requisições em uma taxa constante em uma operação *hard-RT*; neste caso basta o tempo médio de resposta ser delimitada superiormente por um valor que impeça que o *buffer* esvazie durante o processo.

Trabalhos substanciais na área de sistemas RT produziram importantes resultados teóricos visando a análise de aplicações com restrições determinísticas de tempo. Processos caracterizados por uma dinâmica determinística de tempo são aperiódicos, porém possuem uma média de ope-

[2]região da memória utilizada temporariamente para armazenar dados enquanto estiver processando outros.

rações por determinado período de tempo. Processos caracterizados por uma dinâmica periódica, presentes comumente em sistemas de monitoramento e controle, como no ambiente de automação industrial, têm motivado o desenvolvimento de abordagens analíticas. O tratamento de sistemas orientados a evento (assíncronos), onde requisições com chegadas e tempo de execução não são determinísticas, são por outro lado consideravelmente mais complexas. Alocação de recursos para sistemas RT em dinâmicas não determinísticas é reconhecidamente desafiador mesmo com técnicas do estado-da-arte. Abordagens heurísticas, por outro lado, prevalecem neste domínio (Casagrande, 2007).

Alternativamente, um conceito que tem emergido no campo da computação de tempo real é o conceito de controle de realimentação (*feedback control*), já estabelecido em outras áreas da Engenharia. Ele se baseia no princípio de auto-adaptação usando o desvio da saída do sistema em relação a um valor desejado. O valor desse desvio é usado como entrada no sistema, realimentando-o, de tal forma que ocorra um ajuste na saída do sistema diminuindo o desvio, e assim fornecendo a saída desejada. O sistema é sempre orientado através do valor de saída e por isso é menos suscetível a variações de parâmetros internos e distúrbios externos.

Um elemento que tem grande influência no desempenho de serviços interativos é o escalonador de processos. O escalonador é responsável por organizar a ordem de atendimento dos processos. A fim de satisfazer restrições de tempo real, políticas de escalonamento apropriadas devem ser aplicadas para gerenciar convenientemente a alocação de recursos. Assim, políticas de escalonamento são um tópico importante para provisão de QoS.

Garantias Estocásticas de Responsividade

Dentre as possíveis especificações de um sistema *soft-RT*, a garantia de limites superiores de tempo de resposta constitui um problema de interesse no contexto da provisão de QoS em computação orientada a serviço. Para um conjunto de classes de serviço, o objetivo é garantir que o tempo médio de resposta calculado sobre uma janela com w requisições passadas para cada cliente do sistema seja limitado superiormente por um valor acordado com base em cada classe. Intuitivamente, se um cliente particular for recorrentemente deixado de lado em diversos ciclos de escalonamento consecutivos, o ART efetivo tenderá a aumentar. Uma abordagem sensata é, então, tomar a diferença entre os contratos estabelecidos e o ART efetivo em consideração quando ocorre a definição de prioridades para a alocação de recursos.

Resultados clássicos da teoria de tempo real têm produzido conhecidos algoritmos de escalonamento (Sha et al., 2004), entre os quais, a EDF (Earliest Deadline First) é uma importante contribuição. A EDF é uma disciplina de escalonamento ótimo para sistemas monoprocessados não-preemptivo que atribui as maiores prioridades para as requisições com os *deadlines* (limite de tempo) mais restritos. Pode parecer que essa heurística baseada em urgência é uma solução simples para o problema levantado, e que é razoável servir primeiro os clientes cujos contratos estão mais perto de uma violação. No entanto, um exame teórico mais cuidadoso à luz da teoria de

controle, revelou que este não é o caso. Se a diferença entre contratos e o ART efetivo é injetado no sistema como um sinal de *feedback* (retorno), e este é o único fator influenciando a decisão do escalonamento, então essa diferença (o erro) será minimizada.

A política *Exigency Based Scheduler* (EBS) (Casagrande, 2007) é uma nova proposta de política de escalonamento para provisão de tempo real, desenvolvida junto ao Grupo de Sistemas Distribuídos e Programação Concorrente (GSDPC-ICMC-USP), que tem por objetivo a provisão de garantias de QoS com limites superiores de tempo médio de resposta em sistemas *soft-RT*. Esses limites de tempo são os chamados contratos, e é definido previamente com um acordo entre o provedor de serviço e o contratante. A EBS faz um gerenciamento entre as requisições, de modo a cumprir a restrição temporal associada a cada um deles, atribuindo as prioridades mais altas para as requisições com o menor produto dado pela Equação 1.1.

$$P = D \cdot C \tag{1.1}$$

Onde D é o *deadline*, C é o tempo esperado de execução e P é a prioridade atribuída a requisição. É ponderada a urgência do pedido com o tempo de execução emprestado do algoritmo SJF (Shortest Job First). SJF prioriza as requisições com o menor tempo de processamento e é conhecida por minimizar o tempo médio de resposta.

A lógica da EBS é priorizar requisições com *deadlines* apertados, mas somente se eles não têm tempo de processamento custoso. A EBS demonstrou a propriedade de garantir um equilíbrio justo na alocação de recursos proporcional às demandas impostas por cada classe de serviço (Casagrande, 2007). Ela atua como bloco de controle de uma arquitetura auto-adaptativa de gerenciamento de recursos, pois a cada requisição atendida é calculado um novo *deadline* de modo que é feito o possível para que o tempo médio total do usuário não ultrapasse o tempo contratado. O tempo de atendimento da requisição atual é computado na média total do usuário realimentando assim o sistema.

1.2 Motivação e Objetivos

O desempenho da EBS no atendimento às especificações estocásticas de responsividade temporal tem se mostrado superior às alternativas convencionais em diversos cenários (Monaco e Nobile, 2009). Nesses testes foram escolhidos contratos arbitrários para os usuários, somente com a preocupação de não escolher contratos muito baixos, que seriam impossíveis para qualquer escalonador cumprir. Para atender bem o cliente, muitas vezes os provedores de serviços disponibilizam mais recursos que o necessário, devido tanto à oscilação do ambiente, como para ter uma margem de segurança. Porém o provedor também deseja minimizar essa margem de segurança de forma confiável, assim minimizando, à medida do possível, o custo total do sistema.

Assim se torna necessário um estudo de como o sistema se comporta às mudanças do ambiente, como por exemplo, o comportamento do sistema quando há um aumento do número de usuários.

Com essa informação o provedor de serviços poderá se preparar para uma eventual mudança nesse fator, podendo manter a satisfação dos clientes sem a necessidade de superdimensionar o sistema. Essa análise poderá ser utilizada para prever a necessidade de recursos e seu redimensionamento mais eficiente.

O objetivo do presente trabalho é o estudo e a análise de resultados do algoritmo de escalonamento EBS, a fim de prever, de forma qualitativa, o comportamento do sistema, de acordo com a variação dos fatores de entrada. Neste trabalho foram analisados quatro fatores de entrada:

- Taxa de Utilização do Sistema
- Média dos Contratos
- Número de Contratos
- Dispersão dos Contratos

E três fatores de saída:

- Tempo Médio de Resposta
- Satisfação dos Usuários
- Variação da Satisfação entre os Usuários

Esse estudo oferece uma contribuição na área de qualidade de serviço em nível de aplicação, para sistemas computacionais de tempo real, ampliando os resultados já obtidos nessa linha de pesquisa, visando à eficiência do uso de recursos computacionais.

1.3 Organização do Documento

Este trabalho de pesquisa apresenta a seguinte organização:

- No presente capítulo foram apresentadas considerações iniciais, a motivação para o desenvolvimento do trabalho, assim como os objetivos pertinentes para seu desenvolvimento e a organização do documento.
- No Capítulo 2 são introduzidos os conceitos teóricos necessários ao desenvolvimento deste trabalho e a revisão de trabalhos relacionados.
- No Capítulo 3 são apresentados o plano de desenvolvimento, os experimentos realizados, e os resultados obtidos.
- No Capítulo 4 são apresentadas as principais conclusões.

- No Apêndice A são apresentadas as tabelas com valores numéricos para vários gráficos exibidos nesse trabalho.

- E por fim as referências bibliográficas.

CAPÍTULO

2

Revisão

Neste capítulo são apresentados os conceitos de qualidade de serviço, sistemas de tempo real, simulação de sistemas, assim como a apresentação do escalonador EBS. Também é explicado o método fatorial completo, que é usado para calcular as influências dos fatores de entrada no sistema. Este capítulo está organizado em seções.

Na Seção 2.1 são introduzidos os conceitos de qualidade de serviço. Em seguida, aborda-se a QoS em nível de rede, descrevendo as arquiteturas de serviços integrados e de serviços diferenciados. Por fim, destaca-se a importância da QoS em nível de aplicação. Além disso, são abordadas algumas pesquisas realizadas sobre Qualidade de Serviço, evidenciando os esforços realizados por diversos pesquisadores nesse campo.

Na Seção 2.2 é apresentada a fundamentação teórica que embasa sistemas de tempo real, definindo suas principais classificações quanto ao sistema e quanto ao escalonamento, apresentando alguns algoritmos difundidos nessa área.

Na Seção 2.3 é introduzido o conceito de simulação de sistemas, assim como a técnica para análise do modelo e da política de escalonamento propostos, descrevendo a teoria a partir da qual ela é fundamentada, bem como conceitos básicos de estatística e probabilidade, familiarizando o leitor sobre a utilização de simulação de sistemas.

Na Seção 2.4 é apresentado resumidamente o método fatorial completo, que calcula a influência dos fatores de entrada nas variáveis de resposta em um determinado ambiente.

E na Seção 2.5 são apresentados os conceitos e a metodologia do algoritmo EBS. É explicado seu funcionamento, o ambiente de simulação usado para a análise e avaliação do seu desempenho e trabalhos derivados.

2.1 Qualidade de Serviço

No modelo geral das redes baseadas em IP, como a *Internet*, as requisições que nela trafegam normalmente são tratadas seguindo a política do melhor esforço (*best-effort*). Isso significa que as requisições não possuem uma taxa de bits e uma taxa de entrega especificada, dependendo do tráfego corrente na rede (Sheldon, 2001). Essas requisições são tratadas, de forma geral, de modo equivalente, ou seja, a entrega dos dados é realizada de modo igual a todos os usuários.

Na ocorrência de congestionamento, os pacotes de dados podem sofrer atrasos ou até serem descartados. Os descartes são feitos sem distinção, o que não garante que o serviço seja bem sucedido, ou mesmo tenha um bom desempenho (Vasiliou, 2000). Superdimensionar a capacidade da rede, facilitando o trafego de pacotes, pode ser uma forma de tratar o problema. Porém essa abordagem, geralmente, exige custos altos que podem pesar contra a aplicação desse método (Tanenbaum, 2003).

Quando a *Internet* era usada somente por professores, estudantes e aplicações não comerciais, QoS não era um fator importante. Contudo, quando a *Internet* se tornou uma plataforma usada por empresas e provedores de serviço, a falta da Qos não pode mais ser negligenciada (Jajszczyk, 2008). Somado com o aumento no volume de tráfego de dados, o modelo de melhor esforço não atende as necessidades dos usuários (Silva et al., 2006). Portanto, a solução é inserir certo nível de inteligência (i.e complexidade) na *Internet*, de modo que diferencie, por exemplo, tráfegos com requisitos estritos de tempo daqueles que suportam atrasos ou perdas. Esse controle de tráfego é implementado por mecanismos de QoS.

QoS pode ser definida de várias maneiras na literatura, de acordo com suas aplicações. Formalmente, a ISO[1] define QoS como efeito coletivo do desempenho de um serviço, o qual determina o grau de satisfação de um usuário (ISO/IEC, 1998). Essa é uma definição genérica e deve ser especificada para o problema que se deseja tratar.

No contexto das redes de computadores, QoS pode ser definida como um controle de reserva de recursos, afim de prover diferentes prioridades a diferentes aplicações, usuários ou fluxo de dados (Marchese, 2007). Dentre suas principais formulações estão: garantia de desempenho e diferenciação de serviço. A primeira está diretamente relacionada com a capacidade de banda, atraso, *jitter*[2] e perda de pacotes; já a segunda, por sua vez, refere-se a oferecer diferentes níveis de prioridade a diferentes aplicações, cada qual com requisitos distintos.

[1] International Organization for Standardization.
[2] Variação de atrasos.

2.1.1 Arquiteturas de QoS

Dentre as várias arquiteturas existentes para provisão de QoS na *Internet* que atuam no nível de rede, dois modelos propostos pela IETF[3] se destacam: Serviços Integrados (*IntServ*) (Braden et al., 1994) e Serviços Diferenciados (*DiffServ*) (Blake et al., 1998).

O modelo *IntServ* é caracterizado pela reserva de recurso, onde utiliza-se um protocolo de sinalização que estabelece um caminho entre dois sistemas finais e realiza a reserva de recurso ao longo dele. Já o *DiffServ* se caracteriza pela marcação dos pacotes de acordo com classes de serviços pré-determinadas e comutados entre o domínio de rede, segundo contratos estabelecidos entre o cliente e o provedor.

Serviços Integrados

A arquitetura de Serviços Integrados é caracterizada pela reserva de recursos e pelo estabelecimento da chamada. Antes do início de uma comunicação, cada roteador que compõe o caminho entre a fonte e o destino deve estar apto a reservar recursos suficientes para garantir as exigências de QoS. Esses recursos podem ser: disponibilidade de *buffers*, capacidade de banda de *enlace*[4] e tempo em que a conexão será mantida.

Para tanto, tais aplicações utilizam o protocolo RSVP[5], um protocolo de controle e sinalização. O RSVP verifica se os nós intermediários suportam a QoS desejada, reservando uma quantidade necessária para a aplicação. Se algum roteador intermediário rejeitar a requisição, uma mensagem de erro é enviada de volta ao emissor. Caso contrário, os recursos necessários para o fluxo são alocados e as informações de estado do fluxo de dados são armazenadas no roteador (Zhao et al., 2000).

No entanto, com o aumento de fluxos, aumenta-se, consideravelmente, a quantidade de informações de estado, devido à reserva de recursos ser realizada para cada fluxo individualmente. Sendo assim, é necessário um elevado poder de processamento dos roteadores, tornando o núcleo da rede mais complexo e, consequentemente prejudicando a escalabilidade da *Internet*. Além disso, todos os roteadores ao longo da rota devem oferecer suporte a serviços integrados. Tais problemas acabam dificultando a implantação desse tipo de abordagem. De forma a solucionar os problemas mencionados, surge uma abordagem alternativa, denominada serviços diferenciados.

Serviços Diferenciados

A arquitetura de Serviços Diferenciados é caracterizada pela definição de classes de serviços (Magalhães e Cardozo, 1999). A arquitetura se utiliza de um campo no cabeçalho do pacote IP

[3] Internet Engineering Task Force.
[4] conexão entre dois pontos da rede.
[5] Resource ReSerVation Protocol.

chamado TOS[6]. Basicamente, o que a arquitetura faz é definir um *layout*[7] para o campo TOS, para especificar um conjunto de procedimentos distintos de envio de pacotes, oferecendo com isso diferentes classes de serviços. O campo TOS passa a ser chamado, na arquitetura de serviços diferenciados, de *DS field*[8].

As classes de serviço são normalmente especificadas através de contratos de serviço, denominados SLA, firmados entre o usuário e o provedor de serviço de *Internet* (ISP[9]). Nesse contrato, o provedor compromete-se a atender o usuário com a QoS (em relação a rede) solicitada e o usuário compromete-se a gerar um fluxo com as características dispostas no contrato, de forma a evitar abusos.

Os contratos SLA podem ser classificados como estáticos ou dinâmicos. Contratos estáticos são negociados de maneira regular (mensalmente ou anualmente, por exemplo). Já contratos dinâmicos possibilitam ao usuário solicitar serviços sob demanda. Este último tipo de contrato se utiliza de algum protocolo de sinalização e controle (como o RSVP) de forma a indicar as necessidades do usuário em determinados momentos (Zhao et al., 2000).

Tal abordagem foi utilizada em Chen e Heidemann (2003), no qual é proposto um algoritmo denominado *Short Flow Differentiating* (SFD) com o intuito de melhorar o desempenho de tráfego *Web* interativo. O algoritmo proposto prioriza tráfegos mais curtos, como os de rajada, visto que, estudos mostraram que tráfegos mais curtos correspondem a mais de 80% das respostas *Web*, além de fluírem mais rapidamente pela rede. Os resultados apresentados, por meio de simulação, mostraram a importância e o impacto na priorização de serviços mais curtos, com uma melhora de mais de 30% no tempo de resposta das transmissões mais curtas.

2.1.2 QoS em Nível de Aplicação

Nas seções anteriores, foram abordadas técnicas de garantia de QoS em nível de rede, onde há esforços para o aprimoramento dessa tecnologia. Embora a QoS tenha aplicação em todos os níveis das arquiteturas convencionais, partes do desenvolvimento nesse campo têm se dado no estudo e elaboração de técnicas aplicáveis às camadas inferiores, especialmente na camada de rede.

Contudo, para que a provisão de QoS seja ampla, é preciso a cooperação entre todas as camadas de rede, de cima a baixo, assim como de todo e qualquer elemento da rede, de fim-a-fim, o que implica na consideração de garantias de QoS em Nível de Aplicação (Casagrande, 2007). Um servidor *Web* não preparado para oferecer QoS poderá anular quaisquer esforços que tenham sido empreendidos pela rede nesse sentido, pois ela trata todas as solicitações que receber por igual, ignorando a priorização relativa das mesmas.

Diversos trabalhos têm abordado QoS em nível de aplicação, tendo como objeto de estudo servidores *Web*. Alguns desses trabalhos merecem destaque, seja pelo embasamento teórico para

[6]Type of Service.
[7]disposição de componentes.
[8]Differentiated Service Field.
[9]Internet Service Provider.

o desenvolvimento de alternativas que venham a complementar as soluções já propostas, como também orientar sobre a necessidade de pesquisas em campos pouco explorados.

Em Teixeira et al. (2005), baseado no modelo *DiffServ* pertinente ao nível de rede, realiza-se uma transposição de seus princípios para a camada de aplicação, concebendo um modelo de Servidor *Web* com Diferenciação de Serviços (SWDS). Nesse modelo, o escalonamento convencional (baseado em FIFO) da fila de requisições pendentes em um servidor *Web* (ou *cluster*[10] de servidores) é substituído por uma política que considera classes com prioridades. Requisições de uma classe mais prioritária têm precedência no atendimento em relação às requisições de classes inferiores.

Outra contribuição desse trabalho é a proposta de um mecanismo de proteção contra negação de serviço, denominado Prioridades Adaptativo (*PRIAdap*), de forma a evitar a monopolização do sistema por partes das classes mais prioritárias. O *PRIAdap* permite regular o nível de priorização do sistema, determinando assim o quão rigoroso será o esquema de prioridades empregado.

Em Estrella et al. (2006) é estendido o trabalho realizado em Teixeira et al. (2005), incluindo um mecanismo de negociação ao módulo de controle de admissão do modelo SWDS, proporcionando com isso melhores médias de tempo de resposta e menores taxas de descarte de requisições, melhorando a QoS oferecida.

Em Traldi et al. (2006) são propostos dois novos algoritmos de diferenciação de serviços com o intuito de prover QoS em servidores *Web*: Reserva Adaptativa de Recursos (*RSVAdap*) e *Weighted Fair Queuing* (WFQ). O algoritmo *RSVAdap* é um algoritmo de particionamento de recursos proposto a partir do trabalho de Teixeira et al. (2005), com a diferença de que realiza a alocação de recursos de forma dinâmica, ou seja, sob demanda, segundo a carga de trabalho vigente. Com isso se obtém uma melhora na utilização do sistema, onde o controle da alocação se baseia no número de requisições de cada classe presente no sistema e no nível de diferenciação pretendido. O algoritmo WFQ é uma adaptação, para o nível de aplicação, do *Weighted Fair Queuing* existente no nível de rede. Esse algoritmo consiste na divisão da capacidade de processamento dos nós entre as classes de requisições, de acordo com os pesos que são atribuídos dinamicamente às classes. Além de oferecer diferenciação entre classes, esse algoritmo de escalonamento tem como característica a ausência da negação de serviço, como pode ocorrer nos mecanismos propostos por Teixeira et al. (2005).

Em Messias (2007) é apresentado um estudo, implementação e avaliação do modelo de servidor *Web* com diferenciação de serviços. Com o intuito de controlar a carga no sistema, algoritmos de reserva de recursos, escalonamento baseado em prioridades e mecanismos de controle de admissão, foram considerados. Embora os algoritmos de reserva de recursos sejam eficientes para provimento de diferenciação entre as classes consideradas, seus desempenhos não foram satisfatórios em algumas situações, devido à arquitetura em que foram implementados e por motivos inerentes ao próprio algoritmo. O algoritmo de escalonamento baseado em prioridades *(PriPro-*

[10] conjunto.

cess), mostrou-se mais eficiente tanto na obtenção de diferenciação de serviço entre as classes, como na obtenção de desempenho. Foi desenvolvido também um mecanismo de controle de admissão com diferenciação de serviços. Os resultados obtidos indicam uma melhora em termos de tempos de respostas e número de requisições completadas para a classe de maior prioridade.

