SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: ____ / ____ / ____

Assinatura : _____

Reduzindo a Dispersão dos Atrasos em Sistemas de Tempo Real Soft com Restrições de Média de Tempo de Resposta

Michelle Nery

Orientador: *Prof. Dr. Francisco José Monaco*

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

USP - São Carlos
Setembro de 2009

**Reduzindo a Dispersão dos Atrasos em Sistemas de
Tempo Real Soft com Restrições de Média de Tempo
de Resposta**

Michelle Nery

*“Seja quem você for, seja
qualquer posição que você tenha
na vida - nível altíssimo ou mais
baixo social - tenha sempre como
meta, muita força, muita
determinação e sempre faça
tudo com muito amor e com
muita fé em Deus, que um dia
você chega lá, de alguma
maneira você chega lá.”*

Ayrton Senna

A meus pais

Agradecimentos

Agradeço, primeiramente, a Deus.

A Universidade de São Paulo, USP - Campus de São Carlos, pela oportunidade de realizar o curso de Pós-Graduação.

Aos meus pais, Terezinha e Pedro, pelo apoio incondicional em todas as horas.

Ao meu irmão Wesley e minha cunhada Alcione, que preocupam-se sempre com meu bem estar.

Lulu, pelo carinho e afeto.

Ao meu orientador e amigo, Monaco, pela dedicação ao me orientar, além do imenso apoio (em diversos aspectos) e da paciência com minhas falhas e dúvidas.

Aos meus amigos e colegas da USP, em especial ao Lucas Casagrande por proporcionar a base deste trabalho. Ao Augusto, pelas discussões e sugestões do trabalho, mostrando-se sempre disposto a ajudar. Ao Ricardo, Marcelo, Lourenço, Matheus, Tott, Maycon, Priscila, Edwin, Pedro, enfim, a todos que contribuíram direta ou indiretamente para realização deste trabalho.

Aos professores Marcos, Regina, Sarita, Kalinka, Paulo, Edson e Eduardo Marques pela promoção do eventos do grupo de Sistemas Distribuídos e Programação Concorrente (SDPC).

Aos funcionários do ICMC-USP, em especial aos funcionários da secretária da Pós-Graduação, pelo convívio amigável e descontraído, e pela disposição em sempre bem atender.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e a FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo), pelo apoio financeiro.

Resumo

A Qualidade de Serviço oferece aos Sistemas de Tempo Real garantias das restrições temporais de aplicações tais como comércio eletrônico, vídeo conferência, telemedicina, entre outras que necessitam de confiabilidade e desempenho para efetivação dos seus serviços. Contribuições nessas áreas têm focado a provisão de qualidade de serviço em termos absolutos, descritos num limite máximo de média de tempo de atendimento, destinado às aplicações *Web*. Todavia, o controle de qualidade baseado em tempo médio de resposta durante todo o período de serviço prestado, não restringe os tempos de resposta quanto a sua dispersão. Um valor médio razoável pode ser resultado da combinação de tempos de resposta elevados e pequenos. O objetivo deste trabalho é reduzir a dispersão dos atrasos em sistemas escalonados com a política *EBS* (*Exigency Based Scheduling*), definidos pela diferença positiva entre o tempo de resposta da requisição do usuário e o valor médio contratado, em um modelo de contrato determinado por janelas deslizantes de operações. A *EBS* é uma política de escalonamento que define um modelo de contratos de tempo médio de resposta avaliados durante todo o tempo de serviço. Avalia-se a dispersão dos atrasos, pois, é preferível que haja a possibilidade dos tempos de resposta ficarem substancialmente abaixo da média contratada. Os resultados são obtidos utilizando métodos de simulação dos algoritmos desenvolvidos durante o projeto.

Abstract

Quality of Service offers Real Time Systems guarantees the time constraints of applications such as electronic commerce, video conferencing, telemedicine, among others that need reliable and effective performance for their services. Contributions in these areas have made the provision of service quality in absolute terms, out of a maximum average time of service for Web applications. However, quality control based on average response time for the entire period of service, does not restrict the response time to their dispersion. A reasonable average value may be the result of the combination of response times of high and low. Objective of this research is reduce the delays dispersion in systems scheduled with policy EBS (Exigency Based Scheduling), defined by the positive difference between the response time of user request and the average contract value, in a model contract by a sliding operation windows. The EBS is a scheduling policy that defines a contracts model for an average response time evaluated throughout the time of service. We analyze the delays dispersion because it is preferable that there is the possibility of response time becomes substantially below the average contracted. Results are obtained using simulation methods of the algorithms developed during the project.

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Motivação e Objetivo	2
1.3	Estrutura	3
2	Fundamentação Teórica	5
2.1	Qualidade de Serviço	5
2.1.1	Conceitos Básicos de QoS	6
2.1.2	Arquiteturas de QoS	7
2.1.3	Qualidade de Serviço em Nível de Aplicação	8
2.2	Tempo-Real	9
2.2.1	Conceitos Básicos de Tempo-Real	10
2.2.2	Sistemas Hard-TR, Soft-TR e Firm-TR	11
2.2.3	Escalonamento de Sistemas de Tempo Real	11
2.3	Simulação de Sistemas	14
2.3.1	Simulação de Filas	15
2.3.2	Probabilidade e Estatística para Simulação	17
2.4	Considerações Finais	19
3	<i>EBS: Exigency Based Scheduling</i>	21
3.1	Política de Escalonamento <i>EBS</i>	21
3.2	Considerações Finais	23
4	Avaliação da <i>EBS</i> em um <i>SLA</i> com Janelas Deslizantes	25
4.1	<i>SLA</i> com Janelas Deslizantes	25
4.2	Materiais e Métodos	27
4.3	Satisfação Contratual	28
4.4	Planejamento de Experimentos	28
4.5	Análise dos Resultados	29

4.5.1	Satisfação Contratual da Classe A	30
4.5.2	Satisfação Contratual da Classe B	33
4.5.3	Tempos de Resposta abaixo da Média Contratada	35
4.5.4	Resumo	38
4.6	Considerações Finais	39
5	Reduzindo o Desvio Padrão dos Atrasos	41
5.1	Política aEBS	41
5.2	Políticas σ EBS	43
5.3	Medidas para Avaliação	44
5.4	Planejamento de Experimentos	44
5.5	Análise dos Resultados	45
5.5.1	Análise quanto ao Tamanho da Janela	45
5.5.2	Análise quanto a Variação Contratual	52
5.5.3	Análise quanto as Políticas de Escalonamento	57
5.6	Considerações Finais	69
6	Conclusão	71
6.1	Conclusões Gerais	71
6.1.1	<i>SLA</i> com Janelas Deslizante	71
6.1.2	Reduzindo o Desvio Padrão dos Atrasos	72
6.2	Contribuições	74
6.3	Trabalhos Futuros	75
	Referências	77
A	Apêndice	81
A.1	Fórmula da Variância	81

Lista de Figuras

2.1	<i>Elementos de uma Fila (Prado, 2004).</i>	15
2.2	<i>Modelos de Redes de Fila.</i>	17
4.1	<i>Janelas Deslizantes de Operações.</i>	26
4.2	<i>Satisfação dos usuários da classe A - Cenário: 10% requisições classe A.</i>	31
4.3	<i>Degradação do Sistema.</i>	31
4.4	<i>Satisfação dos usuários da classe A - Cenário: 50% requisições classe A.</i>	32
4.5	<i>Satisfação dos usuários da classe A - Cenário: 90% requisições classe A.</i>	33
4.6	<i>Comportamento do escalonador.</i>	33
4.7	<i>Satisfação dos usuários da classe B - Cenário: 10% requisições classe B.</i>	34
4.8	<i>Satisfação dos usuários da classe B - Cenário: 50% requisições classe B.</i>	35
4.9	<i>Satisfação dos usuários da classe B - Cenário: 90% requisições classe B.</i>	35
4.10	<i>Tempos de Resposta abaixo da média contratada - Cenário 10% requisições classe A.</i>	37
4.11	<i>Tempos de Resposta abaixo da média contratada - Cenário 50% requisições classe A.</i>	37
4.12	<i>Tempos de Resposta abaixo da média contratada - Cenário 90% requisições classe A.</i>	37
4.13	<i>Tempos de Resposta abaixo da média contratada - Cenário 10% requisições classe B.</i>	39
4.14	<i>Tempos de Resposta abaixo da média contratada - Cenário 50% requisições classe B.</i>	39
4.15	<i>Tempos de Resposta abaixo da média contratada - Cenário 90% requisições classe B.</i>	39
5.1	<i>Exemplo de janela com baixos tempos de resposta.</i>	42
5.2	<i>Desvio Padrão dos Atrasos - Cenário: 25% requisição classe A.</i>	46
5.3	<i>Desvio Padrão dos Atrasos - Cenário: 75% requisição classe B.</i>	47
5.4	<i>Média dos Atrasos - Cenário: 25% requisição classe A.</i>	47

5.5	<i>Média dos Atrasos - Cenário: 75% requisição classe B.</i>	48
5.6	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe A.</i>	49
5.7	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe B.</i>	49
5.8	<i>Desvio Padrão dos Atrasos - Cenário: 75% requisição classe A.</i>	50
5.9	<i>Desvio Padrão dos Atrasos - Cenário: 25% requisição classe B.</i>	50
5.10	<i>Média dos Atrasos - Cenário: 75% requisição classe A.</i>	51
5.11	<i>Média dos Atrasos - Cenário: 25% requisição classe B.</i>	51
5.12	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe A.</i>	52
5.13	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe B.</i>	52
5.14	<i>Desvio Padrão dos Atrasos - Cenário: 25% requisição classe A.</i>	53
5.15	<i>Desvio Padrão dos Atrasos - Cenário: 75% requisição classe B.</i>	54
5.16	<i>Média dos Atrasos - Cenário: 25% requisição classe A.</i>	54
5.17	<i>Média dos Atrasos - Cenário: 75% requisição classe B.</i>	55
5.18	<i>Desvio Padrão dos Atrasos - Cenário: 75% requisição classe A.</i>	55
5.19	<i>Desvio Padrão dos Atrasos - Cenário: 25% requisição classe B.</i>	56
5.20	<i>Média dos Atrasos - Cenário: 75% requisição classe A.</i>	56
5.21	<i>Média dos Atrasos - Cenário: 25% requisição classe B.</i>	57
5.22	<i>Satisfação Contratual - Cenário: 25% requisição classe A.</i>	58
5.23	<i>Satisfação Contratual - Cenário: 75% requisição classe B.</i>	58
5.24	<i>Satisfação Contratual - Cenário: 75% requisição classe A.</i>	59
5.25	<i>Satisfação Contratual - Cenário: 25% requisição classe B.</i>	59
5.26	<i>Média dos Atrasos - Cenário: 25% requisição classe A.</i>	60
5.27	<i>Média dos Atrasos - Cenário: 75% requisição classe B.</i>	60
5.28	<i>Média dos Atrasos - Cenário: 75% requisição classe A.</i>	61
5.29	<i>Média dos Atrasos - Cenário: 25% requisição classe B.</i>	61
5.30	<i>Desvio Padrão dos Atrasos - Cenário: 25% requisição classe A.</i>	62
5.31	<i>Desvio Padrão dos Atrasos - Cenário: 75% requisição classe B.</i>	63
5.32	<i>Desvio Padrão dos Atrasos - Cenário: 75% requisição classe A.</i>	63
5.33	<i>Desvio Padrão dos Atrasos - Cenário: 25% requisição classe B.</i>	64
5.34	<i>Comportamento do Desvio Padrão durante a Simulação - Cenário: 25% A e 75% B, Variação Contratual 10%.</i>	65
5.35	<i>Comportamento do Desvio Padrão durante a Simulação - Cenário: 75% A e 25% B, Variação Contratual 10%.</i>	65
5.36	<i>Comportamento do Desvio Padrão durante a Simulação - Cenário: 25% A e 75% B, Variação Contratual 20%.</i>	66

5.37	<i>Comportamento do Desvio Padrão durante a Simulação - Cenário: 75% A e 25% B, Variação Contratual 20%.</i>	66
5.38	<i>Comportamento do Desvio Padrão durante a Simulação - Cenário: 25% A e 75% B, Variação Contratual 30%.</i>	67
5.39	<i>Comportamento do Desvio Padrão durante a Simulação - Cenário: 75% A e 25% B, Variação Contratual 30%.</i>	67
5.40	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe A.</i>	68
5.41	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe B.</i>	69
5.42	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe A.</i>	69
5.43	<i>Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe B.</i>	70

Lista de Tabelas

4.1	Δ referente a cada variação contratual e proporção de requisições na classe	
	A.	38
4.2	Δ referente a cada variação contratual e proporção de requisições na classe	
	B.	40

Lista de Siglas

aEBS	<i>Aged Exigency Based Scheduling</i>
DiffServ	<i>Service Differentiation</i>
EBS	<i>Exigency Based Scheduling</i>
EDF	<i>Earliest Deadline First</i>
FIFO	<i>First In First Out</i>
FTP	<i>File Transfer Protocol</i>
IETF	<i>Internet Engineering Task Force</i>
IntServ	<i>Integrated Services</i>
IP	<i>Internet Protocol</i>
ISO	<i>International Organization for Standardization</i>
LCFS	<i>Last Come, First Served</i>
QoS	<i>Qualidade de Serviço</i>
RSVP	<i>Resource Reservation Protocol</i>
RSVAdap	<i>Reserva Adaptativa de Recursos</i>
WFQ	<i>Weighted Fair Queuing</i>
SJF	<i>Shortest Job First</i>
SLA	<i>Service Level Agreement</i>
SWDS	<i>Servidor Web com Diferenciação de Serviços</i>

Lista de Símbolos

D_j	<i>Deadline da requisição j</i>
P_j	<i>Prioridade da requisição j</i>
R_u	<i>Número de requisições do usuário u</i>
R_{ρ_u}	<i>Número de requisições atrasadas do usuário u</i>
$R_{tr\downarrow}$	<i>Requisições com o tempo de resposta abaixo da média contratada</i>
$time()$	<i>Tempo atual</i>
$timeStamp_j$	<i>Tempo de chegada da requisição j</i>
T_{p_j}	<i>Tempo de processamento da requisição j</i>
T_{w_j}	<i>Tempo de espera da requisição j</i>
wS_u	<i>Tamanho da janela do usuário u</i>
$W[x]_{i_u}$	<i>Tempo de resposta da requisição mais antiga dentro da janela do usuário u</i>
Ω_u	<i>Tempo médio de resposta do usuário u</i>
Ω'_u	<i>Antigo tempo médio de resposta do usuário u</i>
Ω_{c_u}	<i>Tempo médio de resposta contratado</i>
$\Omega_{c_{uA}}$	<i>Tempo médio de resposta contratado pelo usuário da classe A</i>
$\Omega_{c_{uB}}$	<i>Tempo médio de resposta contratado pelo usuário da classe B</i>
Ω_{FIFO}	<i>Tempo médio de resposta da política FIFO</i>
ρ_j	<i>Atraso da requisição j</i>
σ_u	<i>Desvio Padrão do usuário u</i>
σ_u^2	<i>Variância do usuário u</i>
$\sigma_u^{2'}$	<i>Antiga Variância do usuário u</i>
$\bar{\rho}_u$	<i>Médio dos atrasos do usuário u</i>
$\bar{\rho}_u'$	<i>Antiga Média de atrasos do usuário u</i>

Introdução

1.1 Contextualização

Sistemas computacionais interativos são facilmente encontrados no cotidiano das pessoas, em forma de tecnologias como comércio eletrônico, sistemas de conferência multimídia, telemedicina, entre outras. Muitas dessas aplicações interativas possuem **restrições temporais** que necessitam de confiabilidade e desempenho para a efetivação dos serviços. O estudo e desenvolvimento de técnicas específicas para a análise e síntese de mecanismos que devem operar, sob estas restrições, correspondem ao domínio tecnológico de sistemas de tempo real.

Segundo as restrições temporais, os sistemas de tempo real podem ser classificados como *hard*, *soft* e *firm*. Para sistemas de tempo real *hard*, as restrições temporais devem rigorosamente ser respeitadas. Para sistemas de tempo real *soft*, são admissíveis atrasos, ocasionalmente. Por último, sistemas de tempo real *firm* também admitem atrasos, porém, as requisições atrasadas devem ser rejeitados pelo sistema, diferente dos sistemas *soft*, cuja decisão de descarte é relativa à finalidade da aplicação (Liu *et al.*, 2006).

Cabe ao mecanismo de Qualidade de Serviço (*QoS*), empregado também no nível de aplicação da arquitetura de redes, supervisionar as restrições temporais dos sistemas de tempo real. A especificação de *QoS* pode se dar de maneira relativa ou absoluta. Quando feita em termos relativos, pretende-se garantir que uma classe de serviço com maior prioridade tenha um tratamento melhor ou igual que o de qualquer classe de serviço inferior. Já a *QoS* absoluta, por sua vez, estabelece contratos a serem cumpridos, como taxa mínima de serviço ou um tempo médio de atendimento máximo para as requisições.

Contribuições das pesquisas empreendidas nessas áreas de Tempo-Real e *QoS* no grupo

de SDPC¹, têm como objetivo principal garantir as restrições temporais em aplicações *Web*, em que a perda ocasional do prazo não é fatal para o sistema, ou seja, uma restrição de tempo real *soft*, e determinam garantias absolutas de parâmetros de *QoS* (Casagrande *et al.*, 2007), (Peixoto *et al.*, 2007), (Peixoto & Monaco, 2008), (Monaco *et al.*, 2009).

Previamente, nesses trabalhos, os parâmetros contratuais tem sido formulados por tempos médios de resposta observados durante todo o tempo de serviço prestado aos usuários. Todavia, em determinadas situações, a descrição do comportamento do sistema utilizando apenas a média não é suficiente, pois, essa não contém informações sobre o espalhamento dos dados, ou seja, o quão distante as amostras estão do valor médio. Medir o tempo médio de resposta em uma conexão, por exemplo, pode não ser suficiente para avaliar a qualidade do serviço, pois, um valor médio razoável pode ser resultado da combinação de tempos de resposta bastante elevados e tempos de resposta pequenos. Essa variação, muitas vezes, pode ser mais prejudicial para a aplicação do que um atraso médio elevado.

Constituído de um problema NP-completo², esforços para minimização das variações dos tempos de resposta, tem se dado ao escalonamento de tarefas *offline* (Eilon & Chowdhry, 1977), (Ye *et al.*, 2007), (Li *et al.*, 2007b), (Li *et al.*, 2007a) ou utilizando algoritmos genéticos (Al-Anzi & Allahverdi, 2005).

1.2 Motivação e Objetivo

Diversos trabalhos realizados pelo grupo de SDPC tem aprimorado a infraestrutura da camada de aplicação, com garantias de qualidade de serviço buscando atender as exigências das novas aplicações *Web* (Teixeira *et al.*, 2004), (Teixeira *et al.*, 2005), (Estrella *et al.*, 2006), (Traldi *et al.*, 2006), (Messias *et al.*, 2007), (Casagrande *et al.*, 2007), (Peixoto & Monaco, 2008). Este trabalho é motivado por ideias complementares, em especial, ao trabalho de Casagrande *et al.* (2007) que introduz uma política de escalonamento de tempo-real não-determinístico (*soft*) para provisão de *QoS* absoluto em nível de aplicações *Web*, nomeada *EBS* (*Exigency Based Scheduling*).

No modelo proposto, o parâmetro de *QoS* contratado por cada cliente é um limite superior de média de tempo de resposta que deve ser observado pelo servidor durante todo o tempo de serviço prestado ao usuário. Para isso, o sistema mantém atualizada uma tabela de contratos em que cada entrada correspondente a um cliente, relaciona o tempo médio de resposta contratado pelo usuário (Ω_{Cu}), fixo, e tempo médio de resposta efetivo do usuário (Ω_u), que reflete a média instantânea de tempo de resposta de fato praticada.

¹Sistemas Distribuídos e Programação Concorrente.

²ou *NP-hard*, consiste em problemas que não podem ser resolvidos em tempos de problemas polinomiais (Cormen *et al.*, 2001).

A política elaborada é baseada na técnica de *feedback scheduling* (escalonamento retroalimentado) (Sha *et al.*, 2004) e tem como princípio comparar, a cada ciclo de escalonamento, a diferença $\Omega_{Cu} - \Omega_u$. A prioridade da requisição é dada pelo produto do custo computacional (ou tempo de processamento) com o *deadline* (tempo em que uma dada requisição pode esperar sem que o contrato seja violado³). É associada a essa abordagem uma heurística baseada nos algoritmos clássicos *EDF*⁴ (Liu, 2000) (política de escalonamento de tempo real) e *SJF*⁵ (Sprunt *et al.*, 1989) (escalonamento sem restrição temporal, que minimiza a média de tempo de atendimento) para aprimorar a confiabilidade e o desempenho do sistema.

Logo, o objetivo deste trabalho é avaliar a dispersão dos atrasos das requisições escalonadas pela *EBS* e apresentar novas abordagens para reduzir essa variação com janelas deslizantes de operações. Atraso é o quanto o tempo de resposta da requisição ultrapassou o tempo médio contratado pelo usuário. E, as janelas de operações deslizantes referenciam a média efetiva dos usuários com os tempos de resposta apenas das requisições mais recentes deles.

1.3 Estrutura

O restante deste documento está organizado da seguinte forma:

- No **Capítulo 2** apresenta-se a fundamentação teórica que engloba este trabalho. Sucintamente serão discutidos os principais conceitos utilizados, dentre eles, o de Qualidade de Serviço, na Seção 2.2, Tempo Real, na Seção 2.3, Simulação de Sistemas.
- No **Capítulo 3** será apresentado com mais detalhes a política de escalonamento *EBS* (*Exigency Based Scheduling*).
- No **Capítulo 4** serão apresentados um novo modelo de contrato determinado em janelas deslizantes na Seção 4.1, os materiais e métodos utilizados para avaliar o desempenho da política sobre janelas na Seção 4.2, a medida que valida a política na Seção 4.3, os experimentos na Seção 4.4 e as análises dos resultados desse modelo sobre a política *EBS* na Seção 4.5.
- No **Capítulo 5** serão apresentadas as duas novas políticas de escalonamento que visam reduzir a dispersão dos atrasos nas Seções 5.1 e 5.2. Também são apresentadas as medidas para validar os resultados na Seção 5.3 e o planejamento de experimentos na Seção 5.4. Os resultados obtidos em comparação com a *EBS* são apresentados na Seção 5.5.

³Quando o tempo avaliado ultrapassa o limite contratado.

⁴Earliest Deadline First.

⁵Shortest Job First.

-
- No **Capítulo 6** apresentam-se os principais resultados obtidos na Seção 6.1, as contribuições do trabalho na Seção 6.2, e sugere-se melhorias como trabalhos futuros na Seção 6.3.
 - No **Capítulo 7** são apresentadas as Referências.

Fundamentação Teórica

Neste capítulo serão discutidos novos desafios e demandas impostos à *Internet*, apresentando um embasamento teórico do modelo proposto para lidar com eles, o de Qualidade de Serviço. Abordagens tem sido desenvolvidas na área de Tempo Real com intuito de prover requisitos de confiabilidade e segurança, sendo importante uma breve revisão de seus conceitos.

Logo, o capítulo inicia com uma introdução sobre os conceitos básicos relacionados à Qualidade de Serviço (Seção 2.1), direcionando para as principais arquiteturas de *QoS* na Seção 2.1.2, e mostra as tendências da *QoS* na camada de aplicação, Seção 2.1.3.

Em seguida, aborda-se resumidamente a aplicabilidade da área de Tempo-Real na Seção 2.2. Alguns conceitos básicos são apresentados na Seção 2.2.1, pois, ao relacionar parâmetros de serviço em termos temporais utiliza-se conceitos dessa área. Em seguida, as classificações segundo as restrições temporais são discutidos na Seção 2.2.2, e na Seção 2.2.3 são mostrados as classificações dos algoritmos de escalonamento de tempo real, e citado as principais políticas de escalonamento.

Como ferramentas úteis para o desenvolvimento e análise das novas políticas e algoritmos de escalonamento, a Seção 2.3 apresenta resumidamente alguns conceitos sobre modelagem e simulação de sistemas. A técnica de simulação é apresentada na Seção 2.3.1, em seguida, é discutido um pouco sobre probabilidade e estatística na Seção 2.3.2 devido a importância de suas ferramentas para a análise de modelos de sistemas.

2.1 Qualidade de Serviço

O modelo original de serviços utilizado na *Internet*, em geral, trata as requisições de modo equivalente, seja no processamento ou na transmissão ao longo da rede. Entretanto,

o crescimento diversificado de aplicações tem requisitado mudanças no paradigma de serviço, tornando a qualidade de serviço cada vez mais necessária e crítica.

As redes baseadas em *IP* (*Internet Protocol*), como a *Internet*, proveem o modelo de melhor esforço (*best effort*) na entrega dos dados. Com esse modelo, a largura de banda é compartilhada com todos os fluxos de dados dos usuários. As transmissões são concorrentes entre si. Em caso de congestionamento, os pacotes de dados podem sofrer atrasos ou até serem descartados sem distinção, o que não garante que o serviço seja bem sucedido ou que tenha um bom desempenho (Vasiliou, 2000).

Aumentar a largura de banda para evitar atrasos e descartes, pode ser uma forma de tratar o problema. Porém, experiências mostram que não importa a capacidade de banda disponibilizada, a tendência é que em pouco tempo, a capacidade seja consumida por tráfego em rajadas por exemplo. Além disso, em algumas aplicações, o ponto crítico está na latência de transmissão, como voz sobre *IP*, e não somente na capacidade de transmissão.

A solução, então, é que um certo nível de inteligência deve ser introduzido na *Internet*, de modo que diferencie tráfegos com requisitos estritos de tempo daqueles que permitem atrasos ou perdas. Esse gerenciamento de tráfego é feito por meio de mecanismos de Qualidade de Serviço (*QoS*), que buscam prover aos usuários da rede níveis de confiabilidade na satisfação dos seus requisitos.

2.1.1 Conceitos Básicos de QoS

Para a ISO¹, *QoS* é definida como o efeito coletivo do desempenho de um serviço, o qual determina o grau de satisfação de um usuário (ISO/IEC, 1998). Essa definição é genérica e deve ser especificada para o problema que se deseja tratar.

As principais formulações de *QoS* são: **garantia de desempenho** e **diferenciação de serviço**. O primeiro está diretamente relacionado com largura de banda, atraso e perda de pacotes, enquanto que, a diferenciação de serviço refere-se aos diferentes níveis de prioridade a diversas aplicações, cada qual com requisitos distintos.

Existem aplicações mais sensíveis a perda de dados, como transferência de arquivo via *FTP*². Por outro lado, aplicações de telefonia toleram perdas ocasionais de pacotes, porém, exigem que seus pacotes cheguem em tempos pré-determinados. Aplicações como multimídia, além de exigirem que seus pacotes cheguem no tempo correto, têm ainda sua utilidade relacionada a previsibilidade do *jitter*³. Além dos fatores técnicos, a paciência humana exige baixos tempos de transferência.

A *QoS* tipicamente é especificada em sistemas finais por meio de valores **absolutos**, como: "nenhum pacote pode atrasar mais que 200 ms" ou "a perda de pacotes não excederá

¹International Organization for Standardization

²File Transfer Protocol.

³Variação dos atrasos.

a 1%", e em termos **relativos**, que define classes de usuários com prioridade mais altas e classes com prioridades mais baixas. As classes com prioridade mais alta terão um serviço melhor (ou no mínimo igual) às classes com prioridade mais baixas.

2.1.2 Arquiteturas de QoS

Dentre os vários modelos existentes para suporte de *QoS*, dois deles (propostos pela *IETF*⁴) se destacam: Serviços Integrados (*IntServ*) (Braden *et al.*, 1994) e os Serviços Diferenciados (*DiffServ*⁵) (Blake *et al.*, 1998). *IntServ*⁶ é um modelo baseado em reserva de recursos, enquanto que, *DiffServ* é um modelo onde os pacotes são marcados 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

O modelo de Serviços Integrados (Braden *et al.*, 1994) apresenta duas características chave. A primeira característica é a de recurso reservado. O roteador precisa saber que quantidade de seus recursos já está reservada para sessões em andamento e o quanto ele pode reservar para sessões futuras. Nesse caso, recursos são: disponibilidade de *buffers* e taxa de transmissão. Outra característica é o estabelecimento de chamada ou aceitação de chamada. Uma sessão que exige garantias de *QoS*, primeiramente deve estar habilitada a reservar recursos suficientes em cada roteador que fará parte do caminho dos dados, para garantir que suas exigências de *QoS* possam ser cumpridas. Portanto, participam dessa etapa todos os roteadores envolvidos no caminho fim a fim. Cada roteador deve estabelecer os recursos locais exigidos pela sessão, considerando a quantidade de recursos já comprometidos com outras sessões em andamento e determinar se os recursos existentes são suficientes para atender a chamada (Kurose & Ross, 2006). A arquitetura *IntServ* define duas classes de serviço:

Serviço de qualidade garantida: Esse tipo de serviço estabelece limites rígidos para os atrasos de fila que um pacote sofrerá em um roteador.

Serviço de rede de carga controlada: Esse tipo de serviço pode admitir que uma "porcentagem muito alta" de seus pacotes passe com sucesso pelo roteador e sofram um atraso próximo de zero na fila do roteador. Nesse tipo de serviço não há garantias quantitativas de desempenho.

Para oferecer tais serviços todos os roteadores ao longo do caminho devem ser capazes de realizar reserva de recursos e manter o estado dos fluxos de dados transmitidos. Um

⁴*Internet Engineering Task Force.*

⁵Service Differentiation.

⁶Integrated Services.

dos protocolos amplamente utilizados é o de sinalização *RSVP*⁷ (Braden *et al.*, 1997), que atua sobre o *IP*, ocupando o lugar do protocolo de transporte.

Para oferecer qualidade de serviço a uma conexão, o *RSVP* estabelece um caminho entre os sistemas origem e destino, verificando se os nós intermediários suportam a *QoS* desejada, e reservando uma quantidade necessária para a aplicação. Para tanto, todos os nós intermediários devem suportar esse protocolo. O protocolo *RSVP* utiliza outros protocolos para efetuar roteamento e transmissão pois, seu único objetivo é a reserva, manutenção e liberação de recursos quando solicitado.

Serviços Diferenciados

DiffServ ou Serviços Diferenciados (Blake *et al.*, 1998) é um modelo de implementação de *QoS* em redes de computadores, criado pelo *IETF*, que utiliza o conceito de classes de serviço. Esse modelo visa agrupar fluxos de tráfego semelhantes em uma mesma classe e apresenta boa escalabilidade, tornando possível o tratamento de uma grande quantidade de fluxos, por possuir uma baixa granularidade. Dessa forma, supre uma das principais deficiências do modelo *IntServ* que, ao tratar fluxos individuais, torna-se inviável em um *backbone*, devido a alta carga de sinalização. No entanto, perde-se em eficiência na alocação de recursos, pois o fluxo mais exigente é quem determinará os recursos a serem reservados para toda a classe.

Em uma solicitação de serviço diferenciado, um contrato deve ser estabelecido entre usuário e provedor. Nesse contrato, chamado de *SLA*⁸ (*Service Level Agreement*), o usuário compromete-se a gerar um fluxo com as características dispostas no contrato, e o provedor compromete-se a encaminhá-lo com a *QoS* solicitada pelo usuário. Em sua maioria, os *SLAs* são estáticos, ou seja, a configuração é feita manualmente, o que não acarreta em problemas na sua administração, pois, os contratos possuem um tempo de vida maior do que aqueles feitos no *IntServ*, onde para cada novo fluxo é criado um *SLA*.

2.1.3 Qualidade de Serviço em Nível de Aplicação

As sessões anteriores descreveram técnicas de garantias de *QoS* em nível de rede, onde há muitos esforços para o aprimoramento dessa tecnologia. Entretanto, qualquer esforço no sentido de oferecer diferenciação de serviço tem maior efeito se houver cooperação entre os elementos da rede, o que implica na consideração de garantias de *QoS* em nível de aplicação.

Diversos trabalhos têm enfocado modelos de provisão de *QoS* em nível de aplicação, tendo como objeto de estudo servidores *Web*. Inspirado no modelo *DiffServ*, pertinente ao nível de rede, Teixeira *et al.* (2004) e Teixeira *et al.* (2005) fazem uma transposição

⁷Resource Reservation Protocol.

⁸Modelos de contratos também negociados no modelo de serviços integrados.

de seus princípios para a camada de aplicação, concebendo um modelo de Servidor *Web* com Diferenciação de Serviços (*SWDS*⁹). Nesse, e em trabalhos derivados, o escalonamento convencional, baseado em *FIFO*, da fila de requisições pendentes em um servidor *Web* (ou *cluster* 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 classe inferiores. Diferentes mecanismos de proteção contra negação de serviço (como envelhecimento ou prioridade não determinística) são utilizados. Em continuidade, Estrella *et al.* (2006), Traldi *et al.* (2006) e Messias *et al.* (2007) complementaram a arquitetura com um mecanismo de negociação ao módulo de controle de admissão, algoritmos de diferenciação de serviços para o fornecimento de qualidade de serviço (*QoS*) (*RSVAdap*¹⁰ e *WFQ*¹¹) e implementação e avaliação do modelo *SWDS*, respectivamente. Vários outros trabalhos têm sido desenvolvido com o objetivo de garantir confiabilidade nos serviços prestados nesse tipo de ambiente imprevisível (Kang *et al.*, 2003), (Zhou *et al.*, 2004), (Ye *et al.*, 2005), (Wu *et al.*, 2006) e (Pan *et al.*, 2008).

Esse tipo de abordagem é o que pode ser denominada *QoS* relativa, pois, os parâmetros de qualidade oferecidos a uma classe de serviço são formulados com respeito à qualidade oferecida à outra classe. Uma abordagem alternativa é a de *QoS* absoluta, na qual os parâmetros de contrato são estabelecidos de modo independente, sem relação com os parâmetros da outra classe. Um modelo de *QoS* absoluta pode ter parâmetros de contrato como tempos médios de resposta a serem garantidos a uma classe de serviço. Para que sejam cumpridos os prazos de atendimentos dessas classes de serviços, são usados algoritmos de escalonamento de tempo-real. Por exemplo, Casagrande *et al.* (2007) introduz uma política de escalonamento de tempo-real não-determinístico (*soft real-time*) para provisão de *QoS* absoluto em nível de aplicação, denominada *EBS* (*Exigency Based Scheduling*). Em continuidade a esse trabalho, Peixoto *et al.* (2007), Peixoto & Monaco (2008) e Monaco *et al.* (2009) desenvolveram um estudo no qual estendem essa abordagem para um ambiente de multi-recursos. E há também trabalhos que relacionam essas duas arquiteturas de *QoS*, como em (Shenoy & Patchararungruang, 2004).

2.2 Tempo-Real

Garantias de requisitos de *QoS* relacionados ao tempo inserem nos problemas tratados no domínio tecnológico de sistemas de tempo real. Com o crescimento da capacidade computacional, sistemas de tempo-real (TR) estão sendo cada vez mais comuns e empregados em aplicações (controle de aviões e tráfego aéreo, robótica, monitoramento de pacientes, realidade virtual) complexas e de grande importância econômica e social. Tais

⁹Servidor Web com Diferenciação de Serviços.

¹⁰Reserva Adaptativa de Recursos.

¹¹Weighted Fair Queuing.

aplicações não implicam em desempenho computacional, mas sim, em assegurar os tempos de resposta do sistema aos eventos externos.

2.2.1 Conceitos Básicos de Tempo-Real

Para sistemas computacionais convencionais serem considerados corretos, eles precisam apresentar uma saída adequada a uma entrada. Há determinados sistemas computacionais que possuem, em suas especificações - restrições temporais - esses são denominados sistemas de tempo real. Em outras palavras, para tais sistemas, estar correto, depende tanto do resultado lógico da computação, quanto do tempo no qual esse resultado foi produzido (Ramamritham & Stankovic, 1994). Assim, além de produzir o resultado esperado deve-se fazê-lo no prazo previamente determinado.

Para a caracterização dos diferentes tipos de sistemas de TR, bem como seus métodos para escalonamento e gerenciamento de recursos, utilizam-se termos gerais para tratar a carga de trabalho de sistemas computacionais e de comunicação. Sendo assim, o trabalho irá adotar requisição como sendo a unidade de trabalho agendada (ou *job*), enquanto que o grupo de requisições que conjuntamente realizam alguma função no sistema, recebe o nome de tarefa (ou *task*).

Existem três modelos de tarefas importantes na área de sistemas de TR, são elas:

- Tarefas Periódicas: quando existem várias instâncias da tarefa e as ativações ocorrem em intervalos regulares, denominados *períodos*;
- Tarefas Esporádicas: apresentam zero ou mais instâncias, e as ativações consecutivas ocorrem após um espaço mínimo de tempo;
- Tarefas Aperiódicas: são tarefas esporádicas com frequência de ocorrência e duração indeterminadas.

Restrições de Tempo

As especificações dos requisitos de sistemas de TR são feitas, geralmente, por meio de parâmetros temporais de suas tarefas. Esses são usados tanto para caracterizar o sistema em questão, quanto para impor o comportamento temporal que deve ser obedecido durante a execução das tarefas. Por exemplo, o limite de tempo para execução de uma determinada tarefa é especificado pelo atributo *deadline*, enquanto que o comportamento temporal relacionado à frequência de execução pode ser especificado como período.