Os trabalho citados têm um tipo de abordagem que pode ser denominada QoS relativa, pois os parâmetros de qualidade oferecidos a uma classe de serviço são formulados com respeito à qualidade oferecida a outra classe. A provisão de QoS pode ser classificada como relativa e absoluta.

Quando a forma de atendimento do sistema é definida com uma diferenciação de prioridade entre as classes, tem-se a QoS relativa. Essa diferenciação se dá de maneira qualitativa, garantindo-se um melhor atendimento a uma determinada classe mais prioritária. No caso específico de tempo de resposta, um contrato de QoS relativa especifica que as requisições provenientes de determinada classe serão atendidas em geral antes daquelas provenientes de uma classe menos prioritária.

Quando a forma de atendimento, porém, é definida com o estabelecimento de métricas e valores de desempenho a serem respeitadas para cada classe individualmente, trabalhando com qualidade em termos quantitativos, tem-se a QoS absoluta. Nessa abordagem, existe a garantia por classe, independente do que é definido para as demais. São estabelecidas taxas mínimas de serviço ou atrasos máximos de atendimento para as requisições. No caso específico de tempo de resposta, um contrato de QoS absoluta especifica limites para o tempo de resposta das requisições de cada classe, independente daquele praticado para as outras.

QoS Absoluta

Mais recentemente têm sido desenvolvidas pesquisas sobre QoS em termos absolutos, com a qual se pode oferecer garantias mais estritas de qualidade.

Em Wei et al. (2005) é proposta uma abordagem de controle *fuzzy*[11] para garantir atrasos absolutos em servidores *Web*. A arquitetura do sistema proposto consiste em um escalonador de conexão, um monitor de sistema e um controlador *fuzzy*.

O escalonador de conexão é responsável pela aceitação de todas as requisições que chegam ao sistema, onde são classificadas de acordo com uma política pré-determinada. O escalonador só aceita a requisição se o número de processos atribuídos a determinada classe de requisições é menor que o contador de processos dessa classe. O monitor é responsável pela medição e informação ao sistema do atraso absoluto de cada classe de requisições. E o controlador *fuzzy* ajusta o contador de cada classe, de forma que o atraso das classes não ultrapasse o limite estipulado.

Em Casagrande (2007) é proposta uma política de escalonamento, chamada EBS, de tempo real não-determinístico (*Soft-RT*) para provisão de garantias de tempo de resposta estocásticas em ambientes interativos, mais especificamente em servidores *Web*.

A EBS utiliza uma estratégia híbrida associada às heurísticas baseadas nos algoritmos clássicos EDF e SJF (ver Seção 2.2.2), levando em consideração, portanto, o tempo de espera em

[11] sistema de auto ajuste.

fila (urgência de *deadline*) e o custo de execução de uma requisição, impondo ao sistema uma menor demanda de recursos e garantindo um compromisso entre desempenho e confiabilidade no atendimento de contratos individuais.

Em Casagrande (2007) foi demonstrado que a política EBS proporciona resultados superiores às heurísticas convencionais. Esses resultados, bem como o funcionamento detalhado dessa política podem ser observados na Seção 2.5.

Devido à crescente utilização e diversificação das aplicações suportadas pela *Internet*, embora seu modelo atual de serviços tenha funcionado e ainda funcione bem para vários tipos de aplicações, seria conveniente a utilização do modelo de Qualidade de Serviço. Como visto (Seção 2.1), as abordagens para provisão de QoS na *Internet* mais utilizadas são as de Serviços Integrados e de Serviços Diferenciados. Além da QoS em nível de rede, foi evidenciada a importância do fornecimento de QoS em nível de aplicação, assunto esse, tema principal do projeto de pesquisa em questão, bem como alguns trabalhos mais relevantes da área estudada, tanto para embasamento teórico para o desenvolvimento de alternativas que venham a complementar as soluções já propostas, como também para orientar sobre a necessidade de pesquisas em campos pouco explorados.

Esforços em outras áreas, como Tempo Real, também têm sido realizados com o intuito de propor novas abordagens para o problema, sendo importante uma breve revisão de seus principais conceitos (Seção 2.2).

2.2 Sistemas de Tempo Real

A garantia de requisitos de QoS absoluta com restrição temporal insere ao problema tratado a necessidade de técnicas associadas a sistemas de tempo real (RT). Com o rápido crescimento da capacidade computacional, os sistemas de tempo real estão sendo empregados em diversas aplicações (automação industrial, centrais nucleares, controle de tráfego aéreo, ferroviário, marítimo e rodoviário, robótica, sistemas de aviação, sistemas de defesa militar, dentre outras) complexas e de grande importância econômica e social.

Combinado com a recente difusão de aplicações interativas como os sistemas de realidade virtual, o leque de aplicabilidade dos sistemas de tempo real está crescendo. Embora essas aplicações em termos de segurança não sejam críticas (não são *Hard-RT*), o cumprimento dos requisitos temporais tem grande importância para o desempenho do sistema. Com isso o interesse no estudo de técnicas para provisão de tempo real está aumentando consideravelmente.

É importante ressaltar que tais sistemas não estão associados com desempenho computacional, mas sim, de assegurar tempos de respostas a eventos externos ao sistema (Nissanke, 1997). Conceitos e técnicas de escalonamento desempenham um papel preponderante no comportamento de sistemas de tempo real (Sha et al., 2004). Dessa forma, tais conceitos e técnicas serão explicitados nas seções subseqüentes 2.2.1 e 2.2.2.

2.2.1 Sistemas de Tempo Real

Uma definição que caracteriza um sistema de tempo real é como aquele em que há requisitos de sincronismo de eventos internos (ao sistema) em relação a eventos externos (do ambiente). Para isto, é necessário garantir que cada requisição seja atendida antes do seu *deadline* (restrição de tempo que corresponde ao tempo máximo que uma requisição deve ser concluída) (Cheng, 2002). Para que um sistema computacional convencional (não de RT) seja considerado correto, eles devem apresentar uma saída adequada a uma entrada. Já para um sistema de tempo real, é necessário além de uma saída adequada, um tempo hábil para essa saída ser fornecida.

Com isto, o requisito-chave na formulação das especificações RT é o limite superior para atrasos no tempo de reação do sistema aos estímulos externos, ou seja, restrições sobre máximos tempos de resposta.

Para a caracterização dos diferentes tipos de sistemas RT, bem como os métodos para escalonamento e gerência de recursos, utilizam-se termos gerais para tratamento da carga de trabalho de sistemas computacionais e de comunicação. Sendo assim este trabalho irá denominar tarefa (ou *task*) como sendo uma função no sistema; já requisição (ou *job*) é uma operação da sequência de operações que compõe uma tarefa. Por exemplo, a tarefa de um servidor *Web* é exibir páginas *Web* para os clientes, e cada pedido de exibição caracteriza uma requisição.

Restrições Temporais

As aplicações RT são caracterizadas por meio de restrições temporais de suas tarefas. Essas restrições impõem o comportamento temporal desejado ou necessário de uma requisição ao sistema. Por exemplo, o limite de tempo de execução de uma tarefa é especificado pelo atributo *deadline* (d). Abaixo, são citadas outras restrições temporais que também são importantes para definir o comportamento temporal:

- **Tempo de chegada** (*arrival time* - at): indica o instante em que o escalonador toma conhecimento de uma nova requisição.
- **Tempo de liberação** (*release time* - rt): indica o instante em que a requisição está disponível para processamento e portanto é incluída na fila de prontas para serem executadas;
- **Tempo de início** (*start time* - st): indica o instante inicial do processamento da requisição;
- **Tempo de execução** (*computation time* - c): indica o tempo gasto para completar a execução de uma determinada requisição. Útil para determinar se uma requisição cumprirá seu *deadline*;
- **Tempo de término** (*completion time* - ct): indica o instante final do processamento da requisição;

- **Tempo de resposta** (*response time* - r): indica o tempo desde a chegada do evento (ativação) até o término da execução da requisição (resposta).
- **Jitter** (J): indica a variação do retardo na entrega de dados, ou seja, a medida de variação do atraso entre as sucessivas requisições.

A Figura 2.1 ilustra uma requisição, indicando o instante das restrições temporais ao longo do tempo.

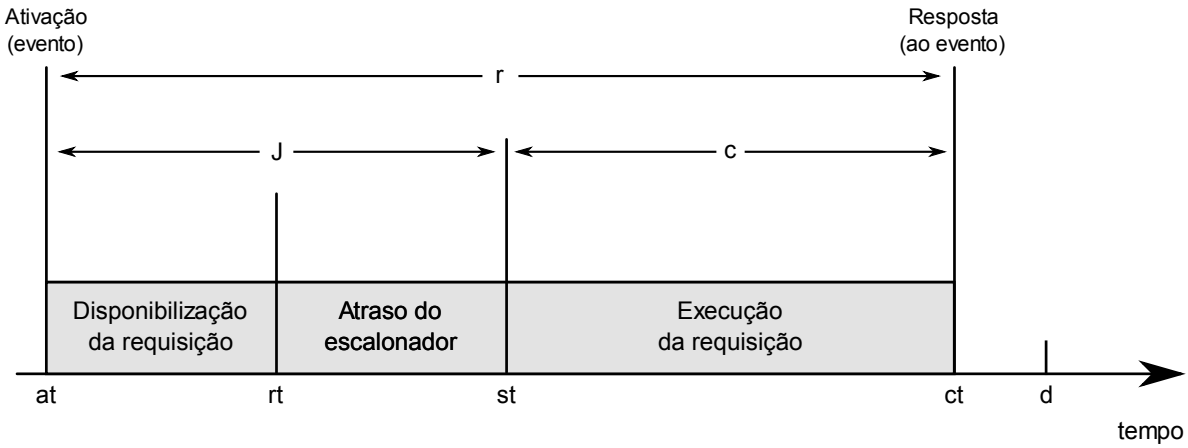


Figura 2.1: Ilustração das Restrições Temporais.

Classificação dos sistemas de tempo real

Os sistemas de tempo real podem ser classificados em três tipos: *Hard-RT*, *Soft-RT* e *Firm-RT*. Eles se diferenciam segundo a forma de restrição de tempo estabelecida pelo sistema a uma determinada tarefa.

Para sistemas *Hard Real-Time*, a precisão de tempo de resposta é criticamente importante e não pode ser depreciada para obter outros ganhos. Nesses sistemas, o não cumprimento de apenas uma restrição temporal (como tempo de liberação e *deadline*) poderá acarretar consequências desastrosas (Zhang et al., 2008), como o comprometimento da confiabilidade do sistema ou até mesmo colocar vida de pessoas em risco (Liu, 2000; Tavares et al., 2008). Por exemplo, uma demora no tempo de resposta de um sistema de manobra de uma aeronave poderá levar a um acidente. Em alguns casos a exatidão do resultado pode até sofrer um relaxamento em favor da garantia de resposta em um tempo hábil.

Para sistemas *Soft Real-Time*, a precisão de tempo é importante, porém não é crítica. Esses sistemas são mais flexíveis, ou seja, o atraso na conclusão de uma tarefa é indesejável, mas é aceitável, pois não trará sérios danos, apenas uma degradação do desempenho. Nesses sistemas, as tarefas são realizadas tão rapidamente quanto possível, podendo haver certo relaxamento na precisão de alguns de seus tempos em situações de sobrecarga do sistema.

Sistemas *Firm Real-Time* são semelhantes aos sistemas *Soft-RT*. A única diferença é em relação ao tratamento das respostas das requisições que estão atrasadas. Nos sistemas *Soft-RT* é possível a escolha da utilização ou não dos dados atrasados. Já nos sistemas *Firm-RT*, por outro lado, não há possibilidade de uso dos dados atrasados. Um exemplo é a comunicação por áudio e vídeo em tempo real, onde não faz sentido executar um trecho de áudio ou vídeo que chegaram atrasados.

Caracterização das Tarefas

Os principais modelos de tarefas tratadas na área de sistemas RT são: periódicas, esporádicas e aperiódicas.

Uma tarefa ($T1$) é periódica (Figura 2.2a) quando os *jobs* chegam ao sistema em um período fixo p (intervalo regular). Por exemplo, uma tarefa que processa sinais de radar a cada 2 segundos.

Uma tarefa (T_2) é esporádica (Figura 2.2b) quando os *jobs* chegam ao sistema em um intervalo mínimo de tempo min. Por exemplo, uma execução de uma manobra de emergência de uma aeronave quando um botão é pressionado, com um tempo mínimo de 20 segundos entre duas requisições de emergência.

Já uma tarefa (T_3) é aperiódica (Figura 2.2c) quando os *jobs* chegam ao sistema com uma frequência indeterminada.

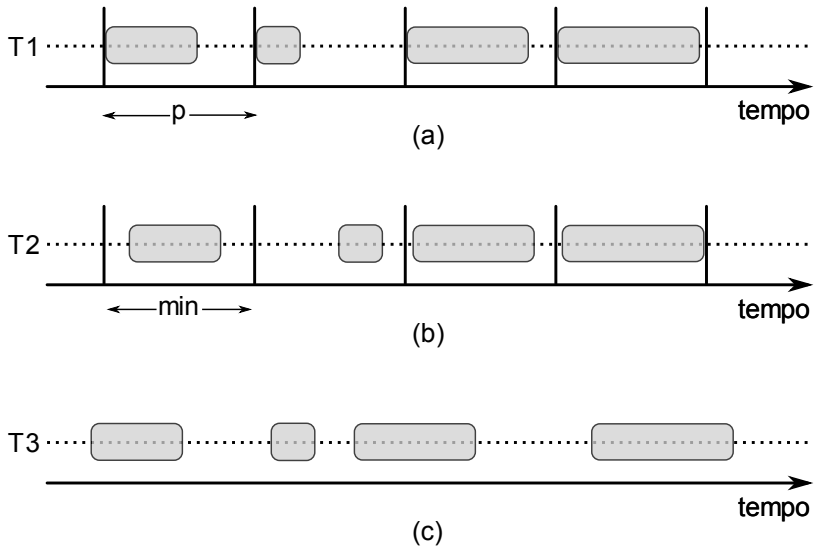


Figura 2.2: Caracterização das tarefas: (a) periódica; (b) esporádica e (c) aperiodica.

2.2.2 Escalonamento em Sistemas RT

O escalonador em um sistema computacional de tempo virtual (Um sistema que não é RT pode ser designado de tempo virtual) é um elemento básico de sistemas multitarefa. Sua meta típica é

decidir qual a ordem de execução das tarefas que estão na fila prontas para serem executadas, de modo a maximizar a média do *throughput* (Número de tarefas atendidas por unidade de tempo) e/ou minimizar a média do *turnaround* (Tempo total entre a submissão de uma tarefa e seu atendimento).

Em sistemas de tempo real, a meta do escalonador corresponde a satisfazer o *deadline* de todas as requisições (Cheng, 2002). Com isso, a relevância do escalonador é ainda mais evidente, visto que está ligada à correção do sistema.

Classificação do Escalonamento

O escalonamento em sistemas RT pode ser classificado de acordo com suas características como natureza do problema, direito de preempção e tipo de abordagem.

Considerando a natureza do problema, o escalonamento pode ser classificado em: estático, dinâmico ou misto. O escalonamento estático é usado em problemas em que se tem conhecimento prévio das características de todas as requisições, onde são usadas para definir a ordem escalonamento. É aplicada em problemas onde a freqüência e o custo das tarefas é bem conhecido. A ordem de escalonamento é definida antes da chegada das requisições, o que leva a vantagens como baixo custo, já que o escalonamento não é feito durante o tempo de execução, além de grande capacidade de predição. O escalonamento dinâmico, por outro lado, é usado em problemas com requisições imprevisíveis. Para cada requisição que chega ao sistema é necessário o reescalonamento da fila. Já o escalonamento misto é a combinação dos dois anteriores, escalonando estaticamente requisições com características conhecidas previamente e ajustando as outras à medida que chegam (Nissanke, 1997).

Considerando o direito de preempção, o escalonamento pode mudar de acordo com a possibilidade ou não de preempção de tarefas quando necessário. Um escalonamento com preempção é aquele que trabalha com a execução de requisições que podem ser interrompidas e continuadas depois sem comprometer sua realização garantindo qualquer restrição de tempo associada. Embora ofereça mais possibilidades de escalonamento, o uso de preempção pode demandar um tempo maior de execução devido à necessidade de troca de contexto. Já um escalonamento sem preempção usufrui de vantagens como maior simplicidade, maior previsibilidade, facilidade de testes e garantia de acesso exclusivo a recursos e dados compartilhados.

Considerando o tipo de abordagem, o escalonamento pode ser classificado em: orientada ao tempo e orientada à prioridade. Na abordagem orientada ao tempo (*Clock-driven* ou *Time-driven*), as decisões sobre qual requisição executar são tomadas em instantes previamente estabelecidos para cada uma antes da execução do sistema. A abordagem orientada à prioridade toma decisões de escalonamento baseadas em eventos (*Event-driven*), como por exemplo, os de liberação e conclusão de requisições. Esse tipo de abordagem só faz sentido em sistemas preemptivos. Uma requisição $T1$ tem maior prioridade sobre outra requisição $T2$ quando a última tem que ser interrompida para permitir a execução de $T1$. Se mais de uma requisição de alta prioridade for requerida

simultaneamente, a que tiver maior prioridade é executada. Em caso de prioridades iguais, uma é escolhida aleatoriamente.

Políticas de Escalonamento de RT

A seguir serão apresentados alguns algoritmos clássicos para sistemas de tempo real com escalonamento orientado a prioridade.

Algoritmo *Rate Monotonic* (RM) Este algoritmo trabalha com requisições periódicas e preemptivas, onde o período da tarefa é igual ao seu *deadline*. O algoritmo atribui prioridades com base em seus períodos: quanto menor o período, maior a prioridade.

A Figura 2.3 ilustra o trecho inicial de um exemplo de comportamento do algoritmo RM, mostrando a ordem em que as requisições seriam processadas. Nesse exemplo, existem duas tarefas periódicas A e B, com tempos de computação (C_i), períodos (P_i) e *deadlines* (D_i) mostrados na Tabela 2.1.

Tabela 2.1: Tabela com exemplo para o algoritmo RM

tarefas periódicas	C_i	P_i	D_i
A	10	20	20
B	25	50	50

A tarefa A sempre terá maior prioridade em relação a B, pois tem período (P_i) menor (20). Essa característica pode ser chamada de prioridade fixa, pois uma tarefa sempre vai ter prioridade em relação à outra. É possível ver também que cada tarefa tem prioridade igual ao seu deadline ($P_i = D_i$). A seguir estão os passos que o algoritmo segue no exemplo:

1. **Instante 0**: Em um sistema hipotético, A e B são executados simultaneamente. A tem maior prioridade, e por isso será executado primeiro.
2. **Instante 10**: A termina seu processamento. B começa a ser executado.
3. **Instante 20**: A suspende B. Faltam 15 unidades de tempo para B terminar sua execução.
4. **Instante 30**: A termina seu processamento. B recomeça a ser executado.
5. **Instante 40**: A suspende novamente B. Faltam 5 unidades de tempo para B terminar sua execução.
6. **Instante 50**: A termina seu processamento. *Deadline* de B é alcançado sem o término da execução. Houve uma falha no sistema.
7. **Instante 55**: B terminará sua execução com atraso de 5 unidades de tempo.

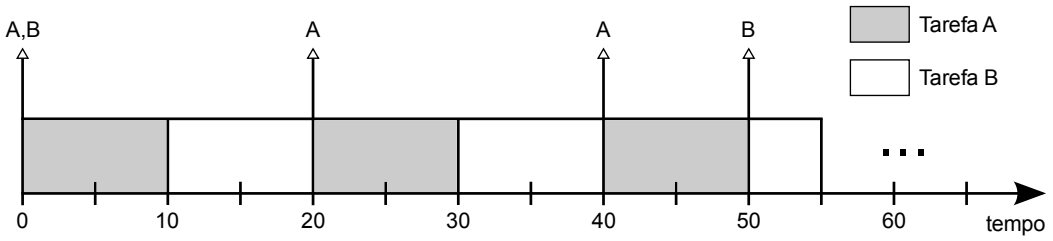


Figura 2.3: Exemplo da Utilização do Algoritmo de Escalonamento RM

Esse algoritmo pode ser considerado um algoritmo guloso, pois sempre escolhe a requisição com menor período (e consequentemente com o menor *deadline*), levando a uma solução ótima local, mas não global. O final do exemplo mostra os passos seguidos em um sistema *Soft-RT*. Em um sistema *Firm-RT* ou *Hard-RT*, B não seria mais executado, pois o seu *deadline* já ocorreu.

Algoritmo *Deadline Monotonic* (DM) Outro algoritmo de prioridade fixa. Este algoritmo trabalha com requisições periódicas e preemptivas. Ele estende o algoritmo RM, sendo mais flexível assumindo *deadlines* menores ou iguais aos períodos das requisições ($D_i \leq P_i$). O algoritmo atribui prioridades baseados em seus *deadlines*: quanto menor o *deadline*, maior a prioridade. Quando $D_i = P_i$, os algoritmos RM e DM serão idênticos.