Pode-se citar outras restrições temporais que também são importantes para definir o comportamento temporal:

- Tempo de Início: corresponde ao instante em que uma requisição começou a ser executada.

- Tempo de Término: corresponde ao instante em que a requisição conclui sua execução.
- Tempo de Execução: corresponde ao tempo gasto para completar a execução de uma dada requisição, quando não a preempção e de posse de todos os recursos necessários.
- Tempo de Resposta: corresponde ao tempo desde que uma requisição foi designada ao sistema até o instante em que ela terminou de executar.
- Tempo de Chegada: corresponde ao tempo em que o escalonador toma conhecimento da ativação da nova requisição.
- Tempo de Liberação: corresponde ao tempo que a requisição é incluída na fila de prontas para serem executadas.
- *Release Jitter*: corresponde à variação dos tempos de respostas de um conjunto de requisições.

2.2.2 Sistemas Hard-TR, Soft-TR e Firm-TR

Os sistemas de tempo real podem ser classificados, segundo suas restrições temporais, em três tipos: **Hard-TR**, **Soft-TR** e **Firm-TR**.

Em sistemas *Hard-TR*, a exatidão do tempo de resposta é criticamente importante, e não pode ser depreciada como alternativa para outros ganhos. Nesses sistemas, o não cumprimento das restrições temporais (como tempo de liberação e *deadline*) acarretaria graves conseqüências. Sistemas *Soft-TR* são mais flexíveis, ou seja, o atraso na conclusão de uma tarefa é indesejável porém aceitável, pois, se acontecer esporadicamente não acarretará danos sérios, apenas a degradação do desempenho a cada não cumprimento do *deadline*. Para sistemas *Soft-TR* as tarefas que violaram o *deadline* podem ou não finalizar sua execução. Em sistemas *Firm-TR* também é aceitável a perda ocasional do *deadline* das tarefas, porém, essas devem ser descartadas pelo sistema.

2.2.3 Escalonamento de Sistemas de Tempo Real

O escalonador é componente básico para a garantia de bom funcionamento em sistemas computacionais, pois, é responsável por decidir qual tarefa irá executar quando houver mais de uma pronta para execução. O algoritmo utilizado para tal finalidade é denominado algoritmo de escalonamento, o qual segue um critério para escolher uma tarefa em detrimento de outras. Em sistemas de tempo real, a relevância do escalonador é ainda mais evidente, visto que está ligado à correção do sistema, pois, a tarefa escalonada deve cumprir as restrições temporais.

Classificação dos Algoritmos de Escalonamento

Os algoritmos de escalonamento para sistemas de TR podem ser classificados de acordo com suas características, dentre elas: preempção, instante de utilização dos valores dos critérios e quanto ao sincronismo.

- **Quanto à Preempção**

- **Preemptivas:** uma tarefa pode ser temporariamente suspensa para a execução de outra mais prioritária.
- **Não-preemptivas:** uma vez iniciada a execução da tarefa, não pode ser interrompida até que seja terminada.

- **Quanto ao instante de utilização dos valores dos critérios**

- **Estático:** algoritmos de escalonamento estáticos assumem o conhecimento prévio das características relevantes de todas as requisições.
- **Dinâmico:** algoritmos de escalonamento dinâmicos são projetados para trabalhar com requisições imprevisíveis quanto ao tempo de chegada, e, possivelmente, com tempo de execução desconhecido.
- **Misto:** um escalonamento misto utiliza elementos dos dois anteriores, escalonando estaticamente as requisições por meio de características previamente conhecidas e ajustando as outras à medida que chegam.

Quanto ao Sincronismo

- **Orientado a Tempo:** toma decisões sobre qual requisição executar de acordo com instantes previamente estabelecidos para cada uma antes da execução do sistema.
- **Orientada à Evento:** toma decisão de escalonamento baseada em eventos (*event-driven*). Uma requisição $T1$ é dita ter uma prioridade maior sobre outra $T2$ quando essa última tem que ser interrompida para permitir a execução de uma nova requisição de $T1$. Se mais de uma requisição com alta prioridade for requerida simultaneamente, a com maior prioridade dentre todas é executada. Em caso de equivalência de prioridades, uma é escolhida arbitrariamente.

Escalonamento de Requisições Periódicas

Considerando-se aplicações de tempo real de uma maneira geral, seja controle de processos ou aplicações de multimídia, as atividades nelas envolvidas caracterizam-se basicamente pelo comportamento periódico de suas ações. As características de tarefas

periódicas que determinam o conhecimento *a priori* dos tempos de chegada e, por consequência, da carga computacional do sistema, permitem que se obtenha garantias em tempo de projeto sobre a escalonabilidade de um conjunto de tarefas periódicas.

As requisições de tarefas periódicas possuem tempo de chegada, período, tempo de execução e *deadline*. O comportamento temporal de uma requisição periódica T_i é descrito pela quádrupla (J_i, C_i, P_i, D_i) , sendo:

- J_i : *release jitter*¹² da requisição;
- C_i : tempo de execução da requisição;
- P_i : período da requisição;
- D_i : *deadline* da requisição.

O escalonamento de tarefas periódicas é baseado em prioridades fixas ou dinâmicas. Para o escalonamento baseado em prioridades fixas, as prioridades atribuídas aos *jobs* do conjunto são iguais e derivadas de suas restrições temporais. Por outro lado, o escalonamento baseado em prioridades dinâmicas atribui prioridades diferentes para cada *job* isoladamente. Entre os algoritmos de escalonamento clássicos de prioridade fixa estão *Rate Monotonic*(Liu & Layland, 1973) e *Deadline Monotonic*(Leung & Whitehead, 1982). Para prioridade dinâmica cita-se o algoritmo de escalonamento *Earliest Deadline First*(Liu & Layland, 1973), adequado, também, para tarefas não periódicas.

Escalonamento de Tarefas Aperiódicas e Esporádicas

Os algoritmos apresentados anteriormente são desenvolvidos para efetuar escalonamento de tarefas periódicas. Quando tarefas periódicas ou esporádicas coexistem com tarefas aperiódicas é necessário prover uma maneira para que se possa escalonar as últimas sem comprometimento das primeiras. Para tarefas aperiódicas não é possível fazer nenhuma previsão sobre o intervalo de chegada ou sobre o tempo necessário para sua execução. Já no caso de tarefas esporádicas, as requisições são descritas pela tripla (C_i, D_i, min_i) , onde:

- C_i : tempo de execução da requisição;
- D_i : *deadline* relativo da requisição;
- min_i : intervalo mínimo entre duas requisições consecutivas.

Cabe aos algoritmos de escalonamento, que consideram também as tarefas aperiódicas e esporádicas, determinar em que momento deve-se executá-las. O objetivo dos algoritmos de tarefas aperiódicas é escalonar essas tarefas assim que possível, sem causar perdas

¹²Variação dos atrasos.

de *deadline* das periódicas ou das esporádicas. Basicamente, existem dois tipos de algoritmos para o escalonamento de requisições aperiódicas: *bandwidth-preserving* e *slack-stealing* (Liu, 2000). Já o escalonamento de requisições esporádicas é baseado no tempo de execução e *deadline* de cada nova tarefa esporádica, em que o escalonador decide se aceita ou rejeita a tarefa. Se a requisição for aceita, ela é escalonada de forma que termine no tempo desejado, sem causar impacto negativo nas requisições periódicas e esporádicas previamente aceitas. Caso seja impossível para alguma requisição esporádica atender seu *deadline*, consideram-se duas alternativas: rejeitar a requisição esporádica que não pode finalizar no tempo determinado, ou aceitar todas as requisições esporádicas e permitir a algumas delas finalizar atrasada.

Alguns algoritmos de escalonamento para esses casos são voltados para políticas baseadas em prioridades fixas, do quais é possível citar: Servidor de *Background*, *Polling Server*, *Deferrable Server* e *Slack Stealing*.

2.3 Simulação de Sistemas

Simulação é a reprodução da composição dinâmica de um sistema do mundo real sobre o tempo, descrita em termos de atividades, processos e eventos (MacDougall, 1989). Pela observação do histórico artificial por ela gerada, busca-se inferir características operacionais de um sistema real, estudando, cientificamente, seu comportamento. Isso é útil quando a complexidade de um sistema real inviabiliza seu estudo por experimentação, quando se pretende analisar isoladamente 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.*, 1989).

A simulação é representada através de modelos, que podem ser considerados como uma descrição do sistema real. A execução de modelos de simulação em computadores tem potencial para fornecer resultados mais precisos sem que seja necessário interferir no sistema real. Tais resultados, quando analisados estatisticamente, produzem informações que podem contribuir na tomada de decisões que visam a solução de problemas.

Alguns termos são importantes para representar um modelo, logo, define-se estado de um sistema como uma coleção de variáveis necessárias para descrever um sistema em um dado período. O estado pode ser classificado como discreto ou contínuo. Um sistema discreto é aquele cujo estado sofre mudanças instantâneas em pontos distintos de tempo, como por exemplo, o número de clientes na fila. Já nos sistemas contínuos, as mudanças ocorrem continuamente ao longo do tempo, como o volume de água em uma hidrelétrica. Os modelos de simulação ainda pode ser classificados da seguinte forma:

- **Estáticos ou Dinâmicos:** denominam-se como modelos estáticos os que visam representar o estado de um sistema em um instante ou que, em suas formulações, a

variável tempo não seja considerada, enquanto os modelos dinâmicos são formulados para representarem as alterações de estado do sistema ao longo da contagem do tempo de simulação;

- **Determinístico ou Estocásticos:** são modelos determinísticos os que em suas formulações não fazem uso de variáveis aleatórias, enquanto que os estocásticos utilizam-se dessas variáveis.

Assim, pode-se formalizar o tipo de simulação utilizado neste trabalho como simulação baseada em eventos discretos, com alterações de estado ao longo do tempo (dinâmico) e com parâmetros estocásticos.

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

2.3.1 Simulação de Filas

As filas estão presentes em muitos pontos de serviços como bancos, supermercados, entre outros. A formação de filas ocorre porque a procura pelo serviço é maior do que a capacidade de atender a essa procura. Dessa forma a Teoria de Filas tenta, através de análises matemáticas detalhadas, encontrar um ponto de equilíbrio que satisfaça o cliente e que seja viável para o provedor do serviço.

Elementos de uma Fila

A Figura 2.1 mostra os elementos que compõem uma Fila.

Figura 2.1: *Elementos de uma Fila (Prado, 2004).*

Pode-se observar que a partir de uma população surgem clientes¹³ que formam uma fila e aguardam por um tipo de serviço, oferecido pelos servidores. Quando um servidor está

¹³O termo cliente é usado de maneira geral e não implica necessariamente em um cliente humano, pode ser um processo esperando para utilizar o servidor.

disponível, um cliente pode iniciar o atendimento. Quando esse atendimento é finalizado, o cliente sai e o servidor pode atender outro cliente.

Características dos Processos de Fila

Na maioria dos casos, seis características básicas de processos de filas fornecem uma descrição adequada de um sistema de filas (Prado, 2004):

1. Padrão de chegada dos clientes;
2. Padrão de serviço dos servidores;
3. Disciplina de filas;
4. Capacidade do sistema;
5. Número de canais de serviço; e
6. Número de estágio de serviços.

Essas seis características de sistemas de filas são suficientes para descrever um processo sob estudo. Assim, para descrever adequadamente o processo sendo modelado, o conhecimento das seis características básicas são essenciais.

Modelos de Redes de Filas

De acordo com o comportamento de entrada e saída dos usuários, os modelos de Redes de Filas podem ser classificados em:

- **Modelos Abertos:** Os usuários entram no sistema e, obrigatoriamente, saem do mesmo;
- **Modelos Fechados:** O número de usuários que circulam pelo sistema é fixo, de forma que nem entram e nem saem usuários;
- **Modelos Mistos:** Para certas classes de usuários o modelo é aberto, enquanto que para outras é fechado.

A figura 2.2 ilustra os três modelos de Redes de Filas (Lazowska *et al.*, 1984).

Para facilitar o estudo da Teoria de Filas, foram definidos padrões de representação do modelo, entre eles tem-se a notação de Kendall, representada da seguinte forma: $A/B/m/K/N/Q$, onde:

- A : Distribuição dos intervalos entre chegadas.
- B : Distribuição dos tempos de serviço.

Figura 2.2: *Modelos de Redes de Fila.*

- m : Número de servidores.
- K : Capacidade do sistema (exemplo: tamanho do *buffer*, número máximo de clientes no sistema).
- N : Tamanho da população (finita ou infinita).
- Q : Disciplina de atendimento (*FIFO*, *LCFS*, etc).

Geralmente omitem-se os parâmetros K ou N quando assumem valores infinitos. Os valores A e B são dependentes do tipo de distribuição da probabilidade que eles se referem:

- M : Exponencial Negativa (ou Marcoviana ou Poisson);
- Em : Erlang de estágio m ;
- Hm : Hiper-exponencial de estágio m ;
- D : Determinística;
- G : Distribuição arbitrária.

Segundo Jain (1991), a distribuição exponencial é usada extensivamente em modelagem de filas.

2.3.2 Probabilidade e Estatística para Simulação

Em estudos de simulação, inferências estatísticas devem ser realizadas baseadas nos resultados experimentais obtidos da simulação. Devido ao fato da simulação possuir elementos aleatórios, as saídas de uma simulação são amostras observadas de variáveis aleatórias e, em consequência, qualquer assertiva que se fizer quanto à operação do sistema, baseada em resultados de simulação, deve levar em consideração a variabilidade dessas saídas. Essa variabilidade é levada em consideração pelo uso de intervalos de confiança ou através de hipóteses de teste.

Para a definição desta variabilidade, pode ser feita uma análise estatística detalhada (durante o processo de obtenção de informação) sobre a precisão e sensibilidade do modelo.

Basicamente, deve-se estudar qual saída seria obtida caso a simulação fosse realizada novamente ou se tivesse um tempo de execução maior. 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.

Para uma análise adequada de um modelo utiliza-se uma série de ferramentas que são apresentadas de forma sucinta a seguir.

Média

A média é obtida pela Equação 2.1, em que se relaciona o somatório de todos os elementos de um determinado conjunto, $\sum_{i=1}^n x_i$, com o número de elementos do conjunto, n .

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (2.1)$$

A média é um importante parâmetro utilizado, na prática, para obter estimativa do comportamento do sistema. Por exemplo, em uma rede de computadores, parâmetros comumente usados para avaliação de desempenho são vazão média e atraso médio.

Variância e Desvio Padrão

Somente com a média não é possível descrever o comportamento do sistema. Muitas vezes a média $\bar{x}$ oculta a variações dos dados, ou seja, a distância das amostras em relação ao valor médio. Por exemplo, a média dos dois conjuntos $A=\{5,10,15\}$ e $B=\{0,10,20\}$ é a mesma, mas o espalhamento é diferente. Nesse caso, um parâmetro para medir a dispersão dos dados em relação ao valor médio é importante.

A variância e o desvio padrão são parâmetros que medem a dispersão dos dados em relação à média. A variância é definida como o desvio quadrático em relação a média - tradicionalmente denotada por σ^2 , e é calculada de uma amostra de dados a partir da Equação 2.2, onde n é o número de elementos da amostra coletada:

$$\sigma^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2.2)$$

A raiz quadrada da variância é chamada de desvio padrão e é representado por σ na Equação 2.4.

$$\sigma = \sqrt{\sigma^2} \quad (2.3)$$

Intervalo de Confiança

Como não é possível encontrar uma estimativa perfeita para a média a partir de um número finito de amostras, a melhor opção é obter limites probabilísticos, por exemplo, c_1 e c_2 , de forma que seja alta a probabilidade $(1 - \alpha)$ de a média exata pertencer ao intervalo $[c_1; c_2]$.

$$P(c_1 \leq \bar{x} \leq c_2) = 1 - \alpha \quad (2.4)$$

O intervalo (c_1, c_2) é denominado intervalo de confiança para a média, α é chamado nível de significância e $1 - \alpha$ é chamado coeficiente de confiança. Tipicamente, o intervalo de confiança é expresso em porcentagem próximo a 100%, por exemplo 90% ou 95%, enquanto o nível de significância α é expresso em fração, geralmente próximo de zero, por exemplo, 0,05 ou 0,1.

2.4 Considerações Finais

O presente capítulo apresentou uma visão geral sobre Qualidade de Serviço, abordando conceitos básicos, e os dois modelos da sua arquitetura: o modelo *IntServ*, cuja principal característica é manter uma determinada qualidade de serviço durante a conexão com a aplicação destino por meio de reserva de recursos; e o modelo *DiffServ*, que caracteriza a forma de encaminhamento de pacotes nas redes. Além da diferenciação de serviços no nível de rede, evidenciou-se a importância do fornecimento de *QoS* no nível de aplicação, citando os esforços realizados por diversos pesquisadores para melhorar a performance dos servidores *Web*.

Para estabelecer parâmetros de serviço em termos temporais utilizam-se conceitos de Tempo Real. Assim, definiu-se alguns conceitos básicos utilizados para a descrição da área e abordando as principais restrições temporais utilizadas na literatura. Em seguida, foi abordada a classificação de sistemas TR em três vertentes: *Hard-TR*, *Soft-TR* e *Firm-TR*. Mostrou-se também algumas formas de classificação para o escalonamento de tarefas de TR, citando alguns algoritmos clássicos da literatura.

Por fim, para o estudo científico de um sistema *Web* utilizou-se a teoria de modelagem, e a simulação como técnica de análise do modelo e das políticas propostas. Conceitos gerais sobre simulação baseada na Teoria de Fila foram apresentados, como: elementos que compõem uma fila, características dos processos da fila, classificação de modelos e a notação de Kendall. Como ferramentas oferecidas pela probabilidade e estatísticas são de extrema importância para uma análise adequada de sistemas, seus principais conceitos também foram abordados.

EBS: Exigency Based Scheduling

Desenvolvida com o objetivo de atender as necessidades impostas por aplicações com restrições temporais, a política de escalonamento *EBS* (*Exigency Based Scheduling*) oferece reações imediatas às requisições mais urgentes. O presente capítulo apresenta com mais detalhes essa política de escalonamento, visto que essa abordagem é a base para realização deste trabalho.

3.1 Política de Escalonamento *EBS*

A *EBS* é uma política de escalonamento de tempo-real híbrida, ou seja, leva em conta o custo de execução de uma requisição e o tempo de espera em fila, distingue-se por buscar impor ao sistema a menor demanda de recursos possível. Essa política garante um compromisso entre desempenho e confiabilidade no atendimento de contratos dos clientes, individualmente. Em sistemas interativos *online*, como servidores *Web*, a *EBS* proporciona resultados superiores às políticas convencionais (Casagrande *et al.*, 2007) .

Essa política é baseada no princípio teoria de *feedback scheduling*, fazendo atualizações do *deadline* sempre que o servidor terminar de atender uma dada requisição do usuário. Dada a Equação 3.1, defini-se o valor de Ω_u como sendo o tempo médio de resposta do usuário u . Esse tempo médio de resposta é obtido através da média entre o antigo tempo médio de resposta do usuário, (Ω'_u) , e o tempo de resposta da requisição j recém atendida. O valor de $time()$ representa o tempo atual, $timeStamp_j$ o tempo de chegada da requisição j e R_u o número de requisições anteriormente submetidas por u .

$$\Omega_u = \frac{(\Omega'_u \cdot R_u) + (time() - timeStamp_j)}{R_u + 1} \quad (3.1)$$

À medida que as requisições chegam ao sistema e não encontram servidor disponível, essas são transferidas para uma fila única de espera. Ao término da execução de uma requisição, o escalonador recalcula o tempo médio de resposta real daquele usuário, e em seguida busca na fila a requisição mais urgente, ou seja, aquela que tolera o menor tempo de espera dentre as existentes na fila de espera do sistema. Essa, terá acesso ao servidor, e o ciclo se repete.

O tempo de espera máximo, também denominado *deadline*, representa o quanto uma requisição ainda pode esperar na fila antes de começar a descumprir o contrato do usuário responsável por ela. Para obtê-lo, isola-se na Inequação 3.2 a variável D_j , que representa o *deadline* da requisição j . Como o interesse é obter o máximo tempo de espera aceitável pelo usuário u antes que o valor do seu tempo médio de resposta real (Ω_u) ultrapasse o seu tempo médio de resposta contratado, igualam-se os termos. Os valores R_u e T_{w_j} expressam respectivamente o número de requisições feitas pelo usuário u e o tempo de espera em fila da requisição j , até o momento.

$$\frac{(\Omega_u \cdot R_u) + T_{w_j} + D_j}{R_u + 1} \leq \Omega_{c_u} \quad (3.2)$$

Quanto menor o valor de D_j , maior é a prioridade atribuída à requisição, pois terá maior urgência. Basicamente, esse é o mesmo princípio utilizado pela política de escalonamento *EDF*, porém mais flexível, pois, mantém valores médios de tempo de resposta do sistema.

Em alguns casos podem existir algumas requisições com os mesmos valores de *deadline*, porém a exigência da requisição imposta ao sistema é dada em conjunto com o peso ou custo computacional dela; pois, realocar as requisições com menores tempos de processamento, como sendo as primeiras, possibilita uma diminuição na média de tempo de resposta oferecida pelo sistema, princípio da política de escalonamento *SJF*.

Na abordagem de escalonamento desenvolvida, a prioridade das requisições são atribuídas segundo seu *deadline* e no valor esperado do tempo de processamento. Determinada pela Equação (3.3), a prioridade de uma dada requisição j em fila, do usuário u , é dada por P_j . Quanto menor for o valor de P_j maior será a prioridade de escalonamento da requisição j , conseqüentemente as requisições que apresentarem menores valores de (D_j) e menores tempos de processamento (T_{p_j}) serão classificadas como mais prioritárias.

$$\begin{aligned} P_j &= D_j \cdot T_{p_j} \\ P_j &= \left(\left(\Omega_{c_u} \cdot (R_u + 1) \right) - \left(\Omega_u \cdot R_u \right) - T_{w_j} \right) \cdot T_{p_j} \end{aligned} \quad (3.3)$$

Na política de escalonamento *EBS* não há o problema de *starvation*, como acontece

com a política *SJF*, pois, utiliza também o *deadline* como critério de priorização. Assim, mesmo tendo requisições com um tempo de processamento muito alto na fila, elas não ficarão indefinidamente no sistema devido à chegada de outras mais curtas. Isso se deve ao fato de que quanto maiores os tempos na fila do sistema, menores serão *deadlines*, o que permitirão a elas serem atendidas, eventualmente.

Em alguns casos podem existir valores negativos¹ de *deadline*, como nos cenários onde o sistema está saturado por requisições mais exigentes. Nesses casos, o escalonamento das requisições torna-se o oposto do esperado. Sendo assim, foi realizada uma correção da abordagem para manter o objetivo proposto. A correção resulta na Equação (3.4), para as requisições mais urgentes que ainda não tiveram seus *deadlines* descumpridos, a prioridade é diretamente proporcional ao seu *deadline* e ao custo esperado de processamento. Para aquelas com descumprimento de *deadline* a prioridade é inversamente proporcional a seu custo esperado de processamento, já que quanto menor for esse custo menor será o valor de P_j resultante, e assim maior será sua prioridade de escalonamento.

$$P_j = \begin{cases} D_j \cdot T_{p_j} & \text{se } D_j \geq 0 \\ D_j \cdot \frac{1}{T_{p_j}} & \text{se } D_j < 0 \end{cases} \quad (3.4)$$

Dessa forma garante-se que as requisições mais urgentes, independente de terem descumprido ou não seus *deadlines*, e com menores custos esperados de processamento sejam escalonadas primeiro.

3.2 Considerações Finais

Esse capítulo apresentou a política de escalonamento de tempo real *EBS*, princípio básico para o entendimento dos capítulos posteriores, dos quais mostrarão avaliações dessa política sobre contratos em janelas deslizantes de operações e o desenvolvimento de políticas que reduzem a dispersão dos atrasos, do qual corresponde ao tempo do tempo de resposta que ultrapassou o valor contratado.

¹Um valor negativo de *deadline* significa que o contrato foi violado, ou seja, o tempo médio efetivo do usuário está acima do valor contratado.

Avaliação da *EBS* em um *SLA* com Janelas Deslizantes

Neste capítulo são apresentados os procedimentos utilizados para o desenvolvimento de parte da pesquisa. Essa primeira parte visa avaliar a política de escalonamento *EBS* com janelas deslizantes de operação. Inicialmente são apresentadas, na Seção 4.1, as modificações impostas à política *EBS* para se obter as janelas de operações. Em seguida, apresentam-se, na Seção 4.2, os materiais e métodos adotados para realização do trabalho. Os experimentos são apresentados na Seção 4.4, indicando quais fatores podem influenciar no comportamento do escalonador. A medida para validar a política com contratos definidos em janelas deslizantes de operações é discutida na Seção 4.3. Finalmente, os resultados obtidos são apresentados na Seção 4.5.

4.1 *SLA* com Janelas Deslizantes

A *EBS* proporciona resultados superiores às políticas convencionais. Esses resultados são satisfatórios por estabelecerem como parâmetro de contrato de *QoS* a satisfação contratual, proveniente da média de tempo de resposta durante todo o período de serviço prestado ao usuário. Ou seja, a saída de uma requisição no sistema gera um tempo de resposta, e esse, é somado com uma infinidade de tempos de resposta de diferentes dispersões analisados sobre o total de requisições no sistema. Dessa forma, os resultados mostraram que o comportamento da média efetiva dos tempos de resposta, à medida que o tempo passa, torna-se aproximadamente constante (Casagrande *et al.*, 2007).

O *SLA*¹, definido em janelas deslizantes, determina, então, que o tempo médio de

¹Service Level Agreement.

resposta esqueça as solicitações envelhecidas do usuário. O objetivo do método é obter a média de tempo de resposta durante um período recente que o usuário esteja utilizando o serviço *Web*. Dessa forma, todas as variáveis provenientes da média de tempo de resposta, como o *deadline*, serão modificadas para análise somente das requisições mais recentes do usuário.

As seções de operações ou janelas de operações dos usuários são caracterizadas por uma lista circular. Insere-se em cada posição da janela, o tempo de resposta da requisição de um dado usuário que acabou de ser atendida pelo servidor. Até que a janela seja completada por tempos de resposta, a média é desconsiderada pela medida, atribuindo a esse tempo o período de aquecimento *warm up*, entretanto, é computada pelo esquema original (Equação 3.1). Depois que a janela estiver completa por tempos de resposta, a cada liberação de uma requisição no servidor, o tempo de resposta mais antigo dentro da janela é retirado para a entrada de um novo tempo de resposta. Em seguida, o cálculo da média é refeito.

A Figura 4.1 exemplifica o *SLA* com janela deslizante. Como mencionado anteriormente, o método é acionado sempre que a janela do usuário estiver completa por tempos de resposta. Sendo assim, a figura caracteriza uma janela com 10 posições, sendo reestruturada a cada liberação do servidor pelos tempos de resposta das requisições de um dado usuário. Logo, o primeiro vetor da figura mostra o primeiro passo que é identificar o tempo de resposta mais antigo. Sendo o tempo 6 da primeira posição, esse, é retirado da janela para a entrada do tempo de resposta da requisição recém atendida pelo servidor, no caso, o tempo 8, como mostra o segundo vetor da figura. Os demais vetores exemplificam como ficariam os próximos passos com mais liberações de requisições provenientes do mesmo usuário. As unidades dos tempos de resposta não são especificadas no trabalho, logo, adota-se u.t que se refere a unidade de tempo.

Figura 4.1: *Janelas Deslizantes de Operações.*

Para se obter as janelas ou seções de operações dos usuários, modificações no tempo médio de resposta são necessárias. Para simplificar a equação, sem comprometimento com o desempenho do cálculo, o tempo médio de resposta, referenciado na Equação 4.1, é obtido através da média entre o antigo tempo médio de resposta de um dado usuário, (Ω'_u) , e a diferença entre o tempo de resposta da requisição j recém atendida e o tempo

de resposta da requisição mais antiga ($W[x]_i$) desse mesmo usuário. O índice x indica a posição na janela que possui a requisição com o tempo de resposta mais antigo dentro da janela W .

$$\Omega_u = \Omega'_u + \frac{((time() - timeStamp_j) - W[x]_{i_u})}{wS_u} \quad (4.1)$$

Onde, wS_u é um número fixo que especifica o tamanho da janela dos usuários. Resumidamente, o que será adicionado ao tempo médio de resposta efetivo do usuário, é a diferença entre, o antigo tempo de resposta de uma dada requisição e o novo tempo de resposta da requisição que acabou de ser atendida pelo servidor, sobre o tamanho da janela determinado por wS_u .

Visto que o tempo de espera máximo é determinado pelo quanto a requisição ainda pode esperar até que o contrato seja violado, alterações no *deadline* das requisições são imprescindíveis para que se obtenha o resultado correto. Como feito na média, o *deadline* inicialmente é formulado pela expressão original (Equação 3.3). Quando a janela estiver completa por tempos de resposta, a espera máxima da requisição na fila é caracterizada também pelo tamanho da janela. A Equação 4.2 determina o *deadline* com as alterações necessárias.

$$D_j = \left((\Omega_{c_u} \cdot wS_u) - (\Omega_u \cdot wS_u) - T_{w_j} \right) \quad (4.2)$$

Existe uma implicação nesse método em janela: para conseguir a janela de operações ou seções do usuário é necessário armazenar um vetor com um tamanho de acordo com o vinculado aos experimentos. Porém, o algoritmo é de complexidade $O(1)$, pois, é necessário apenas uma única operação de busca para encontrar o tempo de resposta mais antigo dentro da janela.

4.2 Materiais e Métodos

Para avaliar o modelo de SLA com janelas deslizantes, utilizou-se como abordagem a modelagem e a simulação orientada a eventos (Seção 2.3). Além de permitir maior flexibilidade para o estudo, a simulação permite também obter boas estimativas sobre o comportamento do sistema após as modificações. Modelos mais complexos e representativos do problema real podem ser construídos a partir da simulação, proporcionando melhores previsões de desempenho e alternativas de comparação, com um tempo de processamento baixo.

Com os conceitos de redes de fila (Seção 2.3.1), o modelo utilizado para representar o servidor *Web* determina eventos que tratam a chegada da requisição ao sistema, a solicitação do serviço e a liberação do recurso, oferecendo também suporte à *QoS*. De acordo

com a notação de Kendall, o modelo é equivalente a $M/M/1$, onde a distribuição exponencial especifica o intervalo entre as chegadas das requisições e os tempos de processamento das requisições. O servidor modelado é do tipo mono-processado, com uma fila única de espera para processamento. A política já implementada *EBS* foi alterada para incluir o modelo de *SLA* com janelas deslizantes.

Para validar o modelo e a política utilizou-se o pacote de simulação *SimpackJ* (Fishwick, 1992), o qual oferece um conjunto de ferramentas voltado para a simulação orientada a eventos. Esse pacote é constituído por uma biblioteca de programação com ferramentas para simulação de redes de fila. A ferramenta foi escolhida devido as experiências do grupo SDPC e por causa das alterações da biblioteca já feitas por trabalhos anteriores para comportar requisições *Web*.

Para o controle da qualidade de serviço oferecido, o parâmetro contratado pelo cliente é o tempo médio de residência das requisições do usuário no sistema (Seção 4.1) durante seções.

Por fim, a medida utilizada para avaliar a satisfação contratual em cumprimento ao contrato relaciona o número de vezes que a média contratual foi cumprida, pelo número total de requisições submetidas ao sistema (Equação 4.3), também conhecida como *Deadline Miss Ratio*, detalhada a seguir.

4.3 Satisfação Contratual

Considera-se que um usuário estará satisfeito com o serviço a ele oferecido quando obtiver uma alta porcentagem de requisições atendidas abaixo do tempo médio contratado (definido na próxima seção). Logo, o escalonador pode tomar como índice de satisfação contratual a Equação 4.3, em que são relacionadas a quantidade total de requisições submetidas pelo i -ésimo usuário (R_i) e o número de vezes em que o tempo de resposta médio dessas requisições ficou abaixo do limiar contrato (A_i).

$$S = \frac{A_i}{R_i} \quad (4.3)$$

4.4 Planejamento de Experimentos

Para os experimentos da simulação, duas classe foram consideradas: A, como sendo a mais exigente, e B, a menos exigente ². É válido lembrar que uma classe de serviço é mais exigente que a outra, quando seu tempo médio de resposta contratado é menor.

Estabeleceu-se que cada classe de serviço (A e B) atende a 10 usuários, totalizando 20 usuários submetendo requisições ao sistema. Dentro de uma mesma classe, todos os

²Exigência refere-se aos tempos contratados pelo usuário (Casagrande *et al.*, 2007).

usuários tem a mesma chance de fazer requisições *Web*, descrita por uma distribuição uniforme.

Dentro de cada classe, cada um dos seus 10 clientes exibe o mesmo modelo de geração de requisições sintéticas, caracterizado pelo intervalo entre chegadas e o tempo de execução da tarefa, ambos descritos pelas distribuições exponenciais com média 4 u.t. e 3 u.t., respectivamente. Assim, o sistema possui uma taxa de utilização de 75%.

Os cenários investigados nesse experimento enfocam três dimensões: a proporção de requisições pertencentes a cada classe, a diferença entre o tempo médio de resposta contratado das classes de serviço e o tamanho da janela.

Para o primeiro parâmetro, especificou-se duas situações extremas, uma com um sistema sobrecarregado de requisições de baixa exigência (10%A e 90%B) e outra com um sistema sobrecarregado com requisições de alta exigência (90%A e 10%B). Definiu-se também uma situação de igualdade no número de requisições de ambas as classes (50%A e 50%B).

Para o segundo parâmetro, foi medido e tomado como referência, o tempo médio de resposta do sistema, com uma simulação da política *FIFO*³. O contrato da classe de serviço A é definido pela Equação 4.4, onde, o contrato médio de tempo de resposta é P% inferior à média de tempo de resposta da política *FIFO*. Enquanto que, para a classe de serviço B, o contrato médio de tempo de resposta é P% superior à *FIFO*, Equação 4.5. O parâmetro P foi variado em 5, 10, 20 e 30.

$$\Omega_{cuA} = \Omega_{FIFO} - (\Omega_{FIFO} \cdot P) \quad (4.4)$$

$$\Omega_{cuB} = \Omega_{FIFO} + (\Omega_{FIFO} \cdot P) \quad (4.5)$$

Para o terceiro parâmetro, como componente principal dos contratos, especificou-se janelas de tamanho 10, 25, 50, 75, 100, 150 e 250. Casagrande *et al.* (2007) abordam janela de tamanho infinito, ou seja, o tamanho da janela é superior ao total de requisições de qualquer usuário e classe.

Para obter uma boa coleta dos dados, o número de requisições no sistema para cada cenário foi definido como 100.000. Cada cenário foi simulado 30 vezes, com diferentes sementes aleatórias. Os resultados apresentados foram obtidos a partir da média dessas 30 simulações com um intervalo de confiança de 90%.

4.5 Análise dos Resultados

Os resultados serão apresentados em gráficos, onde serão mostrados os diversos cenários delineados acima (Seção 4.4). Nos gráficos, o eixo das abscissas informa os

³*First In First Out*

tamanhos de janela. O eixo das ordenadas informa as porcentagens que representam a satisfação contratual. As legendas ou as colunas, mostram a porcentagem de requisições destinadas aos usuários da classe em questão, junto as variações contratuais. São apresentados três gráficos para cada classe, cada um apresenta uma proporção de requisição, ou seja, o primeiro gráfico indica que 10% das requisições submetidas ao sistema são de usuários da classe indicada na subseção, o segundo, indica 50% das requisições submetidas ao sistema e o terceiro indica 90% das requisições submetidas ao sistema.

É importante descrever um termo utilizado no decorrer da análise dos gráficos, a degradação do sistema. Um sistema começa a se degradar quando há uma perda superior a 5% na satisfação de seus usuários devido às janelas de operações. Ou seja, de acordo com a Equação 4.6, compara-se a satisfação dos usuários que utilizam a janela de maior tamanho dos experimentos (250), com a satisfação dos usuários que utilizam janela de tamanhos menores (10, 25, 50, 75, 100, 150). Se há diferença da satisfação maior que 5%, significa que houve uma degradação.

$$S(250) - S(wS_u) > 5\% \implies \text{Degradação} \quad (4.6)$$

4.5.1 Satisfação Contratual da Classe A

Os gráficos apresentados nas Figuras 4.2, 4.4, 4.5 mostram valores que dizem respeito à satisfação dos usuários da classe A.

No primeiro gráfico, apresentado na Figura 4.2, é feito um comparativo entre os variados tamanhos de janela. Nesse cenário há poucas requisições de classe A, ou seja, 10% das requisições submetidas ao sistema estão divididas entre os usuários da classe A. Observa-se que não há influências das variações contratuais para qualquer tamanho de janela, isso acontece, pois, a satisfação média (referente a satisfação das 30 simulações) de qualquer uma das variações contratuais avaliadas, de uma determinada janela, está dentro dos intervalos de confiança das demais satisfações médias. Constata-se também, que quanto menor o tamanho da janela, menor é a satisfação contratual. Todavia, o sistema só começa a se degradar considerando janelas menores que 50.