A Figura 2.4 ilustra o trecho inicial de um exemplo de comportamento do algoritmo DM, mostrando a ordem em que as requisições seriam processadas. Nesse exemplo, existem três tarefas periódicas A, B e C com tempos de computação (C_i), períodos (P_i) e *deadlines* (D_i) mostrados na Tabela 2.2.

Tabela 2.2: Tabela com exemplo para o algoritmo DM

tarefas periódicas	C_i	P_i	D_i
A	2	10	6
B	2	10	8
C	8	20	16

Observando os *deadlines* da Tabela 2.1, a ordem de prioridade, da maior para menor, é A (6), B (8) e C (16). A seguir estão os passos que o algoritmo segue no exemplo:

1. **Instante 0**: Em um sistema hipostético, A, B e C são executados simultaneamente. A tem maior prioridade, e por isso será executado primeiro.

2. **Instante 2**: A termina seu processamento. B começa a ser executado.

3. **Instante 4**: B termina seu processamento. C começa a ser executado.

4. **Instante 10**: A e B são executados. A suspende C. Faltam 2 unidades de tempo para C terminar sua execução.

5. **Instante 12**: A termina seu processamento. B começa a ser executado.

6. **Instante 14**: B termina seu processamento. C recomeça a ser executado.

7. **Instante 16**: C termina seu processamento.

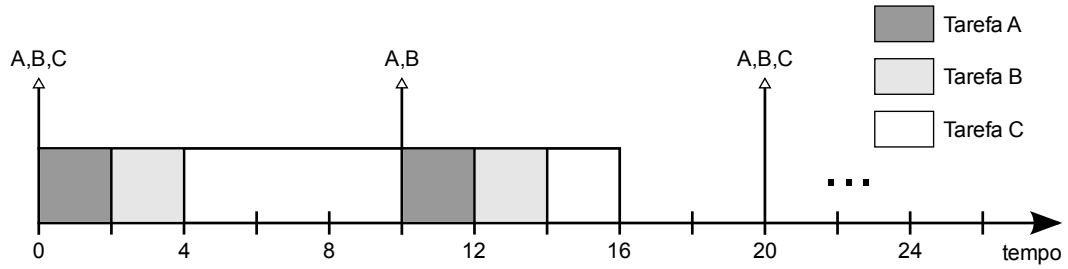


Figura 2.4: Exemplo da Utilização do Algoritmo de Escalonamento DM

Quando o *deadline* de todas as tarefas for proporcional ao seu período, os algoritmos RM e DM serão idênticos. Porém, quando os *deadlines* forem arbitrários, o algoritmo DM apresentará melhor desempenho, pois ele pode apresentar escalonamento factível quando RM falha, enquanto o algoritmo RM sempre falha quando o DM falha (Nissanke, 1997).

Algoritmo *Earliest Deadline First* (EDF) Este algoritmo trabalha com tarefas preemptivas que podem ser periódicas ou não. O escalonamento das requisições é realizado de forma dinâmica (em tempo de execução), atribuindo prioridades baseados em seus *deadlines* absolutos (d_i). d_i é calculado somando-se o instante no tempo de chegada com o seu *deadline* relativo. A requisição mais prioritária é aquela que tem o *deadline* absoluto mais próximo do tempo atual, ou seja, a priorização se dá baseado na urgência das requisições. A cada nova requisição que chega ao sistema, a fila de requisições prontas é reordenada, atualizando a nova distribuição de prioridades (Farines et al., 2000; Liu e Layland, 2002).

A Figura 2.5 ilustra o trecho inicial de um exemplo de comportamento do algoritmo EDF, mostrando a ordem em que as requisições seriam processadas. O exemplo é o mesmo utilizado no algoritmo RM (Tabela 2.1). A seguir estão os passos que o algoritmo segue no exemplo:

1. **Instante 0**: A e B são executados simultaneamente. A tem maior prioridade, pois $d_A = 20$ e $d_B = 50$.

2. **Instante 10**: A termina seu processamento. B começa a ser executado.

3. **Instante 20**: A suspende B, pois $d_A = 40$ e $d_B = 50$. Faltam 15 unidades de tempo para B terminar sua execução.

4. **Instante 30**: A termina seu processamento. B recomeça a ser executado.

5. **Instante 40**: A não suspende B, pois $d_A = 60$ e $d_B = 50$.

6. **Instante 45**: B termina seu processamento. A começa a ser executado.

7. **Instante 50**: B não suspende A, pois $d_A = 60$ e $d_B = 100$.

8. **Instante 55**: A termina sua execução.

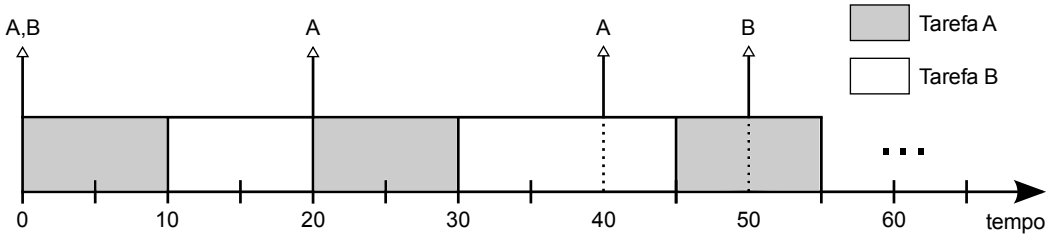


Figura 2.5: Exemplo da Utilização do Algoritmo de Escalonamento EDF

É possível observar que, ao contrário do algoritmo RM, a EDF não causou uma falha no sistema no final do exemplo.

Essa seção introduziu os conceitos principais da área de sistemas de tempo real, os quais são responsáveis por estabelecer parâmetros de serviço em termos temporais. Foram abordadas as principais restrições temporais utilizadas na literatura. Em seguida foi discutida a classificação em três vertentes: *Hard-RT*, *Soft-RT* e *Firm-RT* (Seção 2.2.1).

Além dos conceitos primordiais ao entendimento da área, a seção apresenta o escalonamento em sistemas RT, incluindo algumas formas de classificação para o mesmo, bem como alguns algoritmos clássicos utilizados em sistemas de tempo real: RM, DM e EDF.

Uma das formas para realizar a validação de algoritmos de escalonamento é a utilização de modelagem baseada em simulação de sistemas, que oferece facilidades para a representação do comportamento de um sistema do mundo real sobre o tempo e o qual será abordado na Seção 2.3.

2.3 Simulação de Sistemas

A simulação é uma ferramenta para estudar e analisar o comportamento e as reações de um determinado sistema, de forma a produzir suas propriedades e características, possibilitando a sua

manipulação e o seu estudo detalhado (MacDougall, 1989; Shannon, 1998). Isso é útil quando a complexidade de um sistema real inviabiliza seu estudo por experimentação; quando se pretende analisar em isolado a influência de mudanças organizacionais ou ambientais no comportamento de um sistema; ou ainda para testar novos projetos ou políticas antes de serem implementados (Banks et al., 2000).

A simulação é representada através de modelos, construídos a partir de um conjunto de suposições operacionais de um sistema real. Uma vez construída e validada, a simulação pode ser utilizada para investigar uma ampla variedade de questões do tipo "o que aconteceria se" acerca de um sistema real, podendo, dessa forma, ser considerado como uma descrição do sistema real. A validação dos modelos pode ser realizada, por meio analítico utilizando a lógica dedutiva da matemática, ou por meio de simulação empregando métodos numéricos na análise (Law e Kelton, 2006).

A execução de modelos de simulação possibilita o fornecimento de resultados satisfatórios sem a necessidade de se interferir no sistema real. Também possibilita um controle melhor do ambiente, descartando fatores externos que não interessam para os propósitos do experimento. Tais resultados, quando analisados estatisticamente produzem informações que podem contribuir na tomada de decisões que visam à solução de problemas.

2.3.1 Modelos de simulação

A definição de alguns termos e conceitos é necessária para a representação de um modelo de simulação. Dessa forma, define-se estado de um sistema como um conjunto de variáveis necessárias para descrever um sistema em um dado período de tempo. O estado pode ser classificado como discreto ou contínuo, dependendo do tipo de estados que predominam no sistema ou no foco de análise. Em sistemas discretos, o estado sofre mudanças instantâneas em pontos distintos de tempo, como por exemplo, o número de clientes em uma fila. Em sistemas contínuos, o estado sofre mudanças contínuas ao longo do tempo, como o volume de água em uma hidrelétrica.

Os modelos de simulação ainda podem ser classificados em:

- **estáticos** ou **dinâmicos**: modelos estáticos representam o estado de um sistema em um determinado instante, sendo que, em suas formulações a variável de tempo não é considerada. Já os modelos dinâmicos representam as alterações de estado do sistema ao longo do tempo de simulação.

- **determinísticos** ou **estocásticos**: modelos determinísticos não usam variáveis aleatórias em suas formulações, sendo que dados uma entrada ou um conjunto de entradas, a saída sempre será a mesma. Já os modelos estocásticos utilizam essas variáveis, de forma que, a saída dependerá da variação dos valores estocásticos.

- **tempo real** ou **tempo simulado**: modelos de tempo real têm escala de tempo real, ou seja, os eventos ocorrem e são tratados na mesma escala de tempo de um sistema real. Por outro

lado, modelos de tempo simulado não acompanham a escala de tempo de um sistema real. Isto quer dizer que muitos anos de tempo de simulação podem ocorrer em poucos segundos de processamento.

Além da descrição da estrutura estática, a modelagem de um sistema abrange também a representação de sua composição dinâmica, ou seja, o modo que o sistema realiza trabalho (MacDougall, 1989). A composição dinâmica pode ser descrita em termos de:

- **processos**: conjunto de atividades relacionadas logicamente, com tempo de execução formado pela soma dos tempos de execução e atraso de cada uma dessas atividades.
- **eventos**: representa uma mudança no estado do sistema, como por exemplo, uma CPU que passa de um estado ocioso para ocupado. Essa mudança de estado resulta da ação de uma atividade.

2.3.2 Simulação de Filas

Filas estão presentes em diversas situações no mundo real, como em bancos, supermercados, entre outros. A formação de filas ocorre porque a procura por um serviço é maior que a capacidade de atender essa procura.

Dentre as técnicas utilizadas para a modelagem de sistemas com ocorrência de filas, tais como servidores *Web*, as Redes de Fila são as mais utilizadas.

A Figura 2.6 ilustra os elementos que compõem uma fila. Podemos observar que a partir de uma população surgem usuários que formam uma fila, aguardando um determinado tipo de serviço oferecido pelos servidores. Quando um servidor está disponível, um usuário é atendido. Quando o atendimento é finalizado, o usuário sai do sistema e o servidor pode atender outro usuário. O termo usuário pode representar tanto uma pessoa, quanto uma requisição.

O conhecimento das características básicas de sistemas de filas é essencial para a descrição adequada do sistema a ser modelado. Sendo assim, tais características são detalhadas a seguir:

- **Tamanho da população**: número potencial (finito ou infinito) de usuários que podem chegar ao sistema;
- **Chegada dos usuários**: o processo de chegada dos usuários apresenta um comportamento estocástico, ou seja, é possível sua previsão em termos de probabilidade;
- **Capacidade do sistema**: corresponde ao número máximo de usuários que o sistema suporta. Em casos de capacidade muito grande, considera-se que a fila seja infinita. Se as filas alcançarem um determinado comprimento, de forma que nenhum usuário possa entrar no sistema até que haja um espaço disponível, considera-se, um sistema de fila finita;

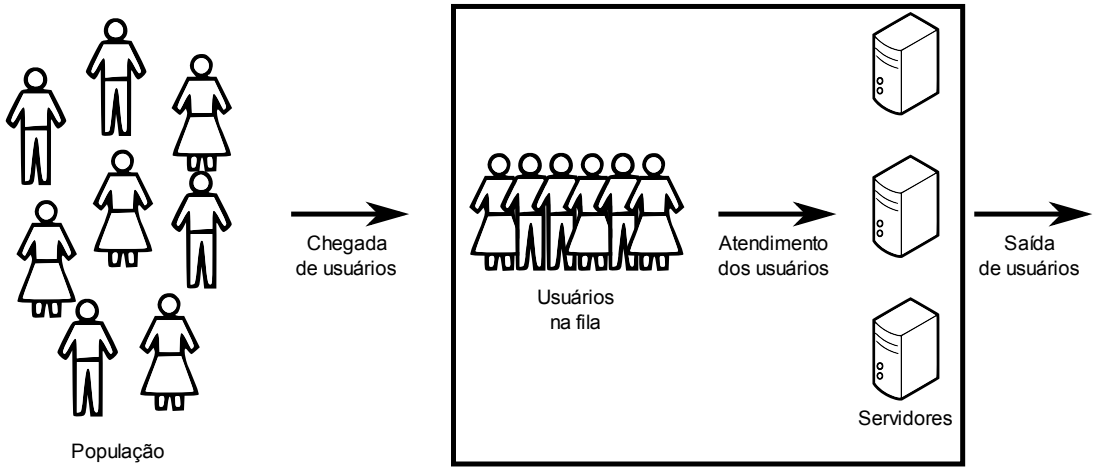


Figura 2.6: Elementos de uma fila

- **Disciplina de atendimento de filas**: determina a ordem de atendimento dos usuários que estão na fila. A disciplina mais usada é a FCFS[12], onde o primeiro a chegar é o primeiro a ser atendido. Há também a LCFS[13], onde o último a chegar é o primeiro a ser atendido. Existem também disciplinas baseadas em esquemas de prioridade. Elas podem ser de dois tipos: preemptivos onde um usuário de maior prioridade é atendido imediatamente, mesmo que um usuário de menor prioridade esteja sendo atendido no momento. Neste caso o serviço do usuário menos prioritário é suspenso para ser reiniciado depois dos atendimentos dos serviços dos usuários mais prioritários; e não-preemptivos onde não ocorre a suspensão de serviço;

- **Número de servidores**: determinam o número de estações de serviços paralelos que podem atender os usuários simultaneamente. A Figura 2.7(a) representa um centro de serviço com um único servidor. Quando um sistema possui mais de um servidor, ela pode apresentar duas variações: uma fila para todos os servidores, como nos caixas de um banco (Figura 2.7(b)); e uma fila para cada servidor, como nos caixas de um supermercado (Figura 2.7(c));

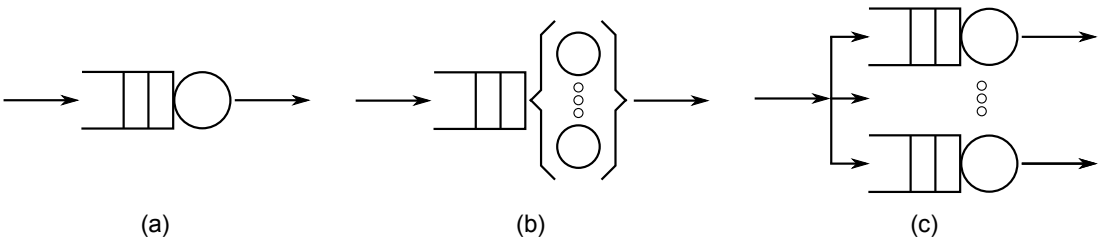


Figura 2.7: Representação de centros de serviços

[12]First Come First Server.
[13]Last Come First Server.

- **Tempo de serviço**: corresponde ao tempo de utilização dos serviços pelos usuários. Da mesma maneira que o processo de chegada dos usuários no sistema, esta apresenta um comportamento estocástico, sendo válidas as mesmas distribuições estatísticas apresentadas.

Baseado no comportamento de entrada e saída dos usuários, os modelos de Redes de Filas podem ser classificados em:

- **modelos abertos**: onde os usuários usam o sistema uma só vez, e obrigatoriamente, saem do mesmo;
- **modelos fechados**: onde o número de usuários que circulam pelo sistema é fixo, de forma que não entram nem saem do sistema;
- **modelos mistos**: corresponde a uma mistura dos dois anteriores, onde temos classes de usuários com modelo aberto e outras classes com modelo fechado.

A Figura 2.8 ilustra esses três modelos de Redes de Filas (Lazowska et al., 1984).

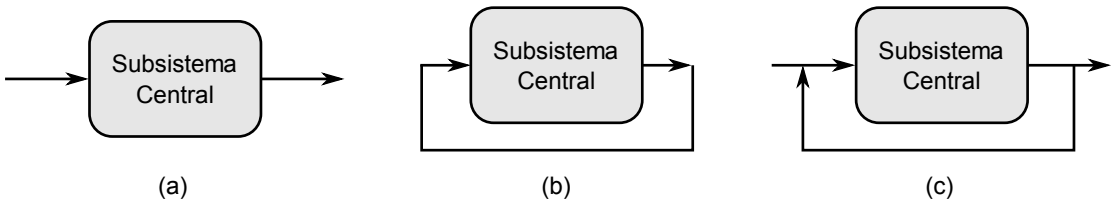


Figura 2.8: Modelos de Redes de Filas: (a) Aberto, (b) Fechado e (c) Misto.

Para facilitar a descrição e classificação desses modelos, em 1953 foi proposta por David G. Kendell, um padrão de representação: a notação de *Kendell*. Esse é o padrão mais utilizado nos trabalhos sobre teoria de filas (Tijms, 2003). Nessa notação, uma fila é representada pelo conjunto de símbolos $A/B/C/K/N/D$. Cada um desses símbolos é descrito a seguir:

- A: Distribuição que descreve o intervalo de chegada dos usuários;
- B: Distribuição que descreve o tempo de serviço dos servidores;
- C: Número de servidores;
- K: Capacidade do sistema;
- N: Tamanho da população;
- D: Disciplina de atendimento da fila.

Quando os parâmetros K e N assumem valores infinitos, podem ser omitidos. O parâmetro D também pode ser omitido quando representa a disciplina de fila FIFO. As distribuições de probabilidade mais comuns para A e B, são representadas pelos símbolos a seguir:

- D: Distribuição determinística (valores constantes)
- M: Distribuição exponencial;
- E_k: Distribuição Erlang de estágio k;
- H_k: Distribuição hiper-exponencial de estágio k;
- G: Distribuição arbitrária.

A distribuição exponencial negativa (ou simplesmente exponencial) é a mais utilizada em modelagem de filas (Jain, 1991). Uma discussão detalhada das várias distribuições de probabilidade, seus parâmetros, e estimativa é feita em Law e Kelton (2006).

2.3.3 Linguagens de Simulação

As linguagens de simulação são utilizadas para o desenvolvimento e a solução de modelos de simulação. Elas podem ser classificadas em orientados a processos ou a eventos.

Linguagens orientadas a processo são mais recomendadas para implementar modelos de simulação de grande escala. Elas trabalham em um nível alto de caracterização do sistema, permitindo uma descrição direta do sistema a ser simulado. Isso permite uma implementação ágil e uma grande similaridade entre o modelo e o sistema. É muito importante, principalmente em um ambiente de desenvolvimento no qual o projeto do sistema sofre constantes mudanças. Exemplos desse tipo de linguagem são o ASPOL (MacDougall e McAlpine, 1973), SIMULA (Birtwhistle et al., 1979) e CSIM (Schwetman, 1986; Hlavicka e Racek, 2002).

Já nas linguagens orientadas a eventos o programa de simulação é organizado como um conjunto de rotinas ou seções de acontecimentos (MacDougall, 1989). Essas linguagens dão ao modelador uma visão global e de alto nível do sistema, agrupando ações de atividades logicamente não relacionadas em uma única rotina de evento, o que causa a perda da "identidade" com a estrutura do sistema e dificulta a modificação do mesmo. Além disso, essa abordagem permite que se conheça o estado de qualquer entidade do sistema em qualquer instante de tempo. Sendo assim, tais linguagens são mais adequadas a modelos de pequena e média escala (Cabral e Souto, 2004; MacDougall, 1989; Lazowska et al., 1984; Jain, 1991). Exemplos desse tipo de linguagem são o SMPL (MacDougall, 1989), SIMPACK (Fishwick, 1992) e NS_2 (Issariyakul e Hossain, 2008).

2.3.4 Probabilidade e Estatística para Simulação

A partir dos resultados obtidos dos experimentos através da simulação, estudos referentes ao comportamento do sistema devem ser realizados. Pelo fato da simulação possuir elementos aleatórios, a variabilidade das saídas deve ser analisada estatisticamente sobre a precisão e sensibilidade do modelo. Basicamente, deve-se estudar qual saída seria obtida caso a simulação fosse realizada

novamente ou tivesse um tempo maior de execução. Quando isso é feito, assume-se que o modelo de simulação é estocástico e que os elementos aleatórios do modelo vão produzir saídas que são probabilísticas. A análise adequada de um modelo requer a utilização de uma série de conceitos e ferramentas pertencentes à probabilidade e estatística, que são apresentados de forma sucinta a seguir.

Variância e Desvio Padrão

A descrição do comportamento do sistema utilizando apenas a média, em determinadas situações, não é suficiente. A média $\bar{x}$ não apresenta nenhuma informação sobre o espalhamento dos dados, ou seja, o quão distante as amostras estão do valor médio. A média de dois conjuntos $A = \{5, 10, 15\}$ e $B = \{0, 10, 20\}$, é a mesma (10), no entanto, o espalhamento é diferente.