Para explicar o fenômeno da degradação necessitou-se analisar outro tipo de gráfico, que mostra o comportamento da média durante todo o tempo de serviço prestado a um certo usuário. O gráfico representado pela Figura 4.3 apresenta nos seus eixos, abscissa e ordenada, o tempo de liberação das requisições e os tempos médios de resposta, respectivamente. Representando um cenário de baixa carga e baixa variação contratual, ressalta-se que quanto menor o tamanho da janela, menor é a satisfação dos usuários, devido a grande variação dos tempos médios de resposta, e muitas vezes ultrapassando o limite contratado.

O segundo gráfico apresentado pela Figura 4.4 mostra um cenário onde 50% das req-

Figura 4.2: Satisfação dos usuários da classe A - Cenário: 10% requisições classe A.

Figura 4.3: Degradação do Sistema.

uisições do sistema estão divididas entre os usuários de classe A, ou seja, a proporção de requisições são equivalentes na classe A e B. Mesmo com uma carga maior que no cenário anterior, ou seja, mesmo que haja mais requisições na classe mais exigente, é possível obter uma satisfação admissível, considerando os 5% de tolerância, para o usuário com a janela de tamanho 50.

Além disso, nota-se que aumentando a proporção de requisições e diminuindo o tamanho da janela, as variações contratuais tendem a influenciar no comportamento do escalonador. Considera-se que a relação da variação contratual para classe de serviço A é

propiciar menores tempos médios contratados, o que torna mais difícil para o escalonador atender às exigências desses usuários.

Figura 4.4: Satisfação dos usuários da classe A - Cenário: 50% requisições classe A.

Quando a maior parte das solicitações está sendo feita por usuários da classe mais exigente, isto é, as solicitações equivalem a 90% das requisições submetidas ao sistema, como apresentado no gráfico da Figura 4.5, nota-se que o aumento de carga é o principal fator responsável pela influência da variação contratual no escalonador.

As janelas por sua vez, deterioram a satisfação quando em tamanhos menores. No entanto, é possível garantir uma satisfação contratual mantendo as ordens das exigências, para as variações contratuais de 5% e 10%, com janela de tamanho 75. Já quando a exigência da classe A imposta ao sistema é de 20% abaixo da média em relação a *FIFO*, é possível, garantir a satisfação com janela de tamanho 150.

Com 30% de variação contratual, observa-se que não é possível atingir uma satisfação de 100% para o usuário. Além disso, pode-se observar o quanto a dispersão das satisfações originadas das 30 simulações, aumenta quando a janela aumenta de tamanho. Esse cenário também pode ser observado no gráfico da Figura 4.6, o qual mostra o comportamento da média de tempo de resposta para a janela de tamanho 250, de um dado usuário. Nota-se que a média tende a se aproximar da linha contratada. Se aumentasse ainda mais o tamanho da janela, por exemplo, 1000, a média continuaria com esse comportamento, porém, com menos variação. Se uma determinada semente for substancialmente melhor que uma outra semente aleatória, ou seja, a aleatoriedade resultasse em tempos médios um pouco mais abaixo do contrato, os resultados obtidos pela satisfação contratual poderiam variar muito. Para solucionar o problema, Casagrande *et al.* (2007) optou por uma tolerância de 3% acima do valor contratual. Entretanto, este trabalho não aborda essa

Figura 4.5: Satisfação dos usuários da classe A - Cenário: 90% requisições classe A.

tolerância.

Figura 4.6: Comportamento do escalonador.

4.5.2 Satisfação Contratual da Classe B

Os gráficos apresentados nas Figuras 4.7, 4.8 e 4.9 mostram a satisfação contratual da classe B.

No primeiro gráfico, apresentando na Figura 4.7, constitui de cenários em que 10% das requisições do sistema são da classe de serviço B; e nota-se a influência da classe de

serviço A, que, nesse caso, possui 90% das requisições do sistema. O princípio da política *EBS* é sempre tentar manter os tempos médios de resposta efetivos dos usuários da classe A abaixo do limiar contratado, mesmo que isso possa afetar os tempos médios de resposta dos usuários da classe de serviço B, ou seja, como esses cenários possuem alta carga na classe mais exigente, consequentemente afeta a classe B.

O gráfico também mostra que para uma variação contratual menor, como 5% ou 10%, é possível atingir uma satisfação contratual desejável com uma janela de tamanho 50. Entretanto, o desempenho tende a diminuir quando a variação contratual é maior. Nesse caso, janelas maiores são solicitadas tal que o desempenho não se degrade, como é o caso quando a variação contratual é de 20%, que atinge o limite máximo de degradação com janela de tamanho 100. Já com variação contratual de 30% não se pode determinar janelas, pois, o sistema está sobrecarregado de requisições de classe A e com exigência muito alta, ou seja, tempo médio de contrato 30% menor que a média da *FIFO*.

Figura 4.7: Satisfação dos usuários da classe B - Cenário: 10% requisições classe B.

No gráfico apresentado na Figura 4.8, em a porcentagem de requisições na classe é de 50% para cada classe B, observa-se que não há influência da variação contratual no escalonamento. Isso implica na possibilidade de garantir a mesma janela, sem degradação da satisfação, para qualquer tipo de exigência dos experimentos da classe B. Logo, a janela de tamanho 50 possui as características necessárias para um ótimo desempenho associado à satisfação contratual.

Com 90% das requisições na classe B, como mostra o gráfico apresentado pela Figura 4.9, a degradação do sistema só começa com a utilização de janela menor que 50, exceto quando a variação contratual é de 30%. Nesse caso, é possível garantir uma perda inferior a 5% com janela menor que 25. Esses cenários estão habilitados a obter melhores resultados,

Figura 4.8: Satisfação dos usuários da classe B - Cenário: 50% requisições classe B.

devido uma menor porcentagem de requisições da classe mais exigente.

Figura 4.9: Satisfação dos usuários da classe B - Cenário: 90% requisições classe B.

4.5.3 Tempos de Resposta abaixo da Média Contratada

A satisfação de um dado usuário mostra o percentual de vezes em que a média real oferecida ficou dentro da faixa de tolerância especificada em cada contrato de serviço. Entretanto, a satisfação pode se manter alta, mas os valores individuais de tempos de resposta variam muito em relação à média efetiva do usuário. Portanto, é feito uma

análise quanto ao número de tempos de resposta que ficaram abaixo da média contratada. Dessa forma, pode-se avaliar melhor a qualidade de atendimento.

Classe A

Os gráficos apresentados nas Figuras 4.10, 4.11 e 4.12 mostram a porcentagem de requisições que tiveram seus tempos de resposta abaixo da média contratada ($R_{tr\downarrow}$).

Como nos gráficos que expõem a satisfação dos usuários, esses também apresentam no eixo abscissa os tamanhos da janela e no eixo da ordenada, a porcentagem de $R_{tr\downarrow}$. É importante ressaltar que o *warm up* ou o período de aquecimento é considerado.

O primeiro gráfico, apresentado pela Figura 4.10, mostra um cenário em 10% das requisições do sistema são de propriedade dos usuários da classe de serviço A. Já o segundo gráfico, apresentado na Figura 4.11, mostra um cenário que possui 50% de requisições do sistema na classe A, e por fim, o terceiro gráfico, apresentado na Figura 4.12, mostra um cenário onde 90% das requisições submetidas ao sistema são dos usuários da classe A.

Todos esses gráficos apresentam padrões. Cada padrão refere-se às três dimensões dos experimentos. Para a primeira dimensão, observa-se que quanto maior a proporção de requisições na classe, menor é a porcentagem de $R_{tr\downarrow}$. Essa perda é agravada quando há aumento da variação contratual. Como sendo a segunda dimensão dos experimentos, a variação dos contratos estipula a exigência da classe A, ou seja, se ela for alta, consequentemente a média contratada é baixa. Sendo assim, haverá mais perda de $R_{tr\downarrow}$, pois, o objetivo da política não é atender as requisições individuais. Esse comportamento é inválido quando há poucas requisições na classe (gráfico da Figura 4.10) e variação contratual baixa (5, 10 e 20%), pois, o intervalo de confiança acusa a média dos resultados das sementes aleatórias dentro deles. E finalmente, o padrão encontrado na terceira dimensão dos experimentos é a tendência de diminuição da porcentagem de $R_{tr\downarrow}$ à medida que a janela diminui de tamanho.

Para averiguar as perdas em valores absolutos, visto que os cenários atribuídos a janelas menores (como tamanho de 10) possuem uma menor porcentagem de $R_{tr\downarrow}$ e os cenários atribuídos a janelas maiores (como tamanho 250) possuem uma maior porcentagem de $R_{tr\downarrow}$, compara-se a porcentagem de $R_{tr\downarrow}$ da maior janela com a porcentagem de $R_{tr\downarrow}$ da menor janela (Equação 4.7).

$$\Delta = R_{tr\downarrow}(250) - R_{tr\downarrow}(10) \quad (4.7)$$

A Tabela 4.1 mostra o Δ para cada cenário referente a cada variação contratual e proporção de requisições na classe.

Sucintamente, a perda média de $R_{tr\downarrow}$ (Equação 4.8) é de aproximadamente 5,69% com um desvio padrão próximo a 0,81%.

Figura 4.10: *Tempos de Resposta abaixo da média contratada - Cenário 10% requisições classe A.*

Figura 4.11: *Tempos de Resposta abaixo da média contratada - Cenário 50% requisições classe A.*

Figura 4.12: *Tempos de Resposta abaixo da média contratada - Cenário 90% requisições classe A.*

$$\mu = \frac{\sum_{p=\{10,50,90\}} \left(\sum_{vc=\{5,10,20,30\}} \Delta_{p,v} \right)}{12} \quad (4.8)$$

Classe B

Os gráficos apresentados nas Figuras 4.13, 4.14 e 4.15 possuem as mesmas características dos gráficos da classe A, em termos de estruturação, porém, esses referenciam as requisições dos usuários da classe B.

Comparando os cenários de classe A com os de classe B, nota-se que quanto maior a proporção de requisições na classe B, maior também é a porcentagem de $R_{tr\downarrow}$. Pois, as

Cenário	Δ
P_10;V.Cont_5	5,58
P_10;V.Cont_10	5,38
P_10;V.Cont_20	4,94
P_10;V.Cont_30	4,28
P_50;V.Cont_5	5,99
P_50;V.Cont_10	6,05
P_50;V.Cont_20	6,03
P_50;V.Cont_30	5,94
P_90;V.Cont_5	6,19
P_90;V.Cont_10	6,57
P_90;V.Cont_20	7,05
P_90;V.Cont_30	4,29

Tabela 4.1: Δ referente a cada variação contratual e proporção de requisições na classe A.

requisições de classe B possuem um tempo médio contratado maior que as requisições de classe A.

No primeiro gráfico, em há poucas requisições na classe B (Figura 4.13), observa-se, à medida que a janela aumenta, que há menores porcentagens de $R_{tr\downarrow}$ quando a variação contratual é de 30%. Isso ocorre, pois, apesar dos contratos dos usuários de classe B serem mais tolerantes, o que poderia significar um aumento de $R_{tr\downarrow}$, verifica-se que o aumento da exigência da classe A é potencializada pela sua maior proporção no sistema. Com isso, a probabilidade das requisições de classe B serem preteridas aumenta, causando a diminuição da qualidade de serviço individual das requisições.

Para os demais gráficos (Figuras 4.14 e 4.15) observa-se que há um aumento na porcentagem de $R_{tr\downarrow}$. Comportamento natural, visto que os contratos de classe B possuem um maior tempo médio de resposta contratado potencializado pela proporção de requisições em uma classe inferior.

Em relação ao valores absolutos, a perda média de $R_{tr\downarrow}$ denotado por μ (Equação 4.8) é de aproximadamente 4,81% com um desvio padrão próximo a 1,31%. A Tabela 4.2 mostra o Δ de cada variante do cenário.

À primeira vista, assinar um contrato do tipo B parece ser a melhor opção para o usuário, pois, oferecem uma maior porcentagem de $R_{tr\downarrow}$. Entretanto, a média contratada pelos usuários de classe B é mais tolerante que a dos usuários de classe A.

4.5.4 Resumo

Os resultados mostraram que as janelas menores degradam a satisfação contratual para qualquer classe de serviço. A degradação é enfatizada quando a variação contratual aumenta para a classe de serviço A. No entanto, para a classe de serviço B há uma

Figura 4.13: *Tempos de Resposta abaixo da média contratada - Cenário 10% requisições classe B.*

Figura 4.14: *Tempos de Resposta abaixo da média contratada - Cenário 50% requisições classe B.*

Figura 4.15: *Tempos de Resposta abaixo da média contratada - Cenário 90% requisições classe B.*

tendência a maiores satisfações quando o tempo médio contratado aumenta (ou a variação contratual aumenta), exceto quando a proporção de requisições na classe A é alta seguido da exigência descrita pela variação contratual. Nesse caso, a classe B é prejudicada, aumentando o tempo médio de resposta efetivo, para tentar manter o tempo médio de resposta efetivo dos usuários de classe A abaixo do limiar contratado.

4.6 Considerações Finais

Neste capítulo foi descrito o modelo de contrato com janelas deslizantes. Na Seção 4.1 discutiu-se as modificações necessárias na EBS para comportar as janelas ou seções dos

Cenário	Δ
P_10;V.Cont_5	5,74
P_10;V.Cont_10	5,70
P_10;V.Cont_20	5,40
P_10;V.Cont_30	1,15
P_50;V.Cont_5	5,86
P_50;V.Cont_10	5,76
P_50;V.Cont_20	5,49
P_50;V.Cont_30	4,97
P_90;V.Cont_5	5,31
P_90;V.Cont_10	4,87
P_90;V.Cont_20	4,09
P_90;V.Cont_30	3,43

Tabela 4.2: Δ referente a cada variação contratual e proporção de requisições na classe B.

usuários. Para melhor compreensão foi também demonstrado um exemplo do funcionamento do modelo contratual.

Em seguida, na Seção 4.2 apresentou-se como técnica de modelagem e simulação de sistemas, os materiais e métodos adotados para estudar a eficácia da política sob janelas de operações. Também foram apresentados a composição geral do modelo, a ferramenta de simulação empregada para validação do modelo e da política, discutindo o motivo de sua escolha; e por fim, o parâmetro de serviço utilizado no controle da qualidade de serviço, bem como a medida de avaliação, mais detalhada na seção 4.3.

Com os métodos bem definidos, pode-se fazer uma melhor análise dos resultados. Para isso, definiu-se o planejamento de experimentos na Seção 4.4.

Após delinear os experimentos, foram descritos os resultados na Seção 4.5. Para organizar a análise, apresentaram-se os dados obtidos em forma de gráficos, delineando a satisfação contratual sob diferentes variações dos parâmetros do sistema, como proporção de requisições, variações contratuais e tamanhos de janela. Apresentou-se na Subseção 4.5.1 os resultados obtidos para os usuários da classe superior. Enquanto que na Subseção 4.5.2 apresentou-se os resultados obtidos para os usuários de uma classe inferior. Não obstante, as seções mostram também a variabilidade dos resultados. Finalmente, a Subseção 4.5.3 apresentou as análises quanto ao tempos de respostas individuais das requisições referentes as suas classes de serviço.

O próximo capítulo investiga o atraso das requisições individuais em relação ao contrato. Também apresenta-se métodos para garantir menores dispersões dos atrasos e os resultados obtidos através desses métodos.

Reduzindo o Desvio Padrão dos Atrasos

Neste capítulo são apresentadas as abordagens e procedimentos utilizados para o desenvolvimento da segunda parte da pesquisa. Nele são introduzidos duas políticas de escalonamento planejadas com o objetivo de reduzir a dispersão dos atrasos das requisições. Discutidas nas Seções 5.1 e 5.2, as políticas são avaliadas segundo a média dos atrasos, o desvio padrão dos atrasos, como mostra a Seção 5.3, e a satisfação contratual definida no capítulo anterior, na Seção 4.3.1. Em seguida, na Seção 5.4 são apresentadas alterações nos níveis dos fatores dos experimentos pelo grande volume de informações a serem analisados. Os resultados são discutidos na Seção 5.5, abordando a influência de cada fator e o desempenho comparativo das abordagens.

5.1 Política aEBS

A política de escalonamento *EBS* é originada por três abordagens. A primeira delas determina que as requisições sejam classificadas como mais urgentes as que possuírem menores *deadlines*, porém, podem existir algumas com os mesmos valores de *deadlines*. Nesse caso, apesar de apresentar a mesma urgência, as requisições podem impor pesos distintos. Baseado no princípio da política *SJF* (*Shortest Job First*), a segunda abordagem oferece o menor tempo médio de resposta. A terceira abordagem é caracterizada pelos valores negativos de *deadline*, determinando a prioridade como sendo inversamente proporcional ao custo esperado de processamento da requisição.

A nova abordagem nomeada como *aEBS* (*Aged Exigency Based Scheduling*) interage com outro fator, o tempo de espera da requisição em fila. É surpresa que esse já faz parte

do escalonamento das requisições, como mostra a Equação 5.1 que define o *deadline* ou o tempo máximo que a requisição pode esperar na fila. Entretanto, o tempo de espera pode ter seu impacto absorvido em certas circunstâncias. Por exemplo, com a utilização dos *SLAs* em janelas deslizantes, percebe-se que o componente do *deadline* $\Omega_{cu} \cdot wS$ é um valor fixo (Equação 5.1) depois que a janela do usuário estiver completa por tempos de resposta, ou seja, quaisquer tempos de resposta que compõem o *shift*¹ da janela não altera o valor desse componente. E, se ainda, a janela contiver valores baixos de tempos de resposta, esse componente do *deadline* certamente será um valor baixo com relação ao do contrato.

$$D_j = \left((\Omega_{cu} \cdot wS) - (\Omega_u \cdot wS) - T_{w_j} \right) \quad (5.1)$$

A Figura 5.1 apresenta um exemplo de janela com tempos de resposta baixos. Observa-se o quão alto é o contrato ($\Omega_{cu} \cdot wS$) em relação a média efetiva do usuário ($\Omega_u \cdot wS$). Logo, o tempo de espera da requisição que está na fila poderá não influenciar tanto quanto deveria. O quanto uma requisição ficará aguardando na fila até ser escalonada, pode, muitas vezes, assumir valores desastrosos para o usuário proprietário dela, e um valor de tempo de espera alto é indesejável para qualquer tipo de classe de serviço.

Janela	2 2 2 2 2 2 2 2 2 2 2 2
Contrato	9
$\Omega_{cu} \cdot wS$	90
$\Omega_u \cdot wS$	20

Figura 5.1: *Exemplo de janela com baixos tempos de resposta.*

Em Casagrande *et al.* (2007), a priorização era dada pelo equilíbrio do tempo de processamento e o *deadline* (Equação 3.4). Nessa abordagem, as requisições mais urgentes e que possuem menores tempo de processamento são escalonadas primeiro. No entanto, a nova abordagem, nomeada *aEBS* (*Aged Exigency Based Scheduling*), referencia também as requisições mais urgentes, como aquelas que ficaram por muito tempo na fila de espera. A prioridade obtida pela Equação 5.2 determina que o produto do custo computacional com o *deadline* seja inversamente proporcional ao tempo de espera, se o *deadline* da requisição for positivo. Caso contrário, o tempo de espera é diretamente proporcional ao *deadline* e inversamente proporcional ao tempo de processamento da requisição. O resultado esperado é a diminuição da prioridade conforme o tempo de espera da requisição. Nos casos onde existem valores entre 0 e 1 de tempo de espera originados de requisições que acabaram de entrar no sistema, são somados a seus valores 1 u.t, pois, o oposto,

¹Representa a troca entre o antigo tempo de resposta pelo novo tempo de resposta na janela.

aumentaria a prioridade da requisição.

$$P_j = \begin{cases} (D_j \cdot T_{p_j}) \cdot \frac{1}{T_{w_j} + 1} & \text{se } D_j \geq 0 \\ \frac{D_j}{T_{p_j}} \cdot (T_{w_j} + 1) & \text{se } D_j < 0 \end{cases} \quad (5.2)$$

5.2 Políticas σEBS

Numa tentativa de diminuir os atrasos das requisições, a política $aEBS$ prioriza-as pelo equilíbrio entre o *deadline*, tempo de processamento e o tempo de espera. Esse método pode abordar todas as classes de requisições, ou seja, aquelas requisições que acabaram de entrar no sistema e possuem um tempo de espera próximo a 0.

A abordagem σEBS difere das outras políticas ao definir como critério de avaliação dos atrasos, a dispersão dos mesmos. O atraso, definido na Equação 5.3, é a diferença do tempo de resposta da requisição recém atendida pelo servidor $((time() - timeStamp_j))$ e o tempo médio contratado pelo usuário u , Ω_{c_u} .

$$\rho_j = (time() - timeStamp_j) - \Omega_{c_u} \quad (5.3)$$

Se o valor de ρ_j for negativo, significa que a requisição recém atendida teve seu tempo de resposta abaixo da média contratada pelo usuário, portanto, não há nenhuma observação sobre a requisição. Caso o valor de ρ_j for positivo, significa que a requisição recém atendida teve seu tempo de resposta acima do valor contratado pelo usuário. Sendo assim, ativa-se o método que indicará qual a dispersão dos atrasos dessa requisição com relação as demais do determinado usuário efetivo no sistema. A Equação 5.4 define o desvio padrão das requisições atrasadas do usuário (σ_u) a cada ciclo do escalonador. Para obtê-lo são indispensáveis a variância anterior do usuário ($\sigma_u^{2'}$) e a média dos atrasos das requisições do usuário ($\bar{\rho}_u$). Necessitou-se essa reformulação da variância para garantir que as equações não degradem o desempenho do sistema quanto ao escalonamento. No Apêndice A está o desenvolvimento da variância.

$$\sigma_u = \sqrt{\sigma_u^2}$$

$$\sigma_u^2 = R_u \cdot \left(\frac{\sigma_u^{2'}}{R_u + 1} + \left(\frac{\bar{\rho}_u - \rho_j}{R_u + 1} \right)^2 \right) \quad (5.4)$$

A abordagem σEBS incide no *deadline*, pois, como já mencionado anteriormente, esse possui fragmento fixo com o uso de janelas deslizantes, o que pode acarretar em um prazo máximo de espera muito alto para determinadas requisições. Desse modo, o método atribui ao *deadline* um tempo de espera superior ao real, baseado no comportamento da

dispersão dos atrasos das requisições do usuário σ_u (Equação 5.5). Esse método tem a finalidade de aumentar a prioridade das requisições já atrasadas. No entanto, desvios-padrão entre 0 e 1 causam o oposto do esperado. Sendo assim, necessitou-se adotar 1 u.t a mais que o desvio padrão ($\sigma_u + 1$).

$$D_j = \left((\Omega_{c_u} \cdot wS) - (\Omega_u \cdot wS) - \left((\sigma_u + 1) \cdot T_{w_j} \right) \right) \quad (5.5)$$

5.3 Medidas para Avaliação

Uma das medidas empregada para validação da política relaciona o número de vezes que a média contratual foi cumprida, pelo número total de requisições submetidas ao sistema, referenciada anteriormente pela Equação 4.3.

As demais medidas avaliam as políticas com respeito a dispersão dos atrasos. Para isso, foram utilizados o desvio padrão dos atrasos referenciado na Equação 5.4 e a média dos atrasos denotada na Equação 5.6, pois, qualquer esforço para reduzir o desvio padrão dos atrasos é inválido se a média dos atrasos das políticas desenvolvidas estiver acima da média dos atrasos obtida pela política *EBS*. Essa média é referenciada apenas pelas requisições que possuem atraso, ou seja, é obtida através da média entre o antigo tempo médio do atraso do usuário $\bar{\rho}_u'$ e o atraso da requisição recém atendida pelo servidor, ρ_j . O valor de R_p especifica o número de requisições atrasadas do usuário u .

$$\bar{\rho}_u = \frac{\bar{\rho}_u' \cdot R_{\rho_u} + \rho_j}{R_{\rho_u} + 1} \quad (5.6)$$

5.4 Planejamento de Experimentos

Os experimentos são similares aos caracterizados no capítulo anterior, ou seja, o mesmo número de classes (A e B), mesmo número de usuários para cada classe (totalizando 20 usuários). Para geração de carga sintética, as mesmas características são usadas: distribuições exponenciais com média 4 u.t para definir o intervalo entre chegadas e com média 3 u.t para definir o tempo de execução da tarefa. Dessa forma, o sistema possuiu uma taxa de utilização de 75%.

Os cenários investigados nesse experimento enfocam as mesmas três dimensões: a proporção de requisições pertencentes a cada classe, a diferença entre o tempo médio de resposta contratado das classes de serviço e o tamanho da janela.

Para o primeiro parâmetro, especificou-se duas situações, uma com um sistema com mais requisições de baixa exigência (25%A e 75%B) e outra com um sistema com mais requisições de alta exigência (75%A e 25%B).

Os contratos também relacionam uma porcentagem abaixo do tempo médio de resposta

da política *FIFO* para os usuários de classe A, e acima para os contratos dos usuários de classe B. No entanto, as porcentagens que determinam a discrepâncias² entre as duas classes são de 10%, 20% e 30%.

O terceiro parâmetro especifica os tamanhos da janela, os quais são caracterizados por uma porcentagem X do número de requisições obtidos por cada usuário de acordo com a menor proporção de requisições dos experimentos (25%). O parâmetro X foi variado em 3, 4, 6, 10 e 20 resultando em janela de tamanhos 75, 100, 150, 250 e 500. A Equação 5.7 mostra como conseguiu-se os tamanhos da janela.

$$wS = \left(\frac{\left(\frac{\left(\frac{minorProportion(proportion[]).requestCount_System}{100} \right)}{class_numberUsers} \right)}{100} \right) \cdot X \quad (5.7)$$

Onde, *minorProportion* é do tipo função que obtém a menor proporção de requisições dentre as investigadas/experimentos, *Proportion*. E, *requestCount_System* é o número de requisições do sistema, e *class_numberUsers* especifica o número de usuários na classe.

O número de requisições no sistema para cada cenário foi definido também como 100.000. Cada cenário foi simulado 30 vezes, com diferentes sementes aleatórias. Os resultados apresentados foram obtidos a partir da média dessas 30 simulações com um intervalo de confiança de 90%.

5.5 Análise dos Resultados

Os gráficos nas figuras dessa seção apresentam os experimentos de simulação que permitem um comparativo entre as políticas *EBS*, *aEBS* e *σEBS* quanto ao desvio padrão dos atrasos. Inicialmente, cada dimensão do experimento é analisada individualmente, a fim de caracterizar a influência dos fatores no comportamento do escalonador. O primeiro deles a ser analisado é o tamanho da janela, em seguida, a variação contratual. Por fim, é feita uma comparação entre as políticas a fim de mostrar qual obteve o melhor desempenho.

5.5.1 Análise quanto ao Tamanho da Janela

Os gráficos dessa subseção são apresentados para averiguar o comportamento das políticas com relação as janelas deslizantes.

²Diferença dos contratos A e B.

Cenário: 25% Classe A e 75% Classe B

O primeiro gráfico, apresentado pela Figura 5.2, mostra um cenário em que 25% das requisições do sistema são da classe de serviço A. Em contrapartida, o gráfico apresentado na Figura 5.3 mostra um cenário em 75% das requisições do sistema pertencem aos usuários de classe B. Esses gráficos mostram no eixo das abscissas, a variação contratual e a política de escalonamento avaliada. O eixo das ordenadas informa o desvio padrão das requisições atrasadas. As legendas ou as colunas mostram os tamanhos de janela.

No primeiro gráfico nota-se que as janelas investigadas não se diferenciam no comportamento do escalonador quanto ao desvio padrão dos atrasos das requisições para a política de escalonamento *EBS*. Estatisticamente, não pode-se afirmar que os desvios-padrão são maiores ou menores comparando-se as janelas com a política *EBS*. O mesmo comportamento pode ser observado com a política de escalonamento *aEBS*. Entretanto, para a política de escalonamento σEBS , quando a variação contratual é de 10%, observa-se que as janelas influenciam no comportamento do escalonador. Nesse cenário, nota-se que quanto maior o tamanho da janela, maior é dispersão dos atrasos. Para as demais variações contratuais, os tamanhos das janelas não se diferenciam com a política σEBS .

Figura 5.2: *Desvio Padrão dos Atrasos - Cenário: 25% requisição classe A.*

Observa-se em linhas gerais, para os cenários de classe B mostrados no gráfico da Figura 5.3, em que a proporção de requisições é de 75%, que há maiores dispersões dos atrasos quando comparado-se ao cenários de classe A. Esse comportamento indica que os contratados mais relaxados são propícios a maiores variações de atrasos. No entanto, não há influência das janelas investigadas sobre o desvio padrão dos atrasos obtidos pela *EBS*. O mesmo acontece para a política *aEBS*. Todavia, para a política de escalonamento σEBS quanto maior a janela, maior o desvio padrão dos atrasos.

Além da variação dos atrasos, outra importante análise é a média dos atrasos com

Figura 5.3: *Desvio Padrão dos Atrasos - Cenário: 75% requisição classe B.*

relação aos tamanhos de janela. Apresentada nos gráficos das Figuras 5.4 e 5.5, o eixo das ordenadas informa a média dos atrasos, enquanto que, o eixo das abscissas informa a variação contratual e a política de escalonamento avaliada. E, as legendas ou as colunas mostram os tamanhos de janela.

Observa-se no gráfico apresentado pela Figura 5.4 - em que 25% das requisições do sistema são pertencentes aos usuários de classe A - que para as política *EBS* e *aEBS*, as médias dos atrasos com relação as janelas não se distinguem. Já a política *σEBS* há uma pequena variação entre os tamanhos de janela, principalmente quando a variação contratual é alta.

Figura 5.4: *Média dos Atrasos - Cenário: 25% requisição classe A.*

Para os cenários de classe B, esse correspondente a 75% das requisições do sistema,

o gráfico apresentado na Figura 5.5 mostra que as médias dos atrasos para a política σEBS aumentam à medida que o tamanho da janela aumenta de tamanho, para os casos onde as variações contratuais são de 10% e 20%. Para as demais políticas, as janelas não influenciam no comportamento do escalonador quanto a média dos atrasos.

Figura 5.5: Média dos Atrasos - Cenário: 75% requisição classe B.

As janelas maiores correspondem a menos atrasos para a política σEBS , ou seja, menos requisições com seus tempos de resposta abaixo da média contratada ($R_{tr\downarrow}$), como mostra os gráficos das Figuras 5.6 e 5.7. Neles, o eixo das abscissas informa a variação contratual e a política de escalonamento avaliada. O eixo das ordenadas informa a porcentagem de $R_{tr\downarrow}$, e as legendas ou as colunas mostram os tamanhos de janela. Para as demais políticas, o número de atrasos, relativo as janelas, não se diferenciam.

Resumidamente, as políticas EBS e $aEBS$ não são influenciadas pelo tamanho da janela. Em contradição, a política σEBS consta influências no desvio padrão dos atrasos, na média de atrasos e no número de requisições abaixo da média contratada.

Cenário: 75% Classe A e 25% Classe B

Com uma maior proporção de requisições na classe mais exigente ou A, tal como 75% das requisições do sistema, observa-se no gráfico apresentado pela Figura 5.8, que o desvio padrão dos atrasos, com a política EBS , não se diferenciam para qualquer uma das janelas dos experimentos. Para a política $aEBS$ há um tendência a diminuir a variação dos atrasos com janelas maiores, entretanto, a comparação entre os tamanhos das janelas são indistinguíveis por causa dos intervalos de confiança. E, com a política σEBS , como mencionado anteriormente, à medida que a janela aumenta, a dispersão dos atrasos aumenta, exceto quando a variação contratual é de 30%. Nesse caso, os desvios-padrão dos atrasos não se distinguem para as janelas investigadas.

Figura 5.6: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe A.*

Figura 5.7: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe B.*

O mesmo comportamento pode ser notado para a classe B, quando essa possui 25% das requisições do sistema, como mostra o gráfico da Figura 5.9. Entretanto, para todas as variações contratuais, a política σEBS aumenta o desvio padrão dos atrasos à medida que a janela aumenta de tamanho.

Com relação a média dos atrasos, pode-se observar no gráfico apresentado pela Figura 5.10, que as políticas EBS e $aEBS$ tende a diminuir a média dos atrasos, à medida que a janela aumenta de tamanho. Porém, para menores variações contratuais, os intervalos de confiança se sobrepõem à média das simulações mantendo-as relativamente iguais, principalmente, quando a discrepância entre as classes é de 10%. Já a política σEBS ,

Figura 5.8: *Desvio Padrão dos Atrasos - Cenário: 75% requisição classe A.*

Figura 5.9: *Desvio Padrão dos Atrasos - Cenário: 25% requisição classe B.*

nota-se que janelas menores conseguem melhores médias de atrasos. Entretanto, com 30% de variação contratual, as médias dos atrasos não se diferenciam.

Para os cenários de classe B, esse constituído de 25% das requisições do sistema, observa-se no gráfico da Figura 5.11, que as janelas tendem a influenciar no comportamento do escalonador para as políticas *aEBS* e *σEBS* quanto a média dos atrasos. Entretanto, a *EBS* mantém o mesmo comportamento para qualquer uma das janelas.

Os gráficos das Figuras 5.12 e 5.13 mostram a porcentagem de requisições com os tempos de resposta abaixo da média contratada ($R_{tr\downarrow}$) no eixo das ordenadas, para a classe A e B, respectivamente. Observa-se que as políticas *EBS* e *aEBS* mantém o mesmo número de $R_{tr\downarrow}$, com relação as janelas. Com a política *σEBS*, à medida que a janela

Figura 5.10: Média dos Atrasos - Cenário: 75% requisição classe A.

Figura 5.11: Média dos Atrasos - Cenário: 25% requisição classe B.

aumenta de tamanho, a porcentagem de $R_{tr\downarrow}$ também aumenta.

Resumidamente, observa-se que quando há maiores proporções de requisições na classe A, os tamanhos das janelas influenciam mais na política *aEBS*. Entretanto, para a política *EBS* a maioria dos cenários investigados não são influenciados pelas janelas. Em contrapartida, a política *σEBS* está diretamente relacionada aos tamanhos das janelas. É conclusivo para a maioria dos cenários, que com a *σEBS* à medida que a janela aumenta de tamanho, há maiores variação dos atrasos, maior média de atrasos e maior porcentagem de $R_{tr\downarrow}$.

Figura 5.12: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe A.*

Figura 5.13: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe B.*

5.5.2 Análise quanto a Variação Contratual

Nessa subseção é feito um comparativo entre as variações contratuais definidas nos experimentos. Os gráficos apresentados mostram qual o comportamento das políticas com relação as discrepâncias dos contratos das classes A e B. A análise é relativa a janela de tamanho 250, visto que, para a política *EBS* não há influência das janelas no comportamento de escalonador quanto a variação dos atrasos, média dos atrasos e porcentagem de $R_{tr\downarrow}$.

Cenário: 25% Classe A e 75% Classe B

O primeiro gráfico apresentado pela Figura 5.14 mostra cenários que representam 25% das requisições no sistema para a classe A. O eixo das ordenadas mostra o desvio padrão dos atrasos, enquanto que, o eixo das abscissas mostra a variação contratual a ser analisada das políticas de escalonamento referenciadas na legenda.

Observa-se no gráfico da Figura 5.14 que quanto maior a variação contratual, para qualquer uma das políticas, menor é desvio padrão dos atrasos. Ou seja, quanto maior a exigência dos contratos, mais restritos são os *deadlines*, e conseqüentemente são escalonadas mais requisições da classe A do que as de classe B.

Figura 5.14: *Desvio Padrão dos Atrasos - Cenário: 25% requisição classe A.*

Para os cenários de classe B, como pode-se observar no gráfico da Figura 5.15, as variações contratuais não influenciam no comportamento do escalonador para nenhuma das políticas. Entretanto, há uma tendência a maiores desvios-padrão dos atrasos quando aumenta-se a discrepância entre as classes. Como observado anteriormente, contratos mais relaxados possuem maiores *deadlines*, em consequência, as requisições de classe B podem esperar por mais tempo em fila, que pode resultar em possíveis maiores dispersões dos atrasos.

Observa-se no gráfico apresentado na Figura 5.16 que as médias dos atrasos diminuem à medida que a variação contratual aumenta para qualquer uma das políticas. Uma das explicações para esse comportamento, é pelo fato que com menores valores contratuais (como 30% de variação contratual), mais restrito são os *deadlines*, de forma a manter a média abaixo do valor contratado; conseqüentemente mais requisições de classe A serão escalonadas do que as de B, e por fim, menores serão os atrasos das requisições de classe A.

Por outro lado, para os cenários de classe B, que representam 75% das requisições do sistema, apresentado pela Figura 5.17, as médias aumentam à medida que a exigência na

Figura 5.15: *Desvio Padrão dos Atrasos - Cenário: 75% requisição classe B.*

Figura 5.16: *Média dos Atrasos - Cenário: 25% requisição classe A.*

classe A aumenta.

Resumidamente, as políticas possuem um mesmo comportamento em relação as variações contratuais. Com menor proporção de requisições na classe A, quanto maior a variação contratual, menor são os desvios e as médias dos atrasos, pois, mais requisições de classe A são escalonadas comparado a uma variação contratual menor. Por outro lado, com maior proporção de requisições na classe B, as políticas tendem a obter maiores desvios-padrão e maiores médias de atrasos a medida que variação contratual aumenta.