A avaliação da qualidade do serviço de aplicações multimídia na *Internet*, por exemplo, não depende somente do valor médio de atraso, mas também da variação (ou dispersão) destes, visto que um valor médio razoável pode ser resultado da combinação de atrasos elevados e atrasos pequenos. Uma grande variação do atraso pode ser mais prejudicial para aplicação do que um atraso médio elevado. Sendo assim, um parâmetro para medir a dispersão dos dados em relação ao valor médio é importante (Kamienski et al., 2002).

A variância e o desvio padrão são parâmetros que medem a dispersão dos dados em relação à média. A variância, tradicionalmente representada por σ^2, é definida como o desvio quadrático médio da média e é calculada de uma amostra de dados a partir da Equação 2.1, onde *n* corresponde ao número de elementos da amostra coletada e $(x_i - \bar{x})^2$ corresponde ao quadrado da distância entre uma amostra x_i e a média da amostra $\bar{x}$. O quadrado da diferença é utilizado para computar o valor absoluto da distância.

$$\sigma^2 = \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n-1} \tag{2.1}$$

O desvio padrão, representado por σ, corresponde à raiz quadrada da variância (Equação 2.2). Tem a mesma função da variância, porém apresenta a vantagem de permitir uma interpretação direta da variação da amostra de dados, pois o desvio padrão é expresso na mesma unidade dos dados.

$$\sigma = \sqrt{variancia} = \sqrt{\sigma^2} \tag{2.2}$$

Intervalo de Confiança

Para encontrar uma estimativa perfeita para a média, é necessário um número infinito de amostras. Como isso não é possível, podemos obter limites probabilísticos (intervalo de confiança), c_1 e c_2, de modo que a média exata $\bar{x}$ pertença ao intervalo $[c_1, c_2]$, com uma certa probabilidade $(1 - \alpha)$ de acerto.

A Equação 2.3 representa a probabilidade de um intervalo de confiança estar correto, onde α corresponde ao nível de significância e $1 - \alpha$ corresponde ao coeficiente de confiança.

$$P(c_1 \leq \bar{x} \leq c_2) = 1 - \alpha \tag{2.3}$$

De forma geral, o intervalo de confiança é explicitado em termo de um percentual próximo de 100%, por exemplo, 90% ou 95%. Já o nível de significância α é explicitado como fração e é usualmente próximo de zero, por exemplo, 0,05 ou 0,1 (Kamienski et al., 2002).

Essa seção introduziu os principais conceitos de simulação de sistemas, que tem como características a elaboração de um modelo de um sistema real (ou hipotético) e a condução de experimentos com o objetivo de entender ou avaliar o comportamento de um sistema. Foi apresentada uma técnica para simulação de filas e a teoria a partir da qual ela é fundamentada.

Na Seção 2.4 será descrito suscintamente como é o processo de cálculo da influência dos fatores de entrada na variável de saída analisada. É utilizado o método fatorial completo descrito por (Jain, 1991), onde é preciso 2 níveis para cada um dos K fatores.

2.4 2^K Fatorial Completo

O fatorial completo (*fatorial design*, em inglês) descreve qual a influência de cada um dos fatores com relação a um parâmetro de resposta, de acordo com os níveis escolhidos. Ela é feita através de um modelo de regressão linear da forma como mostra a Equação 2.4 (exemplo com 2 fatores, ou $K = 2$).

$$y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B \tag{2.4}$$

Onde y é a variável de resposta e x_w (w é o fator, nesse caso A e B) pode assumir os valores **-1** e **1** que representa cada um dos 2 níveis escolhidos. Combinando os níveis de cada fator, teremos $2^{k=2}$ cenários:

$$\begin{aligned} y_1 &= q_0 - q_A - q_B + q_{AB} \\ y_2 &= q_0 + q_A - q_B - q_{AB} \\ y_3 &= q_0 - q_A + q_B - q_{AB} \\ y_4 &= q_0 + q_A + q_B + q_{AB} \end{aligned} \tag{2.5}$$

Assim, podemos obter os valores de q_w em relação à variável de resposta de cada um dos cenários:

$$\begin{aligned} q_0 &= \tfrac{1}{4}(y_1 + y_2 + y_3 + y_4) \\ q_A &= \tfrac{1}{4}(-y_1 + y_2 - y_3 + y_4) \\ q_B &= \tfrac{1}{4}(-y_1 - y_2 + y_3 + y_4) \\ q_{AB} &= \tfrac{1}{4}(y_1 - y_2 - y_3 + y_4) \end{aligned} \tag{2.6}$$

A importância de um fator é medida pela proporção na variação total da variável de resposta que é explicada por esse fator. Se um fator explica por 95% e outro fator explica por 5% da variação total, então o segundo fator pode ser considerado não importante em muitas situações práticas. O dividendo da Equação 2.1 é chamado de variação total de y ou soma dos quadrados total (SST [14]), como mostra a Equação 2.7, onde K é a quantidade de fatores, y_i é a variável de resposta para o experimento i e $\bar{y}$ é a média da variável de resposta de todos os experimentos.

$$\text{Variação total de } y = \text{SST} = \sum_{i=1}^{2^K} (y_i - \bar{y})^2 \tag{2.7}$$

Para $k = 2$, a variação pode ser dividida em três partes, segundo a Equação 2.8.

$$SST = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2 \tag{2.8}$$

As três partes do lado direito da equação representam a porção da variação total explicada pelo fator A, B e pela interação AB, respectivamente. Então, $2^2 q_A^2$ é a porção de SST explicado pelo fator A e é denotado por SSA. Similarmente, SSB é $2^2 q_B^2$ e SSAB é $2^2 q_{AB}^2$. Assim temos a Equação 2.9.

$$SST = SSA + SSB + SSAB \tag{2.9}$$

Então a fração explicada pelo fator A pode ser representada pela Equação 2.10. O mesmo se aplica ao fator B e a interação AB.

$$\text{Fração da variação explicada por } A = \frac{SSA}{SST} \tag{2.10}$$

Quando expressada por porcentagem, essa fração mede a importância do fator A. Os fatores com alta porcentagem de variação são considerados importantes.

Essa seção explica como são calculadas as influências de um fator em um dado experimento e quando elas podem ser consideradas importantes. O cálculo das influências é utilizado em toda a análise desse trabalho de mestrado.

Na Seção 2.5 será apresentada a política de escalonamento para sistemas de tempo real EBS (*Exigency Based Scheduling*), visto que essa abordagem é a base principal para a realização deste trabalho.

[14] Sum of Squares Total

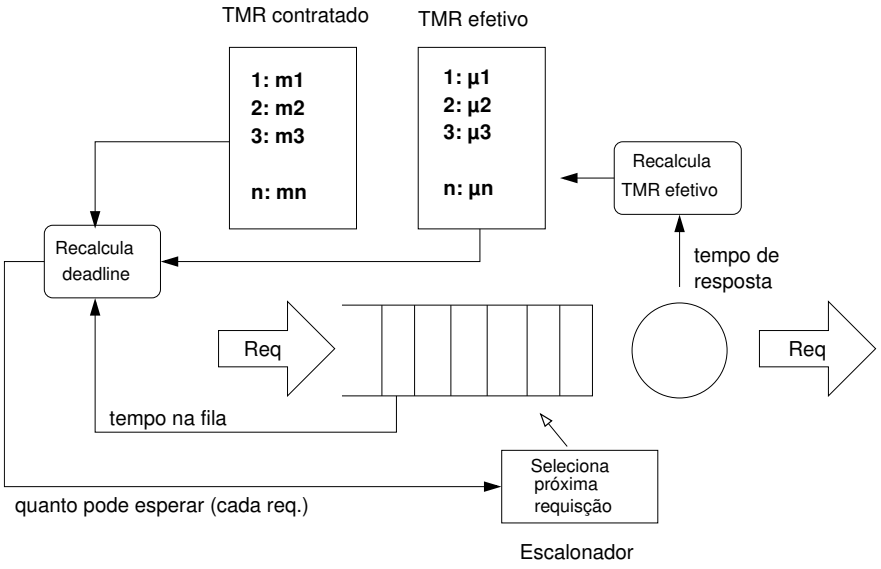
2.5 EBS: *Exigency Based Scheduling*

A política EBS (Casagrande et al., 2007a), a qual realiza o escalonamento de requisições *Web* em sistemas *Soft-RT* não-determinísticos, conforme apresentado na Seção 2.1.2, provê garantias de QoS absoluta em nível de aplicação.

A seguir é apresentado o funcionamento da política EBS, visto que essa abordagem é a base principal para a realização deste trabalho.

2.5.1 Política de escalonamento EBS

Arquitetura EBS (Casagrande et al., 2007b) utiliza estratétiga escalonamento realimentado (feedback):



A cada ciclo de escalonamento, o algoritmo calcula quanto tempo cada requisição pode aguardar para ser executada (deadline), de modo que essa espera não cause um aumento excessivo no TMR do respectivo usuário. O deadline é utilizado pelo escalonador para tomar a decisão sobre a próxima requisição a ser atendida.

A EBS é um algoritmo de escalonamento híbrido, resultado da combinação entre os algoritmos EDF (explicado na sessão 2.2.2) e SJF. No algoritmo SJF, as requisições que possuírem menor tempo de processamento estimado, têm prioridade sobre as demais, o que proporciona baixa latência no sistema. Essa combinação garante um compromisso entre o desempenho e a confiabilidade no atendimento de contratos dos clientes, permitindo um ajuste ponderado de comportamento entre o desempenho proporcionado pela SJF e o maior grau de diferenciação de serviços oferecido pela EDF, ao tentar escalonar as requisições mais urgentes primeiro.

A EBS é usada em uma arquitetura de escalonamento realimentada (*feedback scheduling*) que implementa um mecanismo adaptativo de alocação de recursos. Após o término de uma requisição j de um usuário u, o tempo de resposta médio de u (Ω_u) é recalculado, conforme a Equação 2.11.

$$\Omega_u = \frac{(\Omega'_u \cdot R_u) + (time() - timeStamp_j)}{R_u + 1} \tag{2.11}$$

Ω'_u representa o antigo tempo médio de resposta de u; R_u o número de requisições anteriormente submetidas por u; $time()$ o tempo atual do sistema; e $timeStamp_j$ o tempo que a requisição j pertencente a u chegou. Assim a equação atualiza o tempo médio de resposta do usuário u depois da última requisição $R_u + 1$ ser atendida.

Essa política não é preemptiva, portanto à medida que as requisições chegam ao sistema e não encontram servidores disponíveis, aguardam em um fila de espera, mesmo se possuírem maior prioridade se comparados àqueles que estão sendo executados no momento. Ao término da execução de uma requisição, o escalonador recalcula o tempo médio de resposta atual daquele usuário através da Equação 2.11, e em seguida busca na fila a requisição mais urgente. A prioridade P_j de uma requisição j na fila é calculada através da multiplicação do *deadline* instantâneo D_j com o tempo esperado de processamento T_{p_j}. Essa relação é mostrada na Equação 2.12.

$$P_j = D_j \cdot T_{p_j} \tag{2.12}$$

Quanto menor o valor de P_j, mais prioritário vai ser sua colocação na fila. Assim sendo, requisições com menor *deadline* instantâneo D_j (ou seja, as requisições que estão mais próximas para expirar) e menor tempo de processamento T_{p_j} serão atendidos antes, e vice-versa.

O *deadline* instantâneo (D_j) é o tempo máximo que uma requisição de um usuário pode esperar sem que seu tempo médio de resposta efetivo (Ω_u) ultrapasse o seu contrato. Ela varia junto com o tempo, tornando-se menor à medida que o tempo passa. Ela é estabelecida através da Equação 2.13.

$$\frac{(\Omega_u \cdot R_u) + T_{w_j} + D_j}{R_u + 1} = \Omega_{c_u} \tag{2.13}$$

T_{w_j} representa o tempo de espera atual da requisição j; e Ω_{c_u} o contrato estabelecido para o cliente u. Sendo assim, à medida que a requisição j espera na fila, T_{w_j} aumenta, fazendo com que D_j tenha que diminuir para que a Equação 2.13 continue válida, pois todos os outros valores são fixos (até que a requisição seja atendida).

A equação completa do cálculo da prioridade de uma requisição j pode ser obtida isolando D_j da Equação 2.13 e substituindo em 2.12 resultando na Equação 2.14.

$$P_j = ((\Omega_{c_u} \cdot (R_u + 1)) - (\Omega_u \cdot R_u) - T_{w_j}) \cdot T_{p_j} \tag{2.14}$$

Na EBS não existe a possibilidade de *starvation*[15], como acontece na política SJF, pois utiliza também o *deadline* como critério de priorização. Dessa forma, mesmo que na fila exista uma

[15]quando uma requisição nunca é atendida, pois ao longo do tempo sempre aparecem requisições de prioridade maior.

requisição que possua um tempo estimado de processamento relativamente grande, elas não ficarão indefinidamente no sistema com a chegada de outras requisições mais curtas, pois quanto maior o tempo de espera, menores serão seus *deadlines* instantâneos, o que eventualmente permitirá a elas serem atendidas.

Nos casos onde o contrato for violado, o *deadline* instantâneo assume valores negativos, representando que o tempo que a requisição pode aguardar na fila é menor que zero. A existência de *deadlines* com valores negativos na Equação 2.12 não satisfaz os objetivos da EBS, invertendo a prioridade das requisições. Por exemplo, a prioridade de duas requisições 1 e 2, ambos com *deadline* instantâneo negativo $D_i = -1$ e tempos esperados de processamento $T_1 = 1$ e $T_2 = 2$, serão $P_1 = -1$ e $P_2 = -2$, respectivamente. Assim sendo, 2 terá prioridade em relação à 1, pois $P_2 < P_1$, mesmo com tempo esperado de processamento maior. Nesses casos, é necessária uma extensão da Equação 2.12, para tratar os *deadlines* "negativos". Assim a Equação 2.15 foi proposta em Casagrande (2007) para lidar com *deadlines* "negativos".

$$P_j = \begin{cases} D_j \cdot T_{p_j} & \text{se } D_j \geq 0 \\ D_j \cdot \frac{1}{T_{p_j}} & \text{se } D_j < 0 \end{cases} \tag{2.15}$$

Dessa forma garante-se que as requisições mais urgentes, independentemente de terem descumprido ou não seus *deadlines*, e com menores custos esperados de processamento sejam escalonados primeiro. Note que essa extensão só é necessária para sistemas *Soft-RT*, já que em sistemas *Firm-RT* as requisições atrasadas não são mais necessárias.

2.5.2 Ambiente de Simulação

Em Casagrande (2007) foi utilizado um modelo de simulação orientado a eventos, discreto, dinâmico, estocástico e de tempo simulado (Seção 2.3.1) para avaliar a eficiência e desempenho da política EBS. Utilizando os conceitos de filas (Seção 2.3.2), foi construído um modelo que representa um servidor *Web* com suporte a QoS, o qual descreve, portanto, os principais eventos que ocorrem em um sistema real do gênero, tais como: eventos para tratar chegada de requisições, solicitação de serviço e liberação de recurso. O modelo do servidor é do tipo monoprocessado com uma fila única de espera.

Para avaliação da QoS oferecida, foram utilizadas três variáveis de resposta: a média do tempo de resposta do sistema, a satisfação dos usuários e a variação da satisfação. A primeira representa o tempo médio de residência das requisições de um usuário no sistema, ou seja, o intervalo entre a submissão e o completo recebimento do resultado da requisição (Jain, 1991). A segunda representa a porcentagem de requisições de cada usuário em que o tempo médio de atendimento ficou abaixo do *deadline*, ou seja, foram cumpridos dentro do prazo. A terceira representa a diferença de tratamento que o sistema atende os usuários, pelo ponto de vista da satisfação, ou seja, se a média de satisfação não esconde usuários com baixa satisfação no sistema.

O *deadline* é o limite superior para a média do tempo de resposta a ser garantido às requisições de um determinado usuário. Essa métrica, denominada, neste texto, tempo médio de resposta contratado (ou simplesmente contrato) pela enésima classe (T_{c_i}), é especificada previamente por um acordo entre o provedor de serviços e o usuário, e utilizado pelo escalonador como base para atribuição de prioridades. Para tanto, o valor da métrica, deve ser observado pelo servidor durante uma sessão.

O estabelecimento de contratos utilizado em Casagrande (2007) para os experimentos é a porcentagem de variação de contrato (P) em relação ao valor de latência média oferecida durante a simulação com a disciplina FIFO com os mesmos parâmetros de entrada.

Primeiramente é calculada a média de tempo de resposta utilizando a disciplina FIFO (L_{FIFO}) para o mesmo cenário de utilização da EBS. A política FIFO se caracteriza por atender prioritariamente as requisições que chegaram ao sistema mais cedo. A partir de L_{FIFO} são calculados dois contratos A e B de acordo com as equações 2.16 e 2.17, respectivamente.

$$L_{c_A} = L_{FIFO} - (L_{FIFO} \cdot P) \tag{2.16}$$

$$L_{c_B} = L_{FIFO} + (L_{FIFO} \cdot P) \tag{2.17}$$

Onde L_{c_A} e L_{c_B} são os contratos da classe mais prioritária e menos prioritária, respectivamente. Esse método estabelece a criação de dois contratos, através dos quais duas classes de usuários são formadas.

Os experimentos mostraram que na maioria dos cenários simulados os parâmetros contratuais de serviços de um sistema foram garantidos ao longo do tempo em patamares iguais ou abaixo dos contratados.

2.5.3 Trabalhos Relacionados

Em Peixoto (2008) é desenvolvido um estudo no qual se estende a abordagem de Casagrande (2007), apresentando e comparando políticas de escalonamento que têm por objetivo prover QoS absoluta para um *array* de servidores *Web* heterogêneos. Para isso, utiliza-se uma arquitetura de escalonamento ortogonal, na qual além da política de escalonamento de requisições atuar na ordenação da fila de *jobs*, uma política de escalonamento de recursos atua atribuindo-se o *job* ao recurso (processador) em que será executado.

Em Tott (2008) é investigado o impacto da carga gerada por cada cliente no sistema como um todo e como essa carga influencia o serviço oferecido a outros clientes utilizando a política EBS. Sendo assim, apresenta-se um mecanismo de controle de admissão de requisições, capaz de administrar o nível de degradação do sistema, isolando o efeito do comportamento de um usuário sobre a qualidade de serviço oferecida aos demais.

Em Nery (2009) é analisada a dispersão dos tempos de respostas obtidos com a política EBS em um servidor *Web*. Embora a meta de manter o tempo médio de resposta efetivamente abaixo do limite estipulado em contrato tenha sido alcançada nos experimentos realizados em Casagrande (2007), nenhuma restrição é aplicada à dispersão dos mesmos.

Considerando-se duas situações de dispersão: a primeira delas, em que os tempos de resposta efetivos, em uma determinada janela considerada, estejam dispersos em torno da média, formando uma distribuição normal; a segunda, em que os tempos de resposta estejam concentrados em duas regiões distantes da média, uma à direita e outra à esquerda desta. Em ambas as situações, o tempo médio de resposta efetivamente praticado pode ser o mesmo, indicando que o contrato foi atendido. No entanto, na primeira situação o usuário recebe uma qualidade de serviço aproximadamente constante ao longo do tempo, enquanto que na segunda, há momentos em que a qualidade oferecida pode exceder a expectativa, e momentos onde o serviço é percebido como muito ruim. Isso acontece, por exemplo, quando um usuário é atendido com uma média muito inferior à contratada, durante um determinado período em que a carga do sistema esteja baixa, o que resulta em uma grande folga em seu contrato, permitindo à regra de tomada de decisão, eventualmente praticar reiteradamente altos tempos de resposta para o usuário, sem comprometer sua média efetiva.

Em Saito (2009) é investigado o comportamento do algoritmo EBS em cenários de carga onde uma classe prioritária recebe atendimento pior que as demais classes. Em contratos de QoS absoluta, onde não existe qualquer relação de prioridade entre classes, isso não constitui dificuldade. Nesses casos, a relação dos tempos de resposta entre as classes de serviço pode ser qualquer, podendo inclusive inverter-se ao longo do tempo, desde que os limites superiores estabelecidos para cada uma delas sejam respeitados.

Embora isso não seja um problema para o caso de QoS absoluta, é plausível considerar qual a percepção dos usuários familiarizados ao modelo de atendimento preferencial da QoS relativa. Se uma classe com contrato mais estrito (menor tempo médio de resposta contratado) demanda mais recursos do sistema, e por isso deva arcar com custos maiores, prover-lhe um tempo de resposta superior pode se apresentar intuitivamente como uma situação de conflito. Quando o contrato de QoS é estabelecido de modo relativo, tal que é prometido a uma classe um serviço "melhor" que à outra, a mesma situação denomina-se "inversão de prioridade" e constitui uma falha no atendimento às especificações. Para o efeito da elaboração de modelos de negócios de provedores de serviços, seria conveniente investigar a possibilidade de evitar tal circunstância, atendendo, assim, às expectativas dos usuários.