Cenário: 75% Classe A e 25% Classe B

Com mais requisições na classe A, 75% das requisições do sistema, como mostra o gráfico da Figura 5.8, nota-se que as políticas tendem ao mesmo comportamento quando comparado-se as variações contratuais. Observa-se que quanto maior a discrepância entre

Figura 5.17: Média dos Atrasos - Cenário: 75% requisição classe B.

as classes, menores são as variações dos atrasos. Como já mencionado, quanto menor o valor contratado definido por variações contratuais de 30%, por exemplo, mais rígidos são os *deadlines*, e conseqüentemente mais requisições de classe A são escalonadas para manter os tempos médios de resposta abaixo do contrato.

Figura 5.18: Desvio Padrão dos Atrasos - Cenário: 75% requisição classe A.

Como os contratos que representam a classe B são maiores ou mais relaxados, o comportamento oposto da classe A é esperado. Visto no gráfico da Figura 5.19, observa-se uma tendência (os intervalos de confiança sobrepõem às médias das simulações) a maiores desvios-padrão dos atrasos para a política *EBS* quando aumenta a variação contratual, entretanto, para *aEBS* é ressaltado esse comportamento. Já a política *σEBS*, não há efeito das variações contratuais.

Com respeito as médias dos atrasos, observado no gráfico da Figura 5.20, para o cenário de classe A, nota-se que quanto maior a variação contratual, menor é a média dos atrasos.

Figura 5.19: *Desvio Padrão dos Atrasos - Cenário: 25% requisição classe B.*

Esse comportamento é demonstrado para qualquer uma das políticas investigadas.

Figura 5.20: *Média dos Atrasos - Cenário: 75% requisição classe A.*

Para os cenários de classe B, mostrado no gráfico da Figura 5.21, nota-se que quanto maior a variação contratual, maior é a média dos atrasos para as políticas *EBS* e *aEBS*. No entanto, as médias dos atrasos se mantêm para a política *σEBS*.

Sucintamente, as políticas possuem em comum, menores desvios-padrão e médias de atrasos para a classe de serviço A, à medida que a variação contratual aumenta. Esse comportamento, deve-se ao fato de que quanto maior a variação contratual, mais serão privilegiadas as requisições classe A por causa dos seus contratos restritos. Entretanto, para a classe de serviço B, as políticas *EBS* e *aEBS* tendem a obter maiores desvios-padrão à medida que a variação contratual aumenta. Diferentemente da política *σEBS* que provê o mesmo comportamento para qualquer uma das variações contratuais.

Figura 5.21: Média dos Atrasos - Cenário: 25% requisição classe B.

5.5.3 Análise quanto as Políticas de Escalonamento

Essa subseção irá mostrar quais políticas produzem melhores desempenhos com relação a satisfação contratual, a média dos atrasos e a variação dos mesmos.

Satisfação Contratual

Os gráficos apresentados nas Figuras 5.22 até 5.25 mostram a satisfação do usuário com respeito ao número de vezes em que o tempo médio de resposta ficou abaixo do contrato. Esse é apresentado no eixo das ordenadas, enquanto, no eixo das abscissas são apresentadas as políticas de escalonamento avaliadas. As legendas identificam cada uma das variações contratuais.

O primeiro gráfico, apresentado pela Figura 5.22, delinea um cenário em que a proporção de requisições na classe de serviço A representa 25% do total de requisições do sistema. Observa-se que para qualquer política de escalonamento a satisfação contratual é cumprida com 100% de vezes em que o *deadline* ficou abaixo da média contratada.

Para o cenário representado por requisições de classe B, em que 75% das requisições são de usuários com um contrato inferior, como apresenta o gráfico da Figura 5.23, observa-se também que as políticas conseguem uma satisfação de 100% para qualquer variação contratual.

Com 75% de requisições na classe prioritária, nota-se no gráfico da Figura 5.24 que a política *EBS* atinge a satisfação de 100% para qualquer variação contratual. Já a política *aEBS* consegue atingir uma satisfação de 100% abrangendo o intervalo de confiança com variações contratuais de 10 e 20%. Entretanto, com 30% de variação contratual, a satisfação contratual cai para 80%, pois, cenários como esse, além de saturados por requisições prioritárias com valores baixo de média contratada, ainda ponderam os tempos de espera

Figura 5.22: *Satisfação Contratual - Cenário: 25% requisição classe A.*

Figura 5.23: *Satisfação Contratual - Cenário: 75% requisição classe B.*

em fila das requisições. Para a política σEBS , com 10% de variação contratual é possível atingir uma satisfação de 100%. Entretanto, quanto maior a variação contratual maior a degradação da satisfação.

Com 25% de requisições na classe inferior, como mostra o gráfico da Figura 5.25, observa-se que as políticas EBS e σEBS atingem uma satisfação desejável para qualquer variação contratual, diferentemente da política $aEBS$ que com 30% atinge uma satisfação de aproximadamente 90%. Para as demais variações contratuais a satisfação é de 100%.

Basicamente, apenas em cenários saturados de requisições prioritárias e com variação contratual alta, as políticas $aEBS$ e σEBS não atingem uma satisfação contratual superior a 80%.

Figura 5.24: *Satisfação Contratual - Cenário: 75% requisição classe A.*

Figura 5.25: *Satisfação Contratual - Cenário: 25% requisição classe B.*

Média dos Atrasos

Os gráficos apresentados nas Figuras 5.26 até 5.29 mostram as médias dos atrasos das requisições pertencentes a cada uma das classes de serviço. Qualquer esforço para reduzir os desvios-padrão dos atrasos é inválido se a média for mais alta que a política *EBS*, do qual o trabalho tem como referência comparativa. Os gráficos apresentam nos eixos abscissas e ordenadas, a variação contratual e a média dos atrasos, respectivamente. A legenda mostra as políticas avaliadas.

O primeiro gráfico apresentado na Figura 5.26 mostra um cenário em que 25% das requisições do sistema estão entre os usuários de classe A. Observa-se que as políticas *EBS* e *aEBS* apresentam a mesma média de atrasos, para qualquer variação contratual. Já a política *σ EBS* tende a maiores médias de atrasos, mas qualquer afirmação sobre o

seu desempenho em comparação a *EBS* é inválido, pois, os intervalos dessas políticas se sobrepõem à média das simulações.

Figura 5.26: Média dos Atrasos - Cenário: 25% requisição classe A.

Com mais requisições na classe menos exigente, ou seja 75% das requisições do sistema são de usuários de classe B, mostrado no gráfico da Figura 5.27, nota-se que para qualquer variação contratual a política *aEBS* possui o melhor desempenho em termos de média de atrasos. Já as políticas *EBS* e *σEBS* não se diferenciam.

Figura 5.27: Média dos Atrasos - Cenário: 75% requisição classe B.

No entanto, à medida que a proporção de requisições na classe A aumenta, como mostra o gráfico da Figura 5.10, e à medida que a variação contratual aumenta, a política *EBS* tende a se igualar as médias dos atrasos da *aEBS*. Já a política *σEBS* permanece com o mesmo desempenho que a *EBS*.

Figura 5.28: Média dos Atrasos - Cenário: 75% requisição classe A.

Com 25% de requisições na classe B, como mostra o gráfico da Figura 5.29, observa-se que as políticas *aEBS* e *σ EBS* obtêm menores médias de atrasos que a política *EBS* para qualquer variação contratual.

Figura 5.29: Média dos Atrasos - Cenário: 25% requisição classe B.

Em linhas gerais, deseja-se que as médias permaneçam no mínimo iguais a da política *EBS*. Os resultados mostraram que para alguns cenários, como os de classe B a política *aEBS* obteve uma média menor que a política *EBS*. Para os cenários de classe A, a política *aEBS* obteve desempenho semelhante a *EBS* com possibilidades de melhor desempenho quando a variação contratual é baixa. Para a política *σ EBS*, o desempenho para a maioria dos cenários foi igual a política *EBS*, com possibilidades de melhores resultados para os usuários de classe B, quando em baixa proporção de requisições na classe.

Desvio Padrão dos Atrasos

O gráfico apresentado nas Figuras de 5.30 a 5.33 mostram o desempenho de cada política relacionado ao desvio padrão dos atrasos (eixo das ordenadas). No eixo das abscissas encontram-se as variações contratuais, enquanto que a legenda mostra as políticas avaliadas.

O primeiro gráfico, apresentado pela Figura 5.30 corresponde ao de classe A, em 25% das solicitações são de usuários mais exigentes. Observa-se que a dispersão dos atrasos tende a diminuir quando aumenta a variação contratual para qualquer uma das políticas. No entanto, com 10% e 20% de variação contratual, a política *aEBS* obtém melhor desempenho que a *EBS*, e a σEBS obtém o mesmo desempenho que a *EBS*. Com 30% de variação contratual, as políticas *EBS* e a *aEBS* não se diferenciam, pois, o intervalo de confiança da σEBS sobrepõem o intervalo de confiança da *EBS*.

Figura 5.30: *Desvio Padrão dos Atrasos - Cenário: 25% requisição classe A.*

Com 75% de requisições na classe B, como mostra o gráfico da Figura 5.31, nota-se que as políticas *aEBS* e σEBS conseguem a mesma variação dos atrasos para qualquer variação contratual. A política *EBS* tem uma dispersão de atrasos consideravelmente maior que as políticas desenvolvidas.

O mesmo comportamento acontece quando 75% das requisições pertencem aos usuários de classe A, apresentado pelo gráfico da Figura 5.32. Ou seja, a política *EBS* mostra uma variação de atrasos maior que as políticas *aEBS* e σEBS , enquanto que essas, comparadas entre si, atinge o mesmo desempenho.

Com menos requisições na classe B, ou seja, 25% das solicitações, nota-se no gráfico apresentado pela Figura 5.33, que a política σEBS consegue seus melhores resultados com variações contratuais maiores, enquanto que a política *aEBS* consegue o mesmo desempenho que a σEBS com variações contratuais menores (por exemplo, 10%). Já a política

Figura 5.31: *Desvio Padrão dos Atrasos - Cenário: 75% requisição classe B.*

Figura 5.32: *Desvio Padrão dos Atrasos - Cenário: 75% requisição classe A.*

EBS oferece menos estabilidade com relação aos atrasos que as políticas desenvolvidas.

As políticas *aEBS* e σ *EBS* consegue menores desvios-padrão de atrasos para todos os cenários, exceto quando 25% de requisições estão na classe A. Nesse caso, com menor variação contratual a política *aEBS* obtém menores desvios que a *EBS*, e com variações contratuais maiores, obtém o mesmo desempenho. Já a política σ *EBS*, nesse cenário, obtém o mesmo desempenho que a *EBS*, exceto quando a variação contratual é de 30%. Nesse caso, não é possível qualquer afirmação sobre o desempenho comparativo das políticas *EBS* e σ *EBS*.

Figura 5.33: *Desvio Padrão dos Atrasos - Cenário: 25% requisição classe B.*

Comportamento do Escalonador quanto ao Desvio Padrão dos Atrasos durante a Simulação

Os gráficos dessa subseção são complementares aos gráficos descritos acima. Esses, por sua vez, relacionam o comportamento do desvio padrão dos atrasos (eixo das ordenadas) com o tempo de liberação das requisições (no eixo das abscissas). As legendas apresentam a política de escalonamento, o tamanho da janela, a proporção em cada classe e o parâmetro de avaliação.

No primeiro gráfico dessa subseção (Figura 5.34) é apresentado um cenário em que 25% das requisições são de usuários de classe A e 75% das requisições são de usuários de classe B, e referente a discrepância de 10% entre as classes de serviço. Observa-se que a política *EBS* tem maiores variações dos atrasos durante os ciclos de escalonamento. Nota-se também que as duas políticas desenvolvidas possuem um desempenho superior a política *EBS* para qualquer classe de serviço.

Com 75% de requisições na classe de serviço A e 25% das requisições do sistema na classe de serviço B, como mostra o gráfico da Figura 5.35, as políticas *aEBS* e *σEBS* apresentam um desempenho similar entres si e superior à política *EBS*.

Com uma discrepância de 20% entre as classe A e B, como mostra o gráfico da Figura 5.36, em que 25% das requisições do sistema são de classe A e 75% das requisições são de classe B, as políticas desenvolvidas apresentam melhores resultados que a política de escalonamento *EBS*.

Com maiores proporções de requisições na classe A, como mostra o gráfico da Figura 5.37, as políticas *aEBS* e *σEBS* também apresentam melhores resultados que a política *EBS*.

Quando a discrepância entre as classes é de 30%, cenário apresentado pelo gráfico

Figura 5.34: Comportamento do Desvio Padrão durante a Simulação - Cenário: 25% A e 75% B, Variação Contratual 10%.

Figura 5.35: Comportamento do Desvio Padrão durante a Simulação - Cenário: 75% A e 25% B, Variação Contratual 10%.

da Figura 5.38, em que há menores quantidades de requisições na classe mais exigente, observa-se que as políticas *EBS*, *aEBS* e σ *EBS* apresentam resultados semelhantes para a classe A. Entretanto, para a classe de serviço B, as novas abordagens apresentam melhores resultados que a *EBS*.

Diferentemente do que acontece quando a proporção de requisições na classe A aumenta e para a classe B diminui, como mostra o gráfico da Figura 5.39. Nesse caso,

Figura 5.36: *Comportamento do Desvio Padrão durante a Simulação - Cenário: 25% A e 75% B, Variação Contratual 20%.*

Figura 5.37: *Comportamento do Desvio Padrão durante a Simulação - Cenário: 75% A e 25% B, Variação Contratual 20%.*

as novas abordagens apresentam melhores resultados que a política de referência, para qualquer classe de serviço.

Esses resultados são caracterizados por apenas uma simulação, referente a um determinado usuário de classe A e outro de classe B.

Figura 5.38: Comportamento do Desvio Padrão durante a Simulação - Cenário: 25% A e 75% B, Variação Contratual 30%.

Figura 5.39: Comportamento do Desvio Padrão durante a Simulação - Cenário: 75% A e 25% B, Variação Contratual 30%.

Tempos de Resposta Abaixo da Média Contratada

Os gráficos apresentados nas Figuras 5.40 a 5.43 mostram, no eixo das ordenadas, o número de vezes em que os tempos de resposta individuais das requisições ficaram abaixo da média contratada para cada política analisada.

Com relação a variação contratual, sabe-se que para a classe A, quanto menor o valor

do contrato, menor é o número de requisições que tiveram seus tempos de resposta abaixo da média contratada. Enquanto que, para a classe de serviço B, quanto maior a variação contratual, maior é o valor contratado, logo, maior é o número de requisições que tiveram seus tempos de resposta abaixo da média contratada ou $R_{tr\downarrow}$.

Quanto a política de escalonamento, o gráfico apresentado no gráfico da Figura 5.40, do qual mostra um cenário onde 25% das solicitações são de usuários de classe A, nota-se que as políticas *EBS* e σEBS obtêm o mesmo número de $R_{tr\downarrow}$. Já a política *aEBS* obtém um desempenho inferior, porém, correspondente a não mais que 2.5% abaixo do desempenho da *EBS*.

Figura 5.40: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe A.*

Com 75% de requisições na classe B, como mostra o gráfico da Figura 5.41, observa-se que a *aEBS* atinge um desempenho inferior a *EBS*, quanto a porcentagem de $R_{tr\downarrow}$. Entretanto, equivalente a 7% de degradação, com tendência a diminuir à medida que a variação contratual aumenta. Para a política σEBS também consta uma porcentagem menor de $R_{tr\downarrow}$ em comparação a *EBS*, mas com um desempenho superior a *aEBS*.

O gráfico da Figura 5.42, por outro lado, mostra um cenário onde há maior número de requisições de classe A, ou seja, 75% das requisições do sistema. Observa-se que as políticas *aEBS* e σEBS possuem o mesmo desempenho. Entretanto, a *EBS* apresenta maior porcentagem de $R_{tr\downarrow}$. Um valor de degradação inferior a 7.5% para a *aEBS* comparada a *EBS*, e inferior a 10% para a σEBS comparada a *EBS*.

Com 25% de requisições na classe B, como mostra o gráfico da Figura 5.43, observa-se que a política *aEBS* perde aproximadamente 7.5% de requisições com tempos de resposta abaixo da média contratada, quando comparado a *EBS*. Já a política σEBS degrada mais à medida que a variação contratual aumenta.

Resumidamente, as políticas *aEBS* e σEBS aumentam o número de requisições com

Figura 5.41: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe B.*

Figura 5.42: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 75% requisição classe A.*

tempos de resposta abaixo do contrato com o objetivo de minimizar o tempo de espera das requisições que estão a muito tempo na fila.

5.6 Considerações Finais

Este capítulo apresentou as políticas desenvolvidas para reduzir a variação dos atrasos nas Seções 5.1 e 5.2. Nessas seções, foram descritos os procedimentos para caracterizar cada uma das políticas. Também apresentou-se as possíveis causas de uma variação de atraso muito alta, assim delineando o desenvolvimento das políticas.

Em seguida, foram apresentadas as medidas para validar as políticas desenvolvidas e

Figura 5.43: *Tempos de Resposta Abaixo da Média Contratada - Cenário: 25% requisição classe B.*

a de referência na Seção 5.3. Apresentou-se também os experimentos do qual caracteriza os cenários para avaliação na Seção 5.4.

Os resultados foram discutidos na Seção 5.5, onde foram analisados primeiramente, a influência de cada fator dos experimentos nas Seções 5.5.1 e 5.5.2 do qual refere-se as janelas de operações e a variação entre os contratos de uma classe de serviço inferior sobre uma classe superior, respectivamente. Por fim, as políticas foram comparadas na Seção 5.5.3, de forma a demonstrar qual obteve melhor desempenho em relação a satisfação contratual, média dos atrasos, o desvio padrão dos atrasos e a porcentagem de requisições que tiveram seus tempos de resposta abaixo da média contratada.

O próximo capítulo apresenta as conclusões sobre o trabalho, incluindo as análises sobre a *EBS* com janelas deslizante de operações, além de um resumo do comportamento das políticas desenvolvidas para reduzir o desvio padrão dos atrasos e também apresenta as contribuições e possíveis ampliações sobre o estudo deste trabalho.

Conclusão

Esse capítulo resume os métodos para o desenvolvimento das janelas deslizantes de operações e das políticas de escalonamento αEBS e σEBS e os principais resultados obtidos das análises. Também são apresentados as contribuições desse trabalho e possíveis pesquisas futuras.

6.1 Conclusões Gerais

A pesquisa se divide em duas partes, a primeira introduz contratos formulados em janelas deslizantes de operações e a segunda parte da pesquisa apresenta formas de reduzir o desvio padrão dos atrasos. Resumidamente, as abordagens serão apresentadas nas Seções 6.1.1 e 6.1.2, abordando os principais resultados obtidos.