Em Mamani (2010) foi desenvolvido um protótipo de servidor *Web* distribuído com diferenciação de serviços em QoS relativa e absoluta, na qual foi implementada a EBS. Nesse trabalho foi transposta a política EBS para o mundo real e comparada com os resultados das simulações.

Resultados demonstram características interessantes, por exemplo, em provisão de QoS em sistemas Web distribuídos. Algumas questões acerca do desempenho do sistema foram levantadas durante esses estudos. Este trabalho tem como motivação investigar essas questões.

A arquitetura tem como objetivo (buscar) garantir o atendimentos de SLAs (limite superior do TMR) para diversos usuários, independentemente (QoS absoluta). Para um sistema com uma dada carga de trabalho

- faz diferença o número de contratos diferentes que ele gerencia?
- faz diferença o quanto parecidos ou diferentes são os contratos?

Essas questões são de interesse na elaboração do modelo de negócios e do planejamento da capacidade do provedor do serviço.

Essa seção apresentou a política de escalonamento EBS, princípio básico para o entendimento da proposta desse trabalho. Foi explicado o funcionamento da política (Seção 2.5.1), juntamente com suas características, como as simulações vêm sendo feitas para a análise do comportamento da EBS (Seção 2.5.2) e como está o andamento das propostas para a melhoria do algoritmo (Seção 2.5.3).

No Capítulo 3 será apresentado a metodologia de desenvolvimento desse trabalho e os resultados obtidos.

CAPÍTULO

3

Resultados

Neste trabalho é realizado o estudo e análise da política de escalonamento EBS, com o intuito de obter resultados e tirar conclusões acerca da influência dos diversos fatores no sistema. Essas influências poderão ser utilizadas como base para a previsão do comportamento do sistema, possibilitando assim manter a satisfação dos clientes saudável mesmo com variações nesses fatores.

A partir dos conceitos dos modelos de fila (Seção 2.3.2), foi construído, por Casagrande (2007), um modelo que representa um servidor com suporte à QoS. Utilizou-se como abordagem o modelo e simulação orientados a eventos, discreto, dinâmico, estocástico e de tempo simulado. Basicamente, esse modelo descreve os principais eventos que ocorrem em um sistema real do gênero.

O servidor modelado é do tipo monoprocessado, com uma única fila de espera e com atendimento de requisições do tipo não-preemptivo como pode ser observado pela Figura 3.1. Nele foi implementada a política de escalonamento EBS. Para a avaliação do modelo e da política de escalonamento utilizou-se o pacote de simulação Simpack (Fishwick, 1992; Cubert e Fishwick, 1995), o qual oferece um conjunto próprio de ferramentas para simulação orientadas a eventos. O Simpack foi desenvolvido a partir da biblioteca SMPL (MacDougall, 1989) e possui uma biblioteca composta por uma ampla gama de métodos voltados à simulação de redes de filas (Kumar e Majhi, 2004), orientada a eventos discretos. O simulador é distribuído com seu código fonte completo, incluindo inúmeros exemplos de aplicações.

Uma métrica de análise é a taxa de utilização (ou carga) imposta ao sistema. Por meio do valor esperado dos tempos de serviço (C) e intervalos de chegada (A) das requisições dos usuários, é possível descrever a taxa de utilização (U) do sistema por meio da Equação 3.1. Nela observa-se que quanto maior for o intervalo entre as chegadas, mais ocioso o sistema ficará. Em contrapartida,

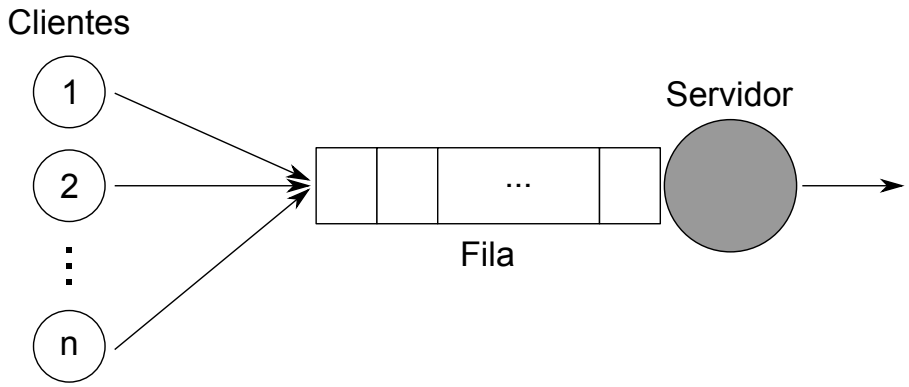


Figura 3.1: Modelo de servidor sequencial

a utilização do sistema cresce proporcionalmente ao aumento do tempo de processamento esperado para requisições que chegam.

$$U = \frac{C}{A} \tag{3.1}$$

A Equação 3.2 mostra como é calculado o índice de satisfação do usuário i, onde se relaciona a quantidade total de requisições submetidas por i (R_i) e o número de vezes em que a média de latência de sistema dessas requisições ficou abaixo do limiar contratado (N_i). Quanto mais próximo de R_i for o valor de N_i, maior será a satisfação proporcionada ao usuário i. Considera-se que um usuário está satisfeito com o serviço a ele oferecido quando obtiver uma alta porcentagem de requisições atendidas, em média, acima do limiar de qualidade contratado.

$$S_i = \frac{N_i}{R_i} \tag{3.2}$$

3.1 Metodologia de Desenvolvimento

Para avaliação da eficiência e desempenho do algoritmo EBS foi utilizado o simulador desenvolvido por Casagrande (2007). A simulação permite maior flexibilidade para o estudo, além de obter boas estimativas de comportamento do sistema após modificações. A partir dos resultados, pode-se construir modelos mais completos e representativos do problema real.

A técnica empregada para o estabelecimento de contratos explicado na Seção 2.5.2 se aplica a somente dois contratos diferentes, ou seja, com suporte a duas classes de serviços. A fim de incluir n contratos no sistema (e não somente duas classes de serviço com vários usuários), foram feitas algumas modificações no simulador. Além de dar suporte a vários contratos diferentes, foram adicionadas, ao simulador, novas funcionalidades, que garantiram um ambiente de teste compatível para os propósitos desse trabalho e de trabalhos relacionados. Além disso, foi retirada a grande necessidade de intervenção humana que o simulador requeria, tornando os experimentos mais independentes. Dentre as novas funcionalidades:

- Variáveis de entradas são lidas de um arquivo;
- Possibilidade de escolha, dentre diversos tipos de distribuição, para o tempo de chegada, serviço e saída de uma requisição;
- Possibilidade de troca da semente inicial da função de aleatoriedade, caso queira resultados diferentes;
- Possibilidade de escolha de contratos fixos ou baseado em outro algoritmo de escalonamento;
- Possibilidade de escolha de até 2 instantes de tempo (que podem ser estendidos) de mudança no comportamento do sistema, caso algum fator mude durante a execução do sistema;
- Melhorias na apresentação da saída, simplificando a análise por ferramentas automatizadas (por exemplo, o *R*[1]); dentre outros.

Sintetizou-se uma carga de trabalho (U) com distribuição exponencial para descrever tanto os tempos de chegada (A) quanto o tempo de execução (C) das requisições. Essa é uma distribuição amplamente utilizada para analisar sistemas de filas (MacDougall, 1989). U e A são descritos com média em unidades de tempo ($u.t.$). Por exemplo, intervalos de chegada com média $4u.t.$ e tempo de execução com média $3u.t.$ possuem taxa de utilização média de 75% (Equação 3.1).

Para analisar o comportamento do sistema, foram escolhidos **3** parâmetros de resposta:

- **Média de Latência do Sistema**: representa o tempo médio de residência no sistema das requisições dos usuários;
- **Média da Satisfação**: representa a média de satisfação dos usuários do sistema (Equação 3.2); e
- **Variação da Satisfação**: representa o desvio padrão da satisfação dos usuários do sistema.

Em relação à média de latência do sistema, em seu cálculo estão inclusos os tempos em fila das requisições, sendo assim esse é um bom parâmetro de serviço, pois verifica-se o nível de influência que o escalonador sofre com a mudança no ambiente. A média da satisfação é um importante parâmetro, pois determina a satisfação dos usuários em termos absolutos. Já a variação da satisfação nos mostra o quão diferente o escalonador trata os diversos usuários do sistema.

Foram usados quatro fatores de entrada no experimento:

- **Taxa de utilização do sistema**: representa a porcentagem de tempo em que o sistema fica ocupado processando as requisições;

[1] é um software livre para computação estatística e geração de gráficos

- **Número de contratos**: representa a quantidade de contratos simultâneos no sistema;
- **Média dos contratos**: representa o tempo médio máximo de resposta a uma requisição do usuário; e
- **Dispersão dos contratos**: representa o desvio padrão dos contratos escolhidos. Neste trabalho, esse fator foi representado como uma porcentagem em relação à média dos contratos.

Em relação à taxa de utilização do sistema, para seu cálculo é usada a Equação 3.1, sendo que, para a variação da utilização do sistema altera-se somente a frequência de chegada das requisições, mantendo-se fixo o tempo esperado de processamento. Isso é feito para que o cálculo de influência leve em consideração somente o tempo de fila das requisições, já que o tempo de processamento está fora do alcance da otimização do escalonador. Assim, para aumentar a taxa de utilização, é diminuída a taxa de chegada das requisições sem alterar o tempo de processamento das requisições e vice-versa. O número de contratos nos experimentos é igual ao número de usuários, ou seja, existe um contrato para cada um dos usuários, inclusive podendo ser todos diferentes. A média e a dispersão dos contratos são importantes fatores, pois influenciam diretamente no deadline das requisições, assim afetando o comportamento do escalonador.

Para realizar o estudo da influência dos fatores de entrada foi utilizado o 2^K *fatorial design* (Seção 2.4), com $K = 4$.

3.2 Resultados

Para cada um dos experimentos realizados nesse trabalho de mestrado, foram feitas **100** replicações. Há uma grande quantidade de replicações para gerar intervalos de confiança baixos e com isso fazer a análise com segurança. Em cada um desses testes são realizadas **100000** requisições no sistema. Para a validação estatística, todas as simulações foram executadas utilizando-se diferentes sementes[2] disponibilizadas pelo Simpack. Todos os intervalos de confiança foram calculados com grau de confiança de **95%**, com **99** graus de liberdade através do teste *t de Student*[3]. Assim, todos os resultados obtidos foram analisados, segundo a média e o intervalo de confiança.

3.2.1 Análise da influência

A análise da influência dos fatores foi realizada com o objetivo de ter uma ideia geral de como os fatores influenciam, separadamente ou conjuntamente, nos fatores de saída. Para isso foram escolhidos 2 níveis arbitrariamente para cada um dos 4 fatores, de modo a não sobrecarregar o sistema. Foram escolhidos os seguintes níveis:

[2]Replicações com fluxos de números aleatórios diferentes.

[3]ou simplesmente distribuição *t*. É uma distribuição de probabilidade para estimar a confiabilidade da média obtida de uma população com distribuição normal.

- Taxa média de utilização do sistema: **60%** e **65%**;
- Número de contratos: **50** e **250**;
- Tempo médio dos contratos: **150** e **250**;
- Dispersão dos contratos: **5%** e **25%**.

Esses níveis representam bem o comportamento geral de um sistema não sobrecarregado, ou seja, sem valores extremos. Assim, para realizarmos o método fatorial completo, precisamos combinar os níveis um a um, tendo no total 2^4 experimentos diferentes, que estão representados na Tabela 3.1.

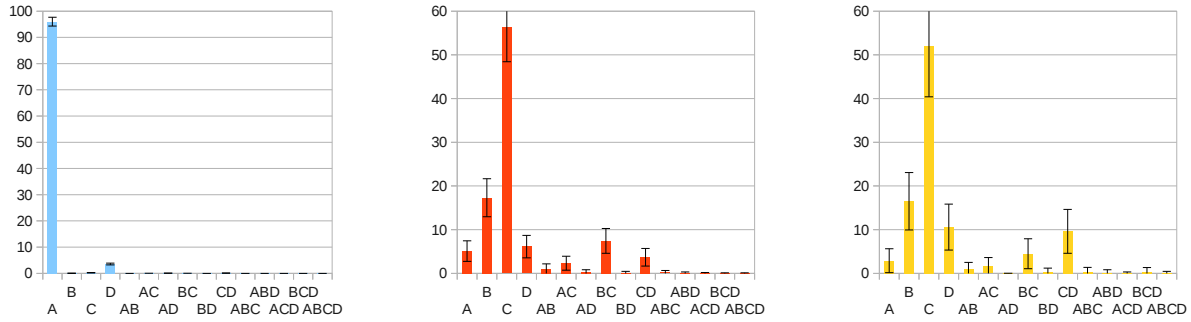
Tabela 3.1: Experimentos

Experimento	Taxa de Utilização	Número de Contratos	Média dos Contratos	Dispersão dos Contratos
1	60%	50	150	5%
2	60%	50	150	25%
3	60%	50	250	5%
4	60%	50	250	25%
5	60%	250	150	5%
6	60%	250	150	25%
7	60%	250	250	5%
8	60%	250	250	25%
9	65%	50	150	5%
10	65%	50	150	25%
11	65%	50	250	5%
12	65%	50	250	25%
13	65%	250	150	5%
14	65%	250	150	25%
15	65%	250	250	5%
16	65%	250	250	25%

A Figura 3.2 é composta por 3 gráficos que representam a influência (eixo das ordenadas), em porcentagem, de cada um dos fatores de entrada (eixo das abscissas). Os Gráficos 3.2(a), 3.2(b) e 3.2(c) mostram as influências no tempo médio de resposta, na satisfação média dos usuários e na variação da satisfação, respectivamente. Os gráficos foram alinhados lado-a-lado para melhor comparação. São utilizadas letras de A até D para representar os fatores de entrada, sendo que os fatores com múltiplas letras representam a interação entre os fatores. Eles são representados da seguinte forma:

- **A**: taxa média de utilização do sistema;
- **B**: número de contratos;
- **C**: média dos contratos;

- **D**: dispersão dos contratos.



(a) Tempo de Resposta (b) Satisfação dos Usuários (c) Variação da Satisfação

Figura 3.2: Gráfico da influência dos fatores.

Para consulta, os resultados em forma numérica são mostrados em 2 tabelas no apêndice. A Tabela A.1 mostra a média das 100 replicações para cada um dos 16 experimentos realizados, juntamente com seus respectivos intervalos de confiança. A Tabela A.2 mostra as influências dos fatores de entrada, com precisão de 4 casas decimais.

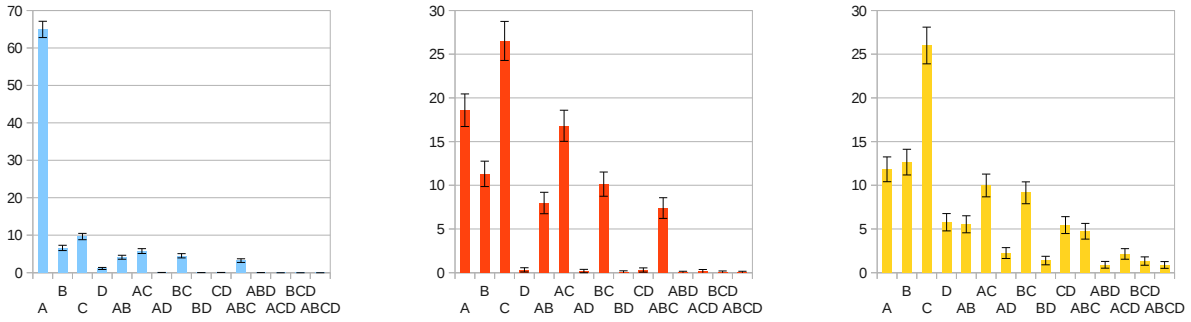
O Gráfico 3.2(a) indica que a taxa de utilização é o fator que tem a maior influência (fator A, com $95,98\%$) no tempo de resposta dentre os fatores analisados, ofuscando todos os outros. Por outro lado, os Gráficos 3.2(b) e 3.2(c) indicam uma menor influência da taxa de utilização na satisfação e na variação da satisfação dos usuários ($5,10\%$ e $2,90\%$, respectivamente). Isso acontece, pois em um sistema não sobrecarregado a taxa de utilização irá afetar diretamente o tempo médio de resposta do sistema, ou seja, uma baixa na taxa de utilização vai forçar uma baixa no tempo médio de resposta do sistema, independentemente de o sistema estar pouco ou muito carregado. Já o mesmo não acontece com a satisfação dos usuários, pois o sistema consegue manter uma boa satisfação, mesmo aumentando a taxa de utilização. É notada uma variação da satisfação com a variação na taxa de utilização (a influência não é estatisticamente nula), porém o efeito é bem menor em comparação com o que acontece com tempo de resposta. O mesmo aplica-se à variação da satisfação.

Para verificar se os fatores influem diferentemente em sistemas pouco ou muito carregados, dois novos níveis de taxas de utilização foram fixados (80% e 85%). Foi tomado o cuidado de variar os níveis na mesma proporção que o experimento anterior, nesse caso em 5%. Para os outros fatores foram mantidos os níveis do experimento anterior, como mostrado na Tabela 3.1, possibilitando uma comparação direta entre os dois experimentos. A tabela 3.3 mostra quais foram os experimentos relizados. A Figura 3.3 mostra os resultados do método fatorial completo aplicado a esse ambiente. Para consulta, os resultados e as influências são mostrados em forma numérica nas tabelas A.3 e A.4, respectivamente.

O Gráfico 3.3(a) indica que a influência da taxa de utilização (fator A) no tempo de resposta é $64,94\%$, que é menor em relação ao que mostrava o Gráfico 3.2(a) ($95,98\%$). Também podemos

Tabela 3.2: Experimentos

Experimento	Taxa de Utilização	Número de Contratos	Média dos Contratos	Dispersão dos Contratos
1	80%	50	150	5%
2	80%	50	150	25%
3	80%	50	250	5%
4	80%	50	250	25%
5	80%	250	150	5%
6	80%	250	150	25%
7	80%	250	250	5%
8	80%	250	250	25%
9	85%	50	150	5%
10	85%	50	150	25%
11	85%	50	250	5%
12	85%	50	250	25%
13	85%	250	150	5%
14	85%	250	150	25%
15	85%	250	250	5%
16	85%	250	250	25%



(a) Tempo de Resposta (b) Satisfação dos Usuários (c) Variação da Satisfação

Figura 3.3: Gráfico da influência dos fatores em um ambiente com maior taxa de utilização

notar uma diferença nos outros fatores, indicando que os fatores influem diferentemente dependendo do estado do sistema. Os Gráficos 3.3(b) e 3.3(c) também mostram diferenças relação aos Gráficos 3.2(b) e 3.2(c). Na seção seguinte (3.2.2) é feita uma análise completa, fator a fator, de como essas mudanças afetam o sistema.

3.2.2 Análise da influência fator a fator

Essa análise tem por objetivo verificar como a influência de um fator reage com a mudança de outro fator. Por exemplo, como a influência do número de contratos muda com o aumento da taxa de utilização. Com isso podemos verificar se a influência de um fator aumenta ou diminui conforme a taxa de utilização varia e também em que taxa ocorre tal mudança. O primeiro experimento

consiste em mudar uniformemente a taxa de utilização e verificar como as influências dos outros fatores se comportam.

Variação da taxa de utilização

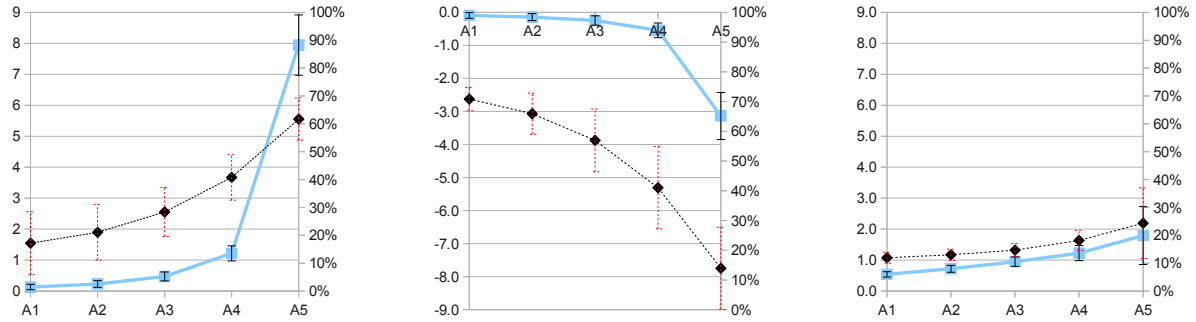
Para fazer essa verificação foram feitos experimentos com taxa de utilização de **60%**, **65%**, **70%**, **75%**, **80%** e **85%**. Foi tomado o cuidado de variar esse fator uniformemente, neste caso em passos de 5%, provendo um aumento linear nesse fator e assim facilitando a posterior análise. Foram feitas 5 análises utilizando o método fatorial completo, de acordo com os níveis da Tabela 3.3. Essa tabela mostra 5 linhas representando cada um dos experimentos, e cada coluna mostra os dois níveis analisados separados por barra ("/"). Os níveis não mudam entre as experiências (com exceção da taxa de utilização, que é o fator variante neste caso) e foram escolhidos de modo a representar um sistema variando de um ambiente pouco exigente para um com maior exigência (do experimento $A1$ para o $A5$). Porém, não é possível a comparação direta da influência entre os 5 experimentos, pois como a obtenção das influências através do método fatorial completo é em porcentagem, ela compara os fatores de um mesmo experimento. Em outras palavras, tomando como exemplo a variação do fator A do Gráfico 3.2(a) $(95,98\%)$ para o 3.3(a) $(64,94\%)$, pode ter ocorrido **duas** situações. Uma análise mais superficial pode nos levar a pensar que a influência do fator A diminuiu entre os experimentos, porém pode ser que a influência aumentou, porém os outros fatores tiveram aumentos proporcionalmente maiores.

Tabela 3.3: Experimentos com mudança da taxa de utilização

Exp.	Taxa de Utilização	Número de Contratos	Média dos Contratos	Dispersão dos Contratos
A1	60% / 65%	2 / 200	125 / 250	5% / 30%
A2	65% / 70%	2 / 200	125 / 250	5% / 30%
A3	70% / 75%	2 / 200	125 / 250	5% / 30%
A4	75% / 80%	2 / 200	125 / 250	5% / 30%
A5	80% / 85%	2 / 200	125 / 250	5% / 30%

Assim, à primeira vista, não podemos identificar se o fator A do Gráfico 3.3(a) diminuiu ou cresceu menos que os outros fatores. Para essa análise é necessário observar os valores absolutos do método fatorial completo, mais explicitamente os valores dos $q_i's$ mostrados na Equação 2.6 da Seção 2.4. Com esses valores podemos comparar a influência entre experimentos diferentes.

A primeira variável de resposta analisada é o tempo médio de resposta. A Figura 3.4 mostra um gráfico para cada um dos 3 fatores de entrada analisados neste experimento (número de contratos, média dos contratos e dispersão dos contratos, respectivamente). Nesse caso, não foi analisada a taxa de utilização, pois é o fator que está variando. O eixo das abscissas representa cada um dos 5 experimentos (Tabela 3.3) enquanto o eixo das ordenadas representa os valores dos $q_i's$ com seus respectivos intervalos de confiança. Os valores numéricos são apresentados na Tabela A.5 com precisão de 5 casas decimais.



(a) Número de Contratos (b) Média dos Contratos (c) Dispersão dos Contratos

Figura 3.4: Influência no tempo de resposta com a variação da taxa de utilização

Podemos verificar a evolução da influência dos fatores de entrada enquanto a taxa de utilização aumenta (de $A1$ para $A5$). Quanto mais afastado de 0, mais um fator influi no sistema, independentemente se é negativo ou positivo. Uma influência "negativa" atua como "diminuidor" da variável de resposta, da mesma forma de uma influência "positiva" atua como "aumentador" da variável de resposta. Nesse caso, um aumento no número de contratos (Gráfico 3.4(a)) ou na dispersão dos contratos (Gráfico 3.4(c)) irá afetar aumentado o tempo médio de resposta (influência positiva). Em contrapartida, o aumento da média dos contratos (Gráfico 3.4(b)) irá afetar diminuindo o tempo médio de resposta (influência negativa).

O gráfico pontilhado representa, em porcentagem, sua influência em relação aos outros fatores. Daí podemos notar que a influência do número de contratos aumenta em relação aos outros fatores quando a taxa de utilização aumenta. Já a média dos contratos faz o caminho inverso, diminuindo. A influência da dispersão dos contratos se mantém.

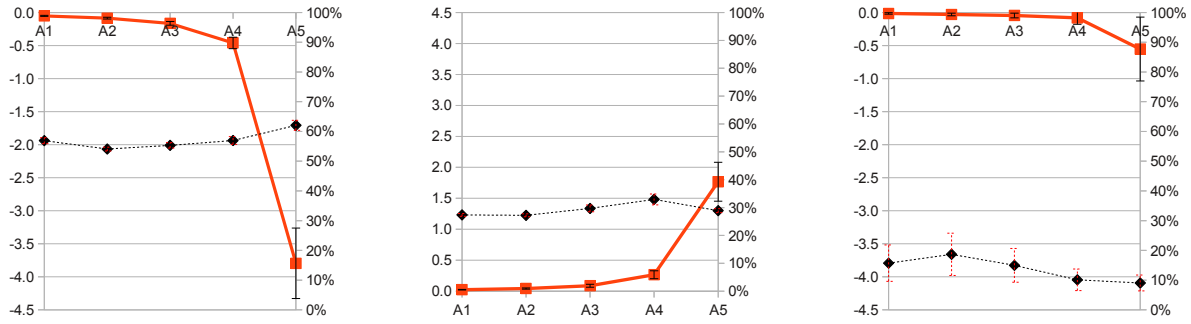
Foi observada uma situação similar nos Gráficos 3.4(a) e 3.4(b), onde o aumento linear da taxa de utilização (de $A1$ para $A5$) é acompanhado com um aumento exponencial da influência desses fatores. No Gráfico 3.4(c) há também um aumento, porém um aumento menor se comparado com os outros dois fatores, visualmente mais próximo a um aumento linear. Podemos verificar que o fator que foi mais influenciado é o número de contratos, pois a sua influência se tornou mais distante de 0 que os demais. Já a média dos contratos é o segundo fator que foi mais influenciado e por último a dispersão dos contratos.

Assim, enquanto o sistema está com baixa taxa de utilização há pouca influência do número de contratos e da média dos contratos (chegando a ser menor que a influência da dispersão dos contratos), porém à medida que aumenta a taxa de utilização, mais esses fatores irão afetar o sistema. Em outras palavras, o número de contratos e a média dos contratos pouco devem ser levados em consideração em sistemas com pouca taxa de utilização e devem ser levados em consideração em sistemas com alta utilização.

A segunda variável de resposta analisada é a satisfação dos usuários. A análise mostrou resultados bastante similares se comparado com a primeira variável de resposta analisada (tempo

médio de resposta). A Figura 3.5 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.6. Nesse experimento houve também um aumento exponencial com o aumento linear da taxa de utilização. A principal diferença (em relação ao experimento anterior) é a inversão nas influências, ou seja, os gráficos do número de contratos (3.5(a)) e da dispersão dos contratos (3.5(c)) mostram influências negativas e o gráfico da média dos contratos (3.5(b)) mostra uma influência positiva. Isso mostra que os fatores que aumentam o tempo de resposta diminuem a satisfação dos usuários. Isso era um resultado esperado, pois em um sistema que tem um aumento no tempo de resposta, tende a diminuir a satisfação dos usuários. Assim como na análise anterior, o número de contratos foi o fator mais influenciado com a variação da taxa de utilização, seguido da média dos contratos e por último a dispersão dos contratos.

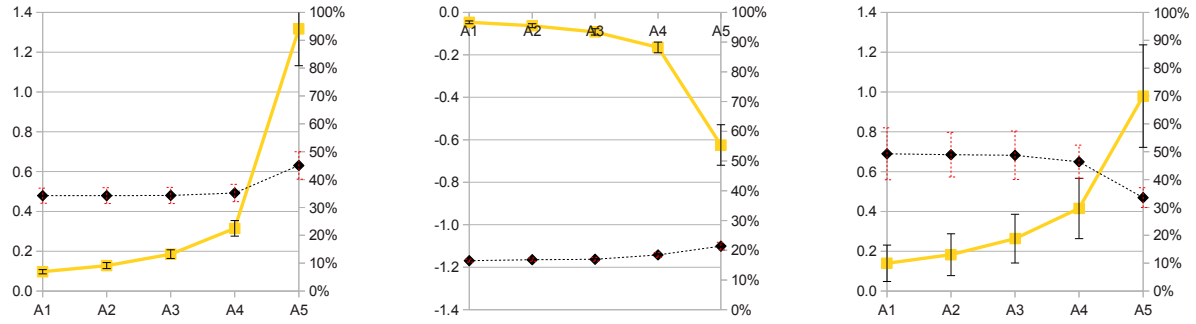
Também é notável que os fatores influenciem mais no tempo de resposta do que na satisfação dos usuários se compararmos a escala dos gráficos da Figura 3.4 com os da Figura 3.5. Isso quer dizer que o escalonador consegue manter a satisfação, mesmo com o tempo médio de resposta aumentando.



(a) Número de Contratos (b) Média dos Contratos (c) Dispersão dos Contratos

Figura 3.5: Influência na satisfação dos usuários com a variação da taxa de utilização

A terceira variável de resposta analisada é a variação da satisfação dos usuários. A Figura 3.6 mostra os resultados da análise e os valores numéricos são mostrados na Tabela A.7. Nesse caso as influências têm a mesma sinalização dos gráficos mostrados na Figura 3.4. Isso quer dizer que os 3 fatores influem a variação da satisfação dos usuários da mesma forma que o tempo médio de resposta, ou seja, o número de contratos e a dispersão dos contratos aumentam a dispersão da satisfação enquanto a média dos contratos diminui a dispersão da satisfação. O número de contratos foi o fator mais influenciado, porém diferente dos experimentos anteriores, a dispersão dos contratos teve uma influência maior do que a média dos contratos, como pode ser visto comparando os Gráficos 3.6(b) e 3.6(c). Assim a variação da satisfação dos usuários teoricamente tenderá a ser mais responsivo à dispersão dos contratos do que com a média dos contratos. A média dos contratos afeta a satisfação como um todo, enquanto a dispersão dos contratos afeta o quão diferente os contratos são e consequentemente o quão diferente o sistema irá tratar os seus usuários.



(a) Número de Contratos (b) Média dos Contratos (c) Dispersão dos Contratos

Figura 3.6: Influência na dispersão da satisfação dos usuários com a variação da taxa de utilização

Portanto, o que pode ser observado nos três experimentos, é que a mudança na taxa de utilização muda a influência dos fatores diferentemente em grau e sinal. A influência varia de acordo com os gráficos apresentados nas Figuras 3.4, 3.5 e 3.6. A influência dos 3 fatores analisados é baixa quando o sistema está com pouca taxa de utilização, aumentando exponencialmente quando a taxa de utilização aumenta linearmente. Em relação ao tempo médio de resposta e à satisfação dos usuários, o número de contratos é o fator que foi mais influenciado com o aumento da taxa de utilização, seguido da média dos contratos e por último da dispersão dos contratos. Já em relação à variação da satisfação dos usuários, o número de contratos também é o fator mais influenciado, porém é seguido da dispersão dos contratos e por último a média dos contratos. Quanto maior a escala do gráfico, maior será a diferença entre os experimentos, portanto maior será a variação da influência do fator.

A seguir foi feito a interação entre todos os fatores, ou seja, como os fatores se comportam com a mudança no número de contratos, na média dos contratos e na dispersão dos contratos. Assim temos uma análise completa da interação entre os fatores, complementando o estudo feito variando a taxa de utilização. Dessa forma, realizou-se a mesma análise, com a variação dos fatores restantes. Como são experimentos parecidos, as conclusões tiradas desse experimento podem ser levadas para os seguintes.

Variação do número de contratos

Nessa seção, o fator variado é o número de contratos. Foram utilizados 6 níveis: **2**, **50**, **100**, **150**, **200** e **250**. Assim como no experimento anterior foi tomado o cuidado de variar os níveis uniformemente (em passos de 50), com exceção do primeiro nível, pois foi julgada desprezível essa diferença. Assim teremos os experimentos mostrados na Tabela 3.4.

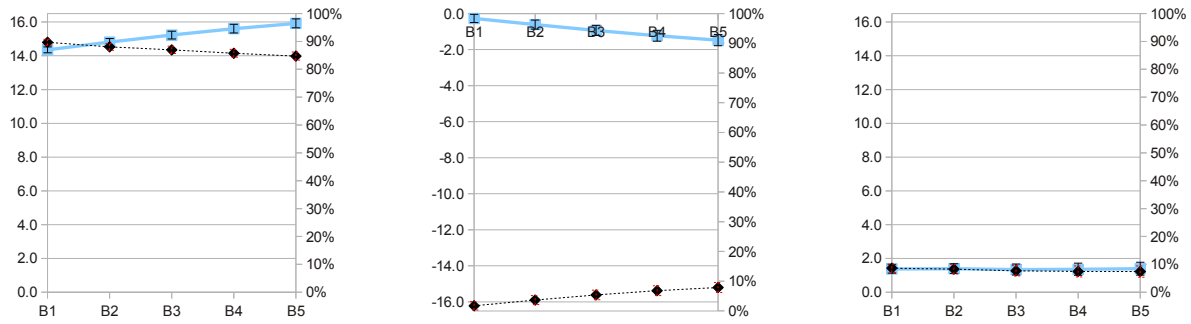
A primeira variável de resposta analisada é o tempo médio de resposta. A Figura 3.7 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.8. Nos Gráficos 3.7(a) e 3.7(b) nota-se um aumento na influência com o aumento do número de contratos (do experimento

Tabela 3.4: Experimentos com mudança no número de contratos

Exp.	Taxa de Utilização	Número de Contratos	Média dos Contratos	Dispersão dos Contratos
B1	60% / 80%	2 / 50	125 / 250	5% / 30%
B2	60% / 80%	50 / 100	125 / 250	5% / 30%
B3	60% / 80%	100 / 150	125 / 250	5% / 30%
B4	60% / 80%	150 / 200	125 / 250	5% / 30%
B5	60% / 80%	200 / 250	125 / 250	5% / 30%

$B1$ para o $B5$). Diferente da variação exponencial visto no experimento anterior, nesse caso essa variação se aproxima bem de uma variação linear. Ou seja, com o aumento linear do número de contratos há um aumento linear na influência da taxa de utilização e da média dos contratos. Já o Gráfico 3.7(c) mostra que não houve variação na influência da dispersão dos contratos.

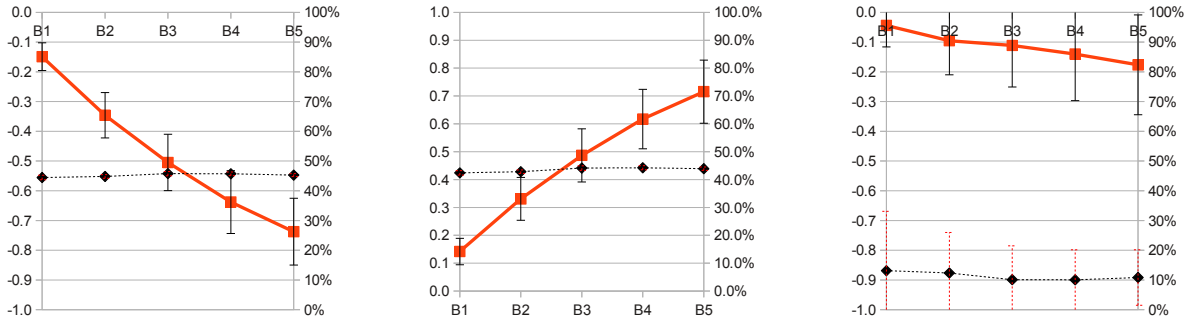
Pelos gráficos é verificado que taxa de utilização é o fator que mais foi influenciado, em seguida a dispersão dos contratos e por último a média dos contratos. Porém há uma tendência da influência da média dos contratos ficar maior que a da dispersão dos contratos à medida que o número de contratos aumenta. Assim a média dos contratos tem pouca influência em sistemas com baixo número de contratos e aumenta linearmente à medida que esse fator aumenta. Já a taxa de utilização (Gráfico 3.7(a)) tem uma influência alta, mesmo com baixo número de usuários.



(a) Taxa de Utilização (b) Média dos Contratos (c) Dispersão dos Contratos

Figura 3.7: Influência no tempo de resposta com a variação do número de contratos

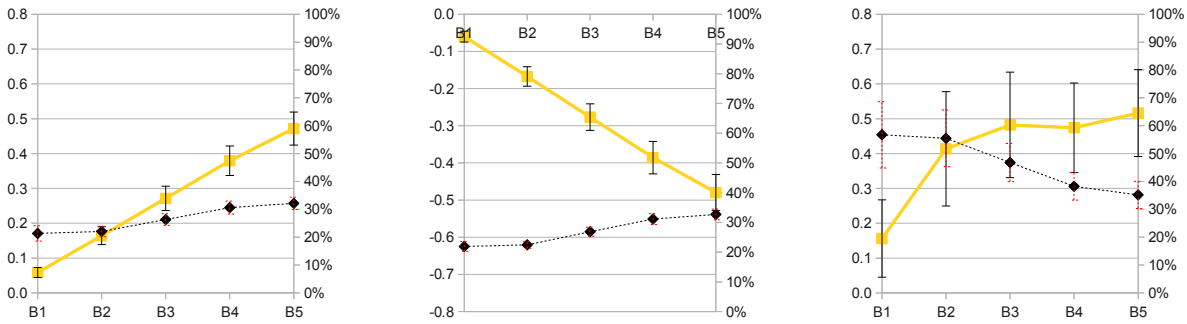
A segunda variável de resposta analisada é a satisfação dos usuários. A Figura 3.8 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.9. Aqui também foi observado um aumento linear da influência da taxa de utilização e da média dos contratos e uma estagnação da dispersão dos contratos. Nesse caso, os Gráficos 3.8(a) e 3.8(b) se mostraram bem espelhados, assim a taxa de utilização e a média dos contratos influem da mesma forma, sendo que uma positivamente e outra negativamente. Quanto ao Gráfico 3.8(c) mostra que o intervalo de confiança engloba o valor 0, o que significa que estatisticamente não há nenhuma influência desse fator na satisfação dos usuários. A taxa de utilização e a média dos contratos foram os fatores que mais influenciaram.



(a) Taxa de Utilização (b) Média dos Contratos (c) Dispersão dos Contratos

Figura 3.8: Influência na satisfação dos usuários com a variação do número de contratos

E a terceira variável de resposta é a variação na satisfação dos usuários. A Figura 3.9 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.10. Os gráficos são parecidos com os obtidos nos dois experimentos anteriores, onde os Gráficos 3.9(a) e 3.9(b) são espelhados e o Gráfico 3.9(c) mostra uma estagnação a partir do experimento $B1$. Nesse caso, a dispersão dos contratos foi o fator que mais influenciou o sistema, porém esse cenário tende a mudar a partir de $B5$, ou seja, a partir de 250 usuários, com o aumento da influência da taxa de utilização e da média dos contratos.



(a) Taxa de Utilização (b) Média dos Contratos (c) Dispersão dos Contratos

Figura 3.9: Influência na satisfação dos usuários com a variação do número de contratos

Assim a variação do número de usuários interferiu pouco na influência dos outros fatores. As influências da taxa de utilização e da média dos contratos aumentam linearmente com o aumento do número de contratos. Já a influência da dispersão dos contratos não aumenta ou há uma leve tendência de aumento com o aumento no número de contratos.

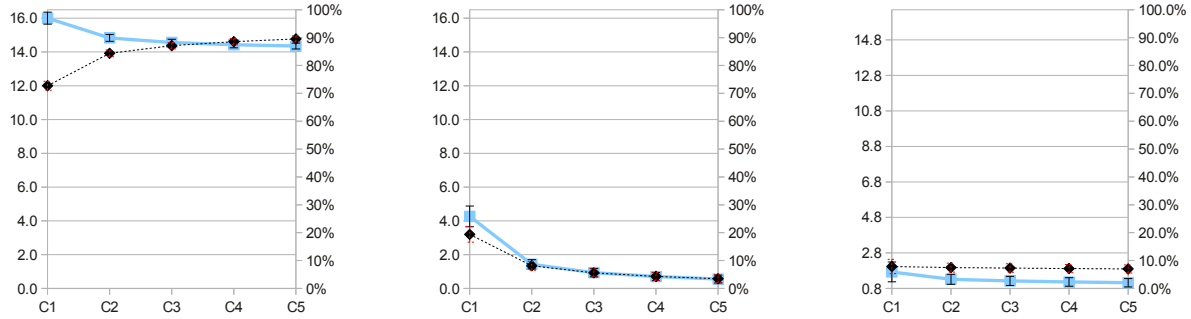
Variação da média dos contratos

Nessa seção, o fator variado é a média dos contratos. Foram utilizados os seguintes níveis: **125**, **150**, **175**, **200**, **225** e **250**. Assim teremos os experimentos mostrados na Tabela 3.5.

Tabela 3.5: Experimentos com mudança na média dos contratos

Exp.	Taxa de Utilização	Número de Contratos	Média dos Contratos	Dispersão dos Contratos
C1	60% / 80%	2 / 200	125 / 150	5% / 30%
C2	60% / 80%	2 / 200	150 / 175	5% / 30%
C3	60% / 80%	2 / 200	175 / 200	5% / 30%
C4	60% / 80%	2 / 200	200 / 225	5% / 30%
C5	60% / 80%	2 / 200	225 / 250	5% / 30%

A primeira variável de resposta analisada é o tempo médio de resposta. A Figura 3.10 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.11. Foi observado um comportamento inverso aos experimentos até então analisados. Nesse caso, o aumento da média dos contratos (de $C1$ para $C5$) diminui a influência dos outros fatores (as influências se aproximam de 0). Isso acontece, pois ao contrário dos outros fatores, um aumento na média dos contratos faz com que o sistema fique menos carregado. É observada uma diminuição logarítmica da influência dos fatores com o aumento linear da média dos contratos. O fator que foi mais influenciado é a taxa de utilização, seguido do número de contratos e por último da dispersão dos contratos. O Gráfico 3.10(c) mostra intervalos de confiança altos, indicando que a influência da dispersão dos contratos é pouco alterada entre os experimentos.

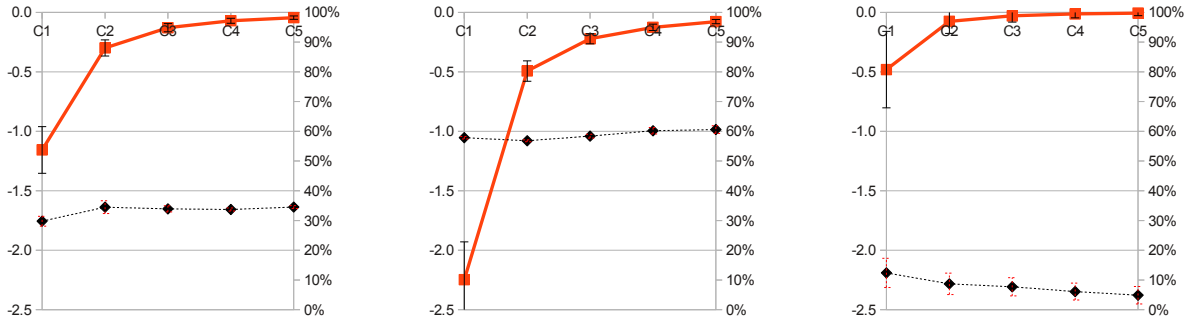


(a) Taxa de Utilização (b) Número de Contratos (c) Dispersão dos Contratos

Figura 3.10: Influência no tempo de resposta com a variação da média dos contratos

A segunda variável de resposta é a satisfação dos usuários. A Figura 3.11 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.12. Como no experimento anterior, verificamos uma diminuição logarítmica da influência dos fatores. Nesse caso, o número de contratos foi o fator que mais foi influenciado, seguido da taxa de utilização e por último a dispersão dos contratos. O Gráfico 3.11(c) mostra que a partir de $B2$ a dispersão dos contratos não tem mais influência na satisfação dos usuários do sistema. Ou seja, com contratos altos a sua dispersão não tem influência na satisfação.

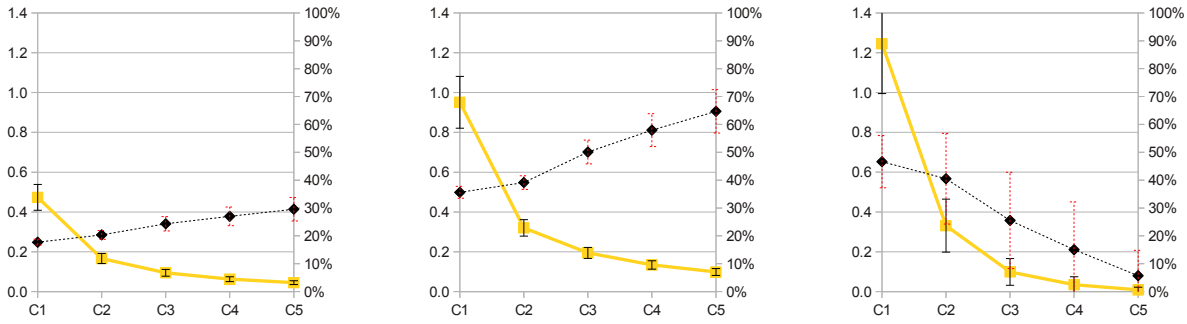
A terceira variável de resposta é a variação da satisfação. A Figura 3.12 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.13. Aqui também as influências



(a) Taxa de Utilização (b) Número de Contratos (c) Dispersão dos Contratos

Figura 3.11: Influência na satisfação dos usuários com a variação da média dos contratos

dos fatores diminuíram logaritmicamente com o aumento dos contratos. A dispersão dos contratos (Gráfico 3.12(c)) é o fator que mais perde influência, inclusive ficando estatisticamente nula a partir de $C4$.



(a) Taxa de Utilização (b) Número de Contratos (c) Dispersão dos Contratos

Figura 3.12: Influência na variação da satisfação dos usuários com a variação da média dos contratos

As influências dos fatores diminuem logaritmicamente com o aumento da média dos contratos. Em cada uma das variáveis de resposta, um fator de entrada foi o que teve mais variação na influência. No caso do tempo de resposta foi a taxa de utilização, na satisfação dos usuários foi o número de contratos e na variação da satisfação foi a dispersão dos contratos. Essa diminuição logarítmica indica que as influências tendem a estabilizar conforme a média dos contratos aumenta, ou seja, há um ponto aonde o aumento dos contratos não irá mais afetar a influência dos outros fatores.

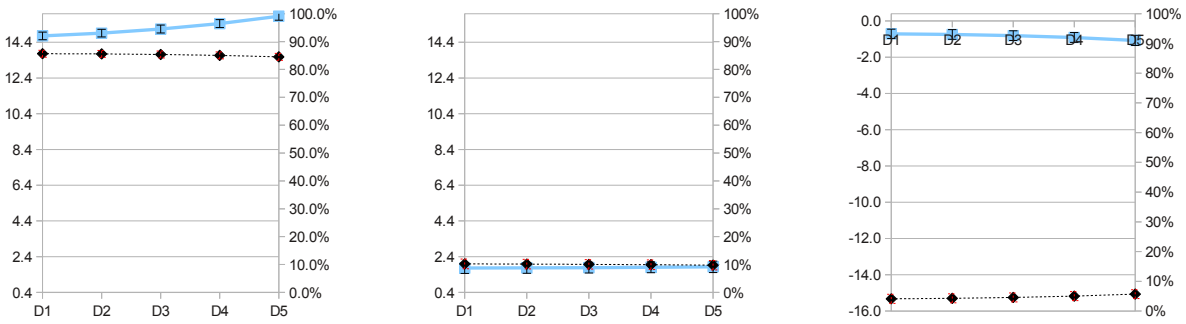
Variação da dispersão dos contratos

E por último, nessa seção o fator variado é a dispersão dos contratos. Foram utilizados os seguintes níveis: **5%**, **10%**, **15%**, **20%**, **25%** e **30%**. Assim teremos os experimentos mostrados na Tabela 3.6.

Tabela 3.6: Experimentos com mudança na dispersão dos contratos

Exp.	Taxa de Utilização	Número de Contratos	Média dos Contratos	Dispersão dos Contratos
D1	60% / 80%	2 / 200	125 / 250	5% / 10%
D2	60% / 80%	2 / 200	125 / 250	10% / 15%
D3	60% / 80%	2 / 200	125 / 250	15% / 20%
D4	60% / 80%	2 / 200	125 / 250	20% / 25%
D5	60% / 80%	2 / 200	125 / 250	25% / 30%

A primeira variável de resposta analisada é o tempo médio de resposta. A Figura 3.13 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.14. Quanto à taxa de utilização (Gráfico 3.13(a)), há uma leve tendência de aumento na sua influência. Já quanto ao número de contratos (3.13(b)) e à média dos contratos (3.7(c)) não ocorre um aumento significativo de suas influências. A taxa de utilização foi o fator que mais influenciou o tempo médio de resposta.



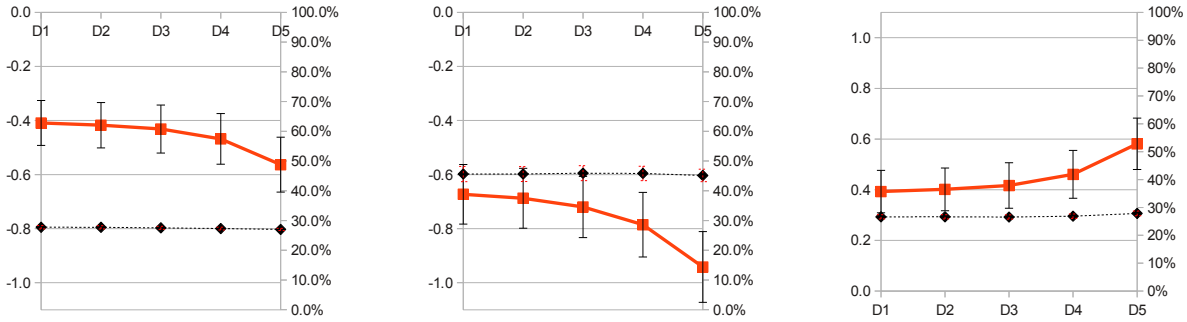
(a) Taxa de Utilização (b) Número de Contratos (c) Média dos Contratos

Figura 3.13: Influência no tempo de resposta com a variação da dispersão dos contratos

A segunda variável de resposta é a satisfação dos usuários. A Figura 3.14 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.15. Como no experimento anterior, os gráficos mostraram um intervalo de confiança bem alto. Em todos os casos ocorreu uma leve tendência de aumento, porém bem pouco se comparado com a variação dos outros fatores.

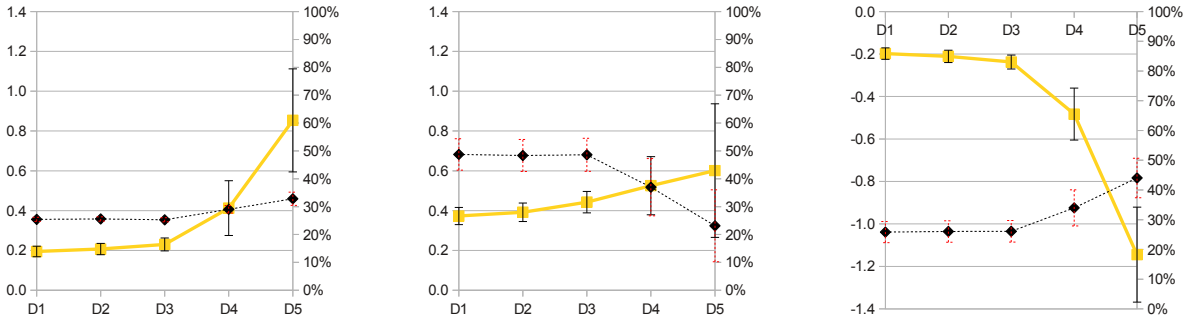
A terceira variável de resposta é a variação da satisfação. A Figura 3.15 mostra os gráficos dessa análise e os valores numéricos são mostrados na Tabela A.16. Esse experimento também mostra que há pouca mudança na influência dos fatores, indicando que o aumento da dispersão dos usuários não aumenta, ou aumenta pouco a influência dos outros fatores. No caso da influência na variação da satisfação, a média dos contratos (Gráfico 3.15(c)) é uma exceção, pois apresenta intervalos de confiança menores, indicando que a influência da média dos contratos aumenta com o aumento da variação dos contratos.

As influências dos fatores variam pouco com o aumento da dispersão dos contratos se comparadas com a variação dos outros fatores. Isso indica que a dispersão dos contratos é o fator que menos interage com os outros fatores.



(a) Taxa de Utilização (b) Número de Contratos (c) Média dos Contratos

Figura 3.14: Influência na satisfação dos usuários com a variação da dispersão dos contratos



(a) Taxa de Utilização (b) Número de Contratos (c) Média dos Contratos

Figura 3.15: Influência na variação da satisfação com a variação da dispersão dos contratos

Tabelas de síntese

Nesta seção são apresentados tabelas para sintetizar a análise feita neste capítulo.

A Tabela 3.7 mostra para cada um dos 4 fatores de entrada (linhas), se a sua influência é positiva ou negativa para as variáveis de resposta (colunas). Se a influência é positiva é simbolizada por "+", caso contrário por "−". Por exemplo, a primeira linha mostra a influência da taxa de utilização no tempo de resposta, na satisfação dos usuários e na variação da satisfação.

Tabela 3.7: Influências positivas e negativas

	Tempo de Resposta	**Satisfação dos Usuários**	**Variação da Satisfação**
Taxa de utilização	+	−	+
Número de contratos	+	−	+
Média dos contratos	−	+	−
Dispersão dos contratos	+	−	+

As Tabelas 3.8, 3.9 e 3.10 mostram o comportamento das influências com o aumento linear de um determinado fator. O símbolo de maior (">"), menor ("<") e de igualdade ("=") representa a

diminuição, o aumento e a não variação da influência, respectivamente. O aumento da influência pode ser linear (representado por "lin." na frente do símbolo), ou exponencial (representado por "exp."), por outro lado a diminuição da influência pode ser linear ou logarítmica (representado por "log"). Por exemplo, a primeira linha da Tabela 3.8 mostra como o aumento da taxa de utilização influi os outros fatores, neste caso, número de contratos, média dos contratos e dispersão dos contratos. Então temos um aumento exponencial da influência do número de contratos com o aumento linear da taxa de utilização. Essas tabelas são somente uma aproximação, já que não é explicitado quando um aumento deixa de ser linear e passa a ser exponencial, portanto essa tabela é só uma aproximação visual, e deverá ser consultada em conjunto com os gráficos. Essas tabelas foram apresentadas com o objetivo de ter uma noção geral do comportamento dos fatores.

Tabela 3.8: Variação das influências no tempo médio de resposta

	Taxa de Utilização	**Número de Contratos**	**Média dos Contratos**	**Dispersão dos Contratos**
Taxa de utilização	X	> exp.	> exp.	> lin.
Número de contratos	> lin.	X	> lin.	=
Média dos contratos	< log.	< log.	X	=
Dispersão dos contratos	> lin.	=	=	X

Tabela 3.9: Variação das influências na satisfação dos usuários

	Taxa de Utilização	**Número de Contratos**	**Média dos Contratos**	**Dispersão dos Contratos**
Taxa de utilização	X	> exp.	> exp.	> exp.
Número de contratos	> lin.	X	> lin.	=
Média dos contratos	= log.	< log.	X	< log.
Dispersão dos contratos	=	=	=	X

Tabela 3.10: Variação das influências na dispersão das satisfações

	Taxa de Utilização	**Número de Contratos**	**Média dos Contratos**	**Dispersão dos Contratos**
Taxa de utilização	X	> exp.	> exp.	> exp.
Número de contratos	> lin.	X	> lin.	=
Média dos contratos	< exp.	< exp.	X	< exp.
Dispersão dos contratos	> exp.	=	> exp.	X

3.3 Notas finais

Nesse capítulo foram apresentados os resultados dos experimentos com gráficos do comportamento de cada um dos fatores de entrada em cada um dos fatores de saída estudados. Os ex-

perimentos mostraram resultados satisfatórios, permitindo obter conclusões que até então eram cogitadas ou desconhecidas.

A análise individual dos fatores permitiu que o estudo fosse simplificado, sem detrimento de uma análise do comportamento do sistema como um todo. Os gráficos obtidos podem ser utilizados para prever o comportamento geral do sistema com a mudança de qualquer um dos fatores de entrada analisados.

O próximo capítulo apresenta as considerações finais, as contribuições desse trabalho e também os trabalhos futuros relacionados à EBS, bem como à sua melhoria e análise.

CAPÍTULO

4

Conclusão

Esse trabalho de mestrado fez uma análise das influências dos fatores em um ambiente com variações controladas, com o objetivo de possibilitar a avaliação do comportamento de um sistema de tempo real com algoritmo de escalonamento EBS, assim sendo possível a preparação do sistema para futuras modificações que venham a ocorrer no sistema ou no ambiente. Primeiramente foi constatado que a influência obtida de cada um dos fatores muda, conforme o ambiente (Seção 3.2.1). Com isso foi necessário uma variação constante do ambiente para que fosse possível uma melhor análise do comportamento desse sistema. Foram analisadas as influências de cada um dos 4 fatores escolhidos, uma a uma, e foram traçados gráficos com o comportamento obtido no decorrer de uma variação (Seção 3.2.2).

Três dos quatro fatores analisados atuam no sistema da mesma forma. A taxa de utilização, o número de contratos e dispersão dos contratos aumentam o tempo de resposta e a variação da satisfação (influência positiva) e diminuem a satisfação dos usuários (influência negativa). Já a média dos contratos atua de forma oposta, ou seja, diminuindo o tempo de resposta e a variação da satisfação e aumentando a satisfação dos usuários. Esse comportamento é o esperado, pois quanto maior a taxa de utilização, o número ou a dispersão dos contratos, teoricamente maior será a dificuldade do sistema, portanto maior será o tempo de resposta e menor será a satisfação dos usuários. Por outro lado, quanto maior a média dos contratos, menor a dificuldade do sistema, portanto menor será o tempo de resposta e maior será a satisfação dos usuários.

Todos os fatores aumentaram sua influência, em menor ou maior grau, com o aumento da taxa de utilização, número ou dispersão dos contratos. Porém esses aumentos diferem um dos outros, com alguns fatores tendo variação mais próxima do exponencial, outras mais próximas do linear e outros não tendo variação. Por outro lado todos os fatores diminuíram com o aumento da média

dos contratos. Assim a influência do fator A mostrado no Gráfico 3.2(a) não diminuiu em relação ao Gráfico 3.2(b), e sim aumentou menos que os outros fatores.

A variação da taxa de utilização resultou em gráficos com aumentos exponenciais na maioria dos casos, mostrando sua grande influência nos outros fatores. Também mostrou que o número de contratos e a média dos contratos não tem muita importância em sistemas com baixa taxa de utilização, porém tem muita importância em sistemas com alta taxa de utilização. A variação do número de contratos por sua vez gerou gráficos com aumentos lineares na taxa de utilização e na média dos contratos e nenhum aumento substantivo na dispersão dos contratos. Isso quer dizer que não importa o número de contratos, a influência da sua dispersão é sempre a mesma. A variação da média dos contratos mostrou diminuições logarítmicas na influência dos fatores analisados à medida que a média dos contratos aumenta. Essas diminuições tendem a um valor, que mostra que a partir de um determinado ponto, o constante aumento da média dos contratos não irá mais beneficiar o sistema. E por último a variação da dispersão dos contratos teve poucas mudanças nas influências na maioria dos casos, indicando que esse fator não interage fortemente com os outros fatores.

4.1 Contribuições

A contribuição principal desse trabalho foi o estudo do comportamento do algoritmo *EBS*, não somente em um ambiente estático, mas sim com variação dos fatores. Esse estudo foi necessário, pois com a mudança do ambiente é detectado mudanças nas influências dos fatores e agora com base nos gráficos apresentados, pode-se projetar o comportamento esperado do sistema.

Também este trabalho respondeu a uma série de dúvidas pertinentes ao comportamento da *EBS*. Como, por exemplo, determinou-se que a importância do número de usuários só é pertinente em ambientes com grande taxa de utilização, sendo que em ambientes com pouca taxa de utilização sua influência é quase nula. Com os gráficos pode-se estimar o comportamento esperado da influência de qualquer interação entre dois fatores.

4.2 Trabalhos Futuros

A seguir são apresentados os trabalhos futuros relacionados a este trabalho e ao algoritmo *EBS*:

- Analisar outros fatores de entrada como: tamanho da janela de requisições, número de servidores e média do tempo de serviço.
- Fazer mudança no ambiente em um mesmo experimento, e analisar o tempo de reação do sistema.

- Mudar o comportamento do sistema ao lidar com *deadlines* negativos (Equação 2.15). Com esse comportamento, uma requisição com *deadline* negativo sempre terá mais prioridade que uma requisição com *deadline* positivo, não importando o tamanho da mesma.
- Inserir o fator satisfação no cálculo da prioridade, dando aos menos satisfeitos maior prioridade. Com isso a satisfação entre os usuários tende a convergir diminuindo sua variação.
- Inserir vários tipos de usuários com comportamentos diferentes.

.

APÊNDICE

A

Tabelas

Tabela A.1: Resultados

Exp.	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
1	58,91389	0,125637	99,97902	0,006082156	0,04327253	0,007621984
2	59,87905	0,1339216	99,94328	0,03274027	0,2467878	0,1946598
3	58,87322	0,1244843	99,99823	0,0008220315	0,006741971	0,001967577
4	59,57772	0,130023	99,99724	0,001104802	0,01021258	0,002700573
5	59,04048	0,1289792	99,9266	0,01158519	0,1989858	0,0195814
6	59,95794	0,1322798	99,88898	0,01439215	0,3723157	0,05790629
7	58,94285	0,1258524	99,99079	0,002609188	0,04932427	0,007585386
8	59,60776	0,1283813	99,98797	0,003056061	0,06527283	0,009084874
9	63,70102	0,1534899	99,9616	0,01249583	0,05872155	0,01077616
10	64,98484	0,166543	99,91533	0,0449681	0,3154187	0,2451253
11	63,6332	0,1507231	99,99701	0,001419155	0,009739701	0,002924656
12	64,57746	0,1594829	99,99566	0,001886285	0,01322345	0,00359823
13	63,91556	0,1595546	99,88276	0,01979398	0,3523923	0,03565591
14	65,1497	0,1647892	99,82857	0,02471236	0,4973413	0,07871535
15	63,73946	0,1531429	99,98469	0,004528561	0,08363652	0,01444899
16	64,63618	0,1577029	99,96787	0,00928503	0,07953763	0,01157153

Tabela A.2: Influência (2^k fatorial completo)

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
A	95,9807%	1,6735	5,0983%	2,3534	2,8994%	2,7533
B	0,0443%	0,0360	17,3113%	4,3366	16,4895%	6,5659
C	0,2345%	0,0827	56,2530%	7,8173	52,0685%	11,6676
D	3,5554%	0,3221	6,1262%	2,5798	10,5943%	5,2629
AB	0,0035%	0,0101	1,0818%	1,0841	0,9355%	1,5639
AC	0,0086%	0,0159	2,3249%	1,5892	1,5803%	2,0326
AD	0,0752%	0,0468	0,2747%	0,5463	0,0004%	0,0315
BC	0,0063%	0,0136	7,4210%	2,8393	4,4889%	3,4258
BD	0,0021%	0,0078	0,1174%	0,3571	0,3130%	0,9046
CD	0,0869%	0,0504	3,6842%	2,0006	9,6184%	5,0147
ABC	0,0007%	0,0046	0,2012%	0,4675	0,3840%	1,0019
ABD	0,0000%	0,0004	0,0619%	0,2593	0,1721%	0,6708
ACD	0,0016%	0,0069	0,0259%	0,1679	0,0335%	0,2960
BCD	0,0000%	0,0004	0,0090%	0,0988	0,3593%	0,9692
ABCD	0,0000%	0,0003	0,0092%	0,1001	0,0630%	0,4060

Tabela A.3: Resultados

Exp.	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
1	87,5447	0,4492399	99,46716	0,1488259	0,2424742	0,04029884
2	90,38229	0,465674	99,3035	0,1832241	0,8508034	0,3695742
3	86,75257	0,3617798	99,94912	0,02410954	0,0435447	0,01315031
4	89,04584	0,3823239	99,94675	0,02717927	0,05001928	0,01696544
5	89,99395	0,6123576	98,59177	0,2564057	0,9788034	0,09156718
6	92,79525	0,6458449	98,25801	0,2871284	1,992646	0,2891385
7	87,41531	0,400255	99,87963	0,03460451	0,1986788	0,03231535
8	89,58184	0,4096841	99,87233	0,03442849	0,2214884	0,03435592
9	105,4042	1,037764	97,08359	0,5672494	0,6981954	0,09566362
10	109,7233	1,24252	96,1859	0,7373968	2,288987	0,5351849
11	101,5863	0,5983136	99,85053	0,06681681	0,08070722	0,01964203
12	104,6405	0,6190227	99,83878	0,07353808	0,1092149	0,03284085
13	129,7403	3,187366	86,50338	1,646047	3,849189	0,3655993
14	133,1924	2,999129	83,57908	1,982358	9,308923	0,963312
15	103,6	0,7586861	99,58567	0,1119799	0,3870152	0,05566905
16	106,4483	0,7633716	99,5603	0,1146129	0,4519874	0,06443636

Tabela A.4: Influência (2^k fatorial completo)

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
A	64,9418%	2,1666	18,5879%	1,8688	11,8324%	1,4193
B	6,6097%	0,6912	11,3018%	1,4572	12,6522%	1,4677
C	9,6505%	0,8352	26,5156%	2,2321	25,9890%	2,1035
D	1,1224%	0,2848	0,3238%	0,2467	5,7695%	0,9911
AB	4,1237%	0,5460	7,9728%	1,2240	5,5355%	0,9708
AC	5,7626%	0,6454	16,8065%	1,7770	9,9757%	1,3032
AD	0,0254%	0,0428	0,1908%	0,1894	2,2499%	0,6189
BC	4,5091%	0,5709	10,1294%	1,3796	9,1451%	1,2478
BD	0,0030%	0,0148	0,0834%	0,1251	1,3965%	0,4876
CD	0,0184%	0,0365	0,3101%	0,2414	5,4519%	0,9634
ABC	3,2289%	0,4831	7,3957%	1,1788	4,7377%	0,8981
ABD	0,0016%	0,0109	0,0591%	0,1054	0,9050%	0,3925
ACD	0,0009%	0,0083	0,1846%	0,1863	2,1460%	0,6044
BCD	0,0006%	0,0068	0,0806%	0,1231	1,3292%	0,4757
ABCD	0,0011%	0,0090	0,0580%	0,1044	0,8842%	0,3880

Tabela A.5: Influência do número de contratos com variação da taxa de utilização

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
A1	0,13105	0,08638	-0,09111	0,08609	0,53993	0,08800
A2	0,22991	0,10900	-0,14205	0,10800	0,71916	0,11096
A3	0,47277	0,14623	-0,24476	0,14331	0,94921	0,14984
A4	1,21816	0,24412	-0,54246	0,22121	1,22505	0,24441
A5	7,94332	0,97243	-3,13690	0,71061	1,79342	0,93330

Tabela A.6: Influência da média dos contratos com variação da taxa de utilização

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
A1	-0,05090	0,00739	0,02449	0,00511	-0,01405	0,01121
A2	-0,08545	0,01341	0,04302	0,01025	-0,02950	0,01969
A3	-0,16448	0,02794	0,08834	0,02258	-0,04451	0,03620
A4	-0,46022	0,08605	0,26619	0,06493	-0,08159	0,09566
A5	-3,79541	0,53284	1,76741	0,31340	-0,55245	0,48349

Tabela A.7: Influência da dispersão dos contratos com variação da taxa de utilização

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
A1	0,09709	0,01087	-0,04678	0,00632	0,13971	0,09129
A2	0,12786	0,01504	-0,06281	0,00946	0,18282	0,10491
A3	0,18551	0,02197	-0,09171	0,01454	0,26346	0,12229
A4	0,31493	0,03898	-0,16495	0,02501	0,41512	0,15143
A5	1,31666	0,18477	-0,62458	0,09538	0,97955	0,25745

Tabela A.8: Influência da taxa de utilização com variação do número de contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
B1	14,35271	0,17572	-0,27186	0,22616	1,37963	0,24482
B2	14,81134	0,20562	-0,61335	0,25409	1,39617	0,28685
B3	15,22387	0,23319	-0,93647	0,27989	1,34365	0,32233
B4	15,60410	0,25535	-1,23547	0,30147	1,35721	0,36009
B5	15,91989	0,26752	-1,47563	0,31407	1,39198	0,38176

Tabela A.9: Influência da média dos contratos com variação do número de contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
B1	-0,14894	0,04685	0,14206	0,04740	-0,04398	0,07233
B2	-0,34610	0,07615	0,33074	0,07690	-0,09542	0,11440
B3	-0,50478	0,09460	0,48681	0,09538	-0,11121	0,13972
B4	-0,63792	0,10579	0,61731	0,10665	-0,14014	0,15694
B5	-0,73761	0,11242	0,71554	0,11329	-0,17638	0,16790

Tabela A.10: Influência da dispersão dos contratos com variação do número de contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
B1	0,05874	0,01430	-0,06024	0,01464	0,15609	0,11111
B2	0,16442	0,02558	-0,16726	0,02625	0,41365	0,16432
B3	0,27134	0,03497	-0,27667	0,03582	0,48227	0,15151
B4	0,37938	0,04238	-0,38569	0,04359	0,47416	0,12844
B5	0,47188	0,04723	-0,47973	0,04855	0,51647	0,12474

Tabela A.11: Influência da taxa de utilização com variação do número de contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
C1	16,00350	0,35372	4,26483	0,61426	1,73043	0,56100
C2	14,83295	0,20616	1,43052	0,27655	1,31573	0,28275
C3	14,56211	0,18719	0,93440	0,24768	1,22174	0,25508
C4	14,42722	0,17815	0,70150	0,23773	1,16276	0,24051
C5	14,34943	0,17299	0,55712	0,23100	1,12066	0,23240

Tabela A.12: Influência do número de contratos com variação do número de contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
C1	-1,15671	0,19602	-2,24798	0,31892	-0,48088	0,31892
C2	-0,29915	0,06790	-0,49299	0,08590	-0,07545	0,08590
C3	-0,12887	0,03470	-0,22177	0,04223	-0,02926	0,04223
C4	-0,07047	0,02119	-0,12583	0,02642	-0,01275	0,02642
C5	-0,04437	0,01404	-0,07787	0,01798	-0,00626	0,02105

Tabela A.13: Influência da dispersão dos contratos com variação do número de contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
C1	0,47370	0,06459	0,95150	0,13017	1,24509	0,24999
C2	0,16635	0,02544	0,32014	0,04104	0,33132	0,13307
C3	0,09455	0,01701	0,19455	0,02753	0,09920	0,06706
C4	0,06253	0,01227	0,13417	0,02090	0,03486	0,03965
C5	0,04507	0,00972	0,09859	0,01722	0,00871	0,01383

Tabela A.14: Influência da taxa de utilização com variação da dispersão dos contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
D1	14,76057	0,21365	1,76234	0,29850	-0,70894	0,26047
D2	14,91374	0,21625	1,77181	0,29869	-0,74453	0,26372
D3	15,14403	0,21996	1,78524	0,29986	-0,81067	0,26748
D4	15,44592	0,22277	1,80587	0,30761	-0,91117	0,27043
D5	15,86161	0,22936	1,83220	0,31645	-1,07142	0,27837

Tabela A.15: Influência do número de contratos com variação da dispersão dos contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
D1	-0,40911	0,08289	-0,67293	0,11033	0,39282	0,08361
D2	-0,41749	0,08388	-0,68756	0,11082	0,40141	0,08461
D3	-0,43157	0,08900	-0,71999	0,11333	0,41653	0,08970
D4	-0,46805	0,09386	-0,78573	0,11947	0,46061	0,09435
D5	-0,56336	0,10169	-0,94201	0,13102	0,58126	0,10150

Tabela A.16: Influência do número de contratos com variação da dispersão dos contratos

Fator	**Tempo de Resposta**	Intervalo de confiança	**Satisfação dos Usuários**	Intervalo de confiança	**Variação da Satisfação**	Intervalo de confiança
D1	0,19473	0,02639	0,37312	0,04320	-0,19770	0,02712
D2	0,20734	0,02846	0,39187	0,04647	-0,21098	0,02921
D3	0,23046	0,03231	0,44288	0,05383	-0,23764	0,03295
D4	0,41260	0,13761	0,52550	0,14615	-0,48264	0,12245
D5	0,85350	0,25895	0,60124	0,33536	-1,14467	0,22344

Referências

BANKS, J.; CARSON, J. S.; NELSON, B. L. *Discrete-event system simulation*. 3 ed. Upper Saddle River, New Jersey 07458: Prentice-Hall, 2000.

BIRTWHISTLE, G.; DAHL, O.; MYHRHAUG, B.; NYGAARD, K. *Simula begin*. 2 ed. Chartwell-Bratt Ltd, 1979.

BLAKE, S.; BLACK, D.; CARLSON, M.; DAVIES, E.; WANG, Z.; WEISS, W. RFC 2475: An architecture for differentiated services. *Internet RFC, Internet Engineering Task Force - IETF*, 1998.

BRADEN, R.; CLARK, D.; SHENKER, S. RFC 1633: Integrated services in the internet architecture: an overview. *Internet RFC, Internet Engineering Task Force - IETF*, 1994.

CABRAL, M. I. C.; SOUTO, L. M. D. Especificação de componentes para a construção de simuladores de redes sem fio ad hoc padrão ieee 802.11. *III Workshop de Dissertações da COPIN - WDCopin*, p. 67–72, 2004.

CASAGRANDE, L. S. *Política de escalonamento de tempo real baseada em exigência para provisão de qos absoluto em serviços web*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2007.

CASAGRANDE, L. S.; MELLO, R. F.; BERTAGNA, R.; ANDRADE FILHO, J. A.; MONACO, F. J. Exigency-based real-time scheduling policy to provide absolute QoS for web services. *19th International Symposium on Computer Architecture and High Performance Computing - SBAC-PAD*, v. 0, p. 255–262, 2007a.

CASAGRANDE, L. S.; MONACO, F. J.; MELLO, R. F.; BERTAGNA, R.; FILHO, J. A. A. Exigency-based real-time scheduling policy to provide absolute qos for web services. In: *SBAC-PAD '07: Proceedings of the 19th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD'06)*, Gramado, RS, Brazil: IEEE Computer Society, 2007b.

CHEN, X.; HEIDEMANN, J. Preferential treatment for short flows to reduce web latency. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, v. 41, n. 6, p. 779–794, 2003.

CHENG, A. M. K. *Real-time systems: Scheduling, analysis, and verification*. 1 ed. Wiley, 2002.

CUBERT, R. M.; FISHWICK, P. *Sim++, version 1.0*. University of Florida, Gainesville, FL: Department of Computer and Information Science and Engineering, 1995.

ESTRELLA, J. C.; TEIXEIRA, M. M.; SANTANA, M. J. Negotiation mechanisms on application level: a new approach to improve quality of service in web servers. *The 4th IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems, and 2nd International Workshop on Collaborative Computing, Integration, and Assurance. SEUS/WCCIA*, v. 0, p. 255–260, 2006.

FARINES, J. M.; FRAGA, J. S.; OLIVEIRA, R. S. *Sistemas de tempo real*, v. 1. 1 ed. São Paulo, SP: Escola de Computação da Sociedade Brasileira de Computação, 201 p., 2000.

FISHWICK, P. A. Simpack: getting started with simulation programming in C and C++. In: *Proceedings of the 24th conference on Winter simulation - WSC*, New York, NY, USA: ACM, 1992, p. 154–162.

HLAVICKA, J.; RACEK, S. C-Sim - the C language enhancement for discrete-time simulations. *Proceedings of the International Conference on Dependable Systems and Networks - DSN*, v. 0, p. 539, 2002.

ISO/IEC *Information technology - quality of service: Framework*. Relatório Técnico, ISO/IEC International Organization for Standardization/International Electrotechnical Commission 13236, 1998.

ISSARIYAKUL, T.; HOSSAIN, E. *Introduction to network simulator ns2*. Springer Publishing Company, Incorporated, 2008.

JAIN, R. *The art of computer systems performance analysis : techniques for experimental design, measurement, simulation, and modeling*. New York, NY, USA, Wiley, 1991.

JAJSZCZYK, A. Quality of service challenges in ip networks. In: *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*, 2008, p. vi.

KAMIENSKI, C.; SADOK, D.; CAVALCANTI, D. A. T.; SOUZA, D. M. T.; DIAS, K. L. *Simulando a internet: Aplicações na pesquisa e no ensino.* Floarianópolis, SC: 21^o Jornada de Atualização em Informática - JAI, 97-138 p., 2002.

KUMAR, K. H.; MAJHI, S. Queuing theory based open loop control of web server. *In Proceedings of the 2004 American Control Conference*, v. 3, p. 2314–2315, 2004.

LAW, A. M.; KELTON, W. *Simulation modeling and analysis*. 4 ed. USA: McGraw-Hill Publishing Co., 2006.

LAZOWSKA, E. D.; ZAHORJAN, J.; GRAHAM, G. S.; SEVCIK, K. C. *Quantitative system performance: Computer system analysis using queueing network models*. 1 ed. Upper Saddle River, NJ, USA: Prentice-Hall Inc., 1984.

LIU, C. L.; LAYLAND, J. W. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Readings in hardware/software co-design*, p. 179–194, 2002.

LIU, J. W. S. *Real-time systems*. 1 ed. Upper Saddle River, New Jersey 07458: Prentice Hall, 2000.

MACDOUGALL, M. H. *Simulating computer systems: techniques and tools*. 2 ed. Cambridge, MA, USA: MIT Press Series in Computer Systems, 1989.

MACDOUGALL, M. H.; MCALPINE, J. S. Computer system simulation with ASPOL. In: *Proceedings of the 1st Symposium on Simulation of Computer Systems - ANSS*, Piscataway, NJ, USA: IEEE Press, 1973, p. 92–103.

MAGALHÃES, M. F.; CARDOZO, E. *Qualidade de serviço na internet*. Relatório Técnico, Faculdade de Engenharia Elétrica e de Computação, Departamento de Engenharia de Computação e Automação Industrial - UNICAMP, Campinas, SP, 1999.

MAMANI, E. L. C. *Um sistema servidor web distribuído com provisão de qos absoluta e relativa*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2010.

MARCHESE, M. *Qos over heterogeneous networks*. 1 ed. Wiley, 2007.

MESSIAS, V. R. *Servidor web distribuído com diferenciação de serviços - implementação e avaliação de um protótipo*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2007.

MONACO, F.; MAMANI, E.; NERY, M.; NOBILE, P. A novel qos modeling approach for soft real-time systems with performance guarantees. *High Performance Computing and Simulation Conference*, p. 89–95, 2009.

MONACO, F. J.; NOBILE, P. N. Feedback-based adaptive resource control in qos-aware soa systems with soft real-time requirements. *Quality of Service in Heterogeneous Networks*, v. 22, n. 1, p. 799–810, 2009.

NERY, M. *Políticas de escalonamento de tempo-real para garantias de qos na web baseada em parâmetros de média tempo de resposta e dispersão dos atrasos*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2009.

NISSANKE, N. *Realtime systems*. 1 ed. Prentice-Hall, 1997.

PEIXOTO, M. L. M. *Políticas de escalonamento de tempo-real para garantia de qos absoluta em array de servidores web heterogêneos*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2008.

SAITO, P. T. M. *Provisão integrada de qos relativa e absoluta em serviços computacionais interativos com requisitos de responsividade de tempo real*. Qualificação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2009.

SCHWETMAN, H. CSIM: A C-based, process-oriented simulation language. In: *Proceedings of the 18th conference on Winter simulation - WSC*, New York, NY, USA: ACM, 1986, p. 387–396.

SHA, L.; ABDELZAHER, T.; ARZÉN, K. E.; CERVIN, A.; BAKER, T.; BURNS, A.; BUTTAZZO, G.; CACCAMO, M.; LEHOCZKY, J.; MOK, A. K. Real time scheduling theory: A historical perspective. *Real-Time Systems*, v. 28, n. 2-3, p. 101–155, 2004.

SHANNON, R. E. Introduction to the art and science of simulation. In: *Proceedings of the 30th Conference on Winter simulation - WSC*, Los Alamitos, CA, USA: IEEE Computer Society Press, 1998, p. 7–14.

SHELDON, T. *Encyclopedia of networking and telecommunication*. 1 ed. McGraw-Hill, 2001.

SILVA, L.; PEREIRA, A.; JR MEIRA, W. Reactivity-based scheduling approaches for internet services. *4th Latin American Web Congress - LA-WEB*, p. 47–58, 2006.

TANENBAUM, A. S. *Computer networks*. 4 ed. Prentice Hall, 2003.

TAVARES, E.; SILVA, B.; MACIEL, P.; DALLEGRAVE, P. Software synthesis for hard real-time embedded systems with energy constraints. In: *Proceedings of the 2008 20th International Symposium on Computer Architecture and High Performance Computing - SBAC-PAD*, Washington, DC, USA: IEEE Computer Society, 2008, p. 115–122.

TEIXEIRA, M. M.; SANTANA, M. J.; SANTANA, R. H. C. Servidor web com diferenciação de serviços: Fornecendo qos para os serviços da internet. In: *XXIII Simpósio Brasileiro de Redes de Computadores (SBRC)*, Fortaleza, CE, 2005, p. 1–14.

TIJMS, H. *A first course in stochastic models*. 1 ed. Wiley, Chichester, 2003.

TOTT, R. F. *Extensões na política ebs - controle de admissão e redução da ordem de complexidade temporal*. Dissertação de mestrado, Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, Brasil, 2008.

TRALDI, O. A.; BARBATO, A. K.; SANTANA, R. H. C. Service differentiating algorithms for QoS-enabled web servers. In: *Proceedings of the 12th Brazilian Symposium on Multimedia and the Web - WebMedia*, New York, NY, USA: ACM, 2006, p. 263–272.

VASILIOU, N. Overview of internet QoS and web server QoS, reading course report. Department of Computer Science, The University of Western Ontario, London, Ontario, Canada., 2000. Disponível em: `www.cs.uwo.ca/Research/DiGS/Papers/nikoread.ps`

WEI, Y.; REN, F.; LIN, C.; VOIGT, T. Fuzzy control for guaranteeing absolute delays in web servers. In: *Proceedings of the 2nd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks - QSHINE*, Washington, DC, USA: IEEE Computer Society, 2005, p. 48–51.

ZHANG, Y.; KRECKER, D. K.; GILL, C.; LU, C.; THAKER, G. H. Practical schedulability analysis for generalized sporadic tasks in distributed real-time systems. In: *Proceedings of the 20th Euromicro Conference on Real-Time Systems - ECRTS*, IEEE Computer Society, 2008, p. 223–232.

ZHAO, W.; OLSHEFSKI, D.; SCHULZRINNE, H. *Internet quality of service: an overview*. Relatório Técnico, Technical Report CUCS-003-00, Columbia University, Computer Science Department, 2000.
Disponível em: `http://www.cs.columbia.edu/techreports/cucs-003-00.pdf`