6.1.1 SLA com Janelas Deslizante

A política de escalonamento EBS concebida para aplicações na provisão de QoS absoluta para *Web*, demonstra resultados superiores as políticas convencionais (Casagrande *et al.*, 2007). O parâmetro contratado pelo usuário é o tempo médio de resposta. Entretanto, a média efetiva proveniente dos tempos de resposta de todas as requisições é questionada quanto ao período de utilidade do sistema. Esse período, definido pelo número total de requisições que o usuário submete ao sistema, é avaliado em proporções menores. Sendo assim, a média é proveniente dos tempos de respostas dessa proporção de requisições. Proporção essa que refere-se às requisições mais recente do usuário, e, é definida como janelas deslizantes de operações.

Os resultados apresentados no Capítulo 4, mostraram que a medida que o tamanho

da janela de operações diminui, menor é a satisfação contratual. O mesmo acontece com a porcentagem de requisições que tiveram seus tempos de resposta abaixo do contrato. Entretanto, ainda sim, é possível garantias absolutas de tempos médios de resposta com janela de tamanhos menores do que o total de requisições que os usuários submetem ao sistema. Quanto maior o número de requisições numa classe mais exigente, maior deverá ser o tamanho da janela.

6.1.2 Reduzindo o Desvio Padrão dos Atrasos

Com os contratos definidos em janelas de operações, requisições podem assumir valores elevados de *deadline*. O *deadline* definido como sendo basicamente a diferença entre a média efetiva imposta pelo usuário no sistema e a média contratada, pode, com janelas deslizantes, conter uma média efetiva baixa em relação a média contratada. Sabendo que a média é proveniente dos tempos de resposta de parte das requisições do usuário, se esses tempos forem baixos, quando diferenciado pelo contrato, pode apresentar um *deadline* alto, ou parecer que a requisição pode aguardar por muito tempo na fila de espera. Nesse caso, duas abordagens foram desenvolvidas. A primeira denominada *aEBS* pondera o tempo de espera em fila, o custo computacional da requisição e o *deadline*. A segunda abordagem, denominada *σEBS* atribui maiores prioridades às requisições já atrasadas, baseado no comportamento da dispersão dos atrasos das requisições do usuário em questão.

Os resultados primeiramente, mostraram quais são as influências das janelas deslizantes de operações e das variações contratuais nas políticas avaliadas. Abaixo a descrição resumida dos resultados de cada um.

Os **tamanhos das janelas** influenciam na política *σEBS* com relação ao desvio padrão dos atrasos, a média dos atrasos e a porcentagem de requisições que tiveram seus tempos de resposta abaixo da média contratada. Basicamente à medida que a janela aumenta de tamanho, maior são os parâmetros avaliados, exceto no cenário em que a pouca proporção de requisições na classe mais exigente. Nesse cenário, o desvio padrão e a média dos atrasos não se diferenciam com relação aos tamanhos das janelas. Com relação a política *aEBS* não se distinguem os parâmetros avaliados em cenários em que há pouca proporção de requisições na classe A e uma maior proporção de requisições na classe B. Entretanto, quando a proporção de requisições na classe A é maior, há um tendência a menores desvios-padrão e médias dos atrasos, entretanto, o intervalo de confiança sobrepõem a média das simulações. Em relação a porcentagem de requisições que tiveram seus tempos de resposta abaixo da média contratada, os resultados não se diferem para qualquer tamanho de janela. Com a política *EBS*, os parâmetros avaliados também não se distinguem para qualquer tamanho de janela.

O comportamento do escalonador quanto a **variação contratual** é o mesmo para

qualquer uma das políticas avaliadas quando a proporção de requisições na classe A é baixa. Basicamente, para os cenários da classe de serviço A, a média e o desvio padrão dos atrasos diminuem conforme a discrepância entre as classes A e B aumenta. Por outro lado, para a classe de serviço B, a média e o desvio padrão dos atrasos aumentam a medida que a variação contratual aumenta. Com maior proporção de requisições na classe mais exigente, a média e o desvio padrão dos atrasos também diminuem conforme a variação contratual aumenta. Entretanto, para a classe de serviço B, somente as políticas **EBS** e **aEBS** tendem a maiores média e desvio padrão de atrasos quando a variação contratual aumenta. Diferentemente da política **σ EBS** que provê o mesmo comportamento para qualquer variação contratual.

As análises compõem também a comparação das políticas de acordo com o desempenho relacionado às medidas. As medidas analisadas são a satisfação contratual do qual avalia a média efetiva do usuário, a média dos atrasos e desvio padrão dos atrasos. Também considera-se a porcentagem de requisições que tiveram seus tempos de resposta abaixo da média contratada. Nessas análises foi considerada apenas um tamanho de janela, ou seja, um tamanho equivalente a 10% das solicitações do usuário, pois as janelas não se diferenciam quanto à média e desvio padrão dos atrasos para a política tomada como referência, **EBS**.

De acordo com a **satisfação contratual**, os gráficos mostraram que com uma proporção menor de requisições na classe mais exigente, as políticas conseguiram atingir a satisfação máxima desejável, tanto para a classe A, quanto para a classe de serviço B. Todavia, com um número significativamente maior de requisições na classe mais exigente, e com uma discrepância alta entre os valores contratuais de cada classe, as políticas **aEBS** e **σ EBS** não obtêm uma satisfação contratual desejável na classe de serviço A. Já para a classe de serviço B, apenas a política **aEBS** não atinge uma satisfação de 100%. No entanto, esses cenários além de saturados de requisições de uma classe superior, ainda, possuem um valor muito baixo de média contratada pelos usuários de classe A. Para a política **EBS**, a satisfação contratual é de 100%, entretanto, sabe-se que a política não se preocupa com os tempos de resposta individuais, e que em determinadas ocasiões, podem atingir valores desastrosos, que são atenuados na média efetiva do usuário.

Os resultados também descrevem a **média dos atrasos** de cada política, como já mencionando anteriormente. Em cenários onde a proporção de requisições na classe mais exigente é pequena, a média dos atrasos entre as políticas **EBS** e **aEBS** são iguais para qualquer variação contratual. Já a política **σ EBS** apresentam uma tendência a maior média dos atrasos que a **EBS**. Para a classe de serviço B, a política **aEBS** apresenta desempenho superior as demais políticas. Enquanto que, as políticas **EBS** e **σ EBS** apresenta a mesma média de atrasos. Com maior proporção de requisições na classe mais exigente, a política **aEBS** apresenta menor média de atrasos comparada a **EBS**, entretanto, a medida que a variação contratual aumenta, elas se igualam. Já a política

σEBS apresenta o mesmo resultado que a EBS . Para a classe de serviço B, as políticas desenvolvidas se sobressaem em comparação com a EBS . Resumidamente é desejável que no mínimo as políticas desenvolvidas possuem o mesmo desempenho que a EBS , o que acontece na maioria dos cenários, e em outros, as políticas tendem a melhores resultados.

Em relação ao **desvio padrão dos atrasos**, quando a proporção de requisições na classe A é pequena, a política $aEBS$ consegue melhores resultados que as demais, entretanto, a medida que a variação contratual aumenta, a política EBS se iguala ao seus resultados. Já a política σEBS possui o mesmo desvio padrão padrão que a política de referência, EBS . Para os cenários de classe B, as políticas $aEBS$ e σEBS obtêm os mesmo resultados para qualquer variação contratual, e, com desvios-padrão significante-mente abaixo da política EBS . O mesmo acontece quando a proporção de requisições na classe A aumenta. Entretanto, quando a classe B possui menos requisições a medida que a variação contratual aumenta, a política $aEBS$ obtém maior desvio padrão que a σEBS . Mesmo assim, as duas políticas desenvolvidas possuem desempenho superior a EBS .

Baseado nessas análises gerais, pode-se observar que as políticas cumpriram com o objetivo proposto, que é fornecer uma menor variação dos atrasos, garantindo também a estabilidade das médias.

Sobre a **porcentagem de requisições com os tempos de resposta abaixo da média contratada**, com uma proporção menor na classe mais exigente, as políticas σEBS e EBS obtêm o mesmo desempenho. Entretanto, a política $aEBS$ possui mais requisições atrasadas. O mesmo acontece para a classe de serviço B, que em contrapartida possui uma maior proporção de requisições. Com um número maior de requisições do sistema para a classe de serviço A, as políticas $aEBS$ e σEBS assemelham-se no número de requisições com os tempos de resposta abaixo da média contratada. Contudo, possuem mais requisições atrasadas comparada a política EBS . Esse comportamento procede para a classe de serviço B, que possui menos requisições que a classe A, mas, a política σEBS atinge um desempenho ainda inferior que a política $aEBS$. Basicamente, as políticas desenvolvidas acarretam mais requisições atrasadas, porém, próximas do valor contratado, como mostra as análises dos desvios-padrão e das médias dos atrasos.

6.2 Contribuições

O trabalho contribui com uma análise da política de escalonamento EBS sobre contratos em janelas ou seções de operações. Em sistema reais, a qualidade de serviço apresentada em um grande período de tempo de atendimento pode mostrar uma satisfação contratual que não condiz com a experimentação do serviço. Para o serviço operado com a EBS pode-se observar que mesmo com seções menores ainda é possível uma satisfação admissível para os usuários do sistema.

O trabalho também contribui com duas políticas de escalonamento que oferecem mel-

hores estabilidade ao serviço oferecido aos usuários com relação aos tempos de espera das solicitações individuais. Anteriormente, a investigação tratava apenas de garantias médias de atendimento. Entretanto, médias não identificam qual o grau de dispersão dos tempos sobre elas. Determinadas aplicações tem como preferência médias um pouco mais altas do que o valor esperado, do que sistemas instáveis durante todo o tempo de atendimento as solicitações.

Além da metodologia, o trabalho ainda contribui com medidas e parâmetros que possibilitam maior caracterização e estudo do impacto da provisão de QoS e provisão de menores desvios-padrão no escalonamento de requisições de um sistema. Esse fundamental permite uma análise mais real do serviço oferecido (pela janela de operações). Além disso, o trabalho oferece uma nova maneira de se obter a dispersão dos atrasos, ou seja, durante ciclos do escalonador.

O trabalho também apresentou um estudo comparativo entre as políticas de escalonamento desenvolvidas e a que provê *QoS* absoluta, destacando-se os pontos positivos e negativos observados quando aplicado mecanismos para reduzir os desvios-padrão dos atrasos das requisições *Web*.

As contribuições do trabalho foram dois artigos publicados em eventos científicos internacionais na área:

- MONACO, F. J. ; NERY, M.; PEIXOTO, M. L. M. . **An Orthogonal Multi-Resource Real-Time Scheduling Architecture for Responsiveness QoS Requirements in SOA Environments.** In: **24th Annual ACM Symposium on Applied Computing, 2008, Honolulu.** ACM-SAC 2009, 2008.
- MONACO, F. J. ; Mamani, E. L. C. ; NERY, M. ; Nobile, P. N. . **Novel QoS Modeling Approach for Soft Real-Time Systems with Performance Guarantees.** In: **The 2009 High Performance Computing and Simulation Conference, 2009, Leipzig.** HPCS '2009, 2009.

6.3 Trabalhos Futuros

A análise da política de escalonamento *EBS* sobre janelas deslizantes de operações é uma das possíveis avaliações que poderiam ser feitas. Além disso, pode-se destacar a análise sobre os atrasos das requisições individuais e uma possível redução dos mesmos. O estudo ainda pode ser ampliando com algumas sugestões à seguir:

- Compor o serviço com garantias relativas e absolutas, ou seja, os usuários de uma classe mais prioritária terá um tempo de resposta médio sempre mais abaixo do que os de classes inferiores.

-
- Investigar a viabilidade de um mecanismo de controle de admissão para as requisições que estão muito atrasadas.
 - Avaliar as políticas (EBS , $aEBS$ e σEBS) com proporções iguais de requisições para cada usuário.
 - Avaliar as políticas (EBS , $aEBS$ e σEBS) para n-classes de serviço.
 - Investigar as políticas (EBS , $aEBS$ e σEBS) com uma taxa de utilização maior que 75%.

Referências

- Al-Anzi, F.; Allahverdi, A. (2005). Using a hybrid evolutionary algorithm to minimize variance in response time for multimedia object requests. *Journal of Mathematical Modelling and Algorithms*, v.4, n.4, p.435–453.
- Banks, J.; Carson, J. S.; Nelson, B. L.; Nicol, D. M. (1989). *Discrete-Event System Simulation*. Prentice-Hall, Upper Saddle River-NJ, USA, 3 edição.
- Berners-Lee, T.; Cailliau, R.; Groff, J.-F.; Pollermann, B. (1992). Word wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, v.2, n.1, p.74–82.
- Blake, S.; Black, D.; Carlson, M.; Davies, E.; Wang, Z.; Weiss, W. (1998). An architecture for differentiated service.
- Braden, R.; Clark, D.; Shenker, S. (1994). *Integrated Services in the Internet Architecture: an Overview*. Request for Comments: 1633.
- Braden, R.; Zhang, L.; Berson, S.; Herzog, S.; Jamin, S. (1997). *Resource ReSerVation Protocol (RSVP)*. Request for Comments: 2205.
- Casagrande, L. S. (2007). Ebs: Política para provisão de qos absoluto em escalonamento web. Dissertação de Mestrado, ICMC-USP.
- Casagrande, L. S.; Mello, R. F.; Bertagna, R.; Andrade Filho, J. A.; Monaco, F. J. (2007). Exigency-based real-time scheduling policy to provide absolute qos for web services. *SBAC-PAD: 19th International Symposium on Computer Architecture and High Performance Computing*, p. 1–8.
- Cormen, T. H.; Leiserson, C. E.; Leiserson, C. E.; Stein, C. (2001). *Introduction to Algorithms*, v. 2. The MIT Press.
- Eilon, S.; Chowdhry, I. G. (1977). Minimizing waiting time variance in the single machine problem. *Management Science*, v.23, p.567–574.

- Estrella, J. C.; Teixeira, M. A. M.; Santana, M. J.; Santana, R. H. C.; Bruschi, S. M. (2006). Negotiation mechanisms on application level: A new approach to improve quality of service in web servers. *WCCIA, The 2nd International Workshop on Collaborative Computing, Integration, and Assurance 2006, Gyeongju. IEEE proceedings*, p. 255–259.
- Fishwick, P. A. (1992). *Simpack: Getting started with simulation programming in c and c++*. Winter Simulation Conference.
- ISO/IEC (1998). *Information technology - Quality of service: Framework*. ISO/IEC International Organization for Standardization/International Electrotechnical Commission 13236.
- Jain, R. K. (1991). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. New York, USA.
- Kamienski, C. A.; Sadok, D.; Cavalcanti, D. A. T.; de Sousa, D. M. T.; Dias, K. L. (2002). Simulando a internet: Aplicações na pesquisa e no ensino. 21^a JAI (Jornada de Atualização em Informática), Congresso da SBC.
- Kang, K.-D.; Son, S. H.; Stankovic, J. A. (2003). Differentiated real-time data services for e-commerce applications. *Electronic Commerce Research*, v.3, n.1-2, p.113–142.
- Kurose, J. F.; Ross, K. W. (2006). *Redes de Computadores e a Internet: Uma abordagem Top-Down*. SP, Brasil, 3 edição.
- Lazowska, E. D.; Zahorjan, J.; Graham, G. S.; Sevcik, K. C. (1984). *Quantitative System Performance: Computer System Analysis Using Queueing Network Models*. Prentice-Hall, Inc., Upper Saddle River-NJ, USA, 2 edição.
- Leung, J. Y. T.; Whitehead, J. (1982). On the complexity of fixed-priority scheduling of periodic, real-time tasks. *Performance Evaluation* 2, v.2, n.4, p.237–250.
- Li, X.; Chen, Y.; Sun, Y.; Sawhney, R. (2007a). Minimizing class-based completion time variance on a single machine. *International Journal of Operations Research*, v.4, n.1.
- Li, X.; Ye, N.; Liu, T.; Sun, Y. (2007b). Job scheduling to minimize the weighted waiting time variance of jobs. *Comput. Ind. Eng.*, v.52, n.1, p.41–56.
- Liu, C. L.; Layland, J. W. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM (JACM)*, v.20, n.1, p.46–61.
- Liu, D.; Hu, X. S.; Lemmon, M. D.; Ling, Q. (2006). Firm real-time system scheduling based on a novel qos constraint. *IEEE Transactions on Computers*, v.55, n.3, p.320–333.

- Liu, J. W. S. (2000). *Real-Time Systems*. Prentice Hall, Upper Saddle River-NJ, USA, 1 edição.
- MacDougall, M. H. (1989). *Simulating Computer Systems - Techniques e Tools*. MIT Press, Cambridge-MA, USA.
- Messias, V. R.; Estrella, J. C.; Santana, M. J.; Santana, R. H. C.; Bruschi, S. M.; Teixeira, M. A. M. (2007). Implementação de um protótipo de servidor web distribuído com diferenciação de serviços. *XIII Simpósio Brasileiro de Sistemas Multimídia e Web WebMedia 2007*, v.1.
- Monaco, F. J.; Nery, M.; Peixoto, M. L. (2009). An orthogonal multi-resource real-time scheduling architecture for responsiveness qos requirements in soa environments. *24th Annual ACM Symposium on Applied Computing*.
- Pan, W.; Mu, D.; Wu, H.; Sun, Q. (2008). Proportional delay differentiation service in web application servers: A feedback control approach. *CCM '08: Proceeding of the 2008 ISECS International Colloquium on Computing, Communication, Control, and Management*, p. 600–604. IEEE Computer Society.
- Peixoto, M. L. M.; Monaco, F. J. (2008). Arquitetura de escalonamentos ortogonal de tempo-real para garantias de qos em servidores web. in: Workshop em desempenho de sistemas computacionais e de comunicação. *XXVIII Congresso da Sociedade Brasileira de Computação - SBC*.
- Peixoto, M. L. M.; Tott, R. F.; Monaco, J. F. (2007). Política de escalonamento de tempo-real para garantia de qos absoluta em cluster de servidores web. *Anais do XIII Webmedia*, p. 159–166.
- Prado, D. (2004). *Teoria das Filas e da Simulação*. INDG Tecnologia e Serviços LTDA, Nova Lima-MG, Brasil, 2 edição.
- Ramamritham, K.; Stankovic, J. A. (1994). Scheduling algorithms and operating systems support for real-time systems. *IEEE Transactions on Software Engineering*, v.20, n.6, p.476–493.
- Sha, L.; Abdelzaher, T.; Karl-Erik; Cervin, A.; Baker, T.; Burns, A.; Buttazzo, G.; Caccamo, M.; Lehoczky, J.; Mok, A. K. (2004). Real time scheduling theory: A historical perspective. *Real-Time System*, v.28, n.2-3, p.101–155.
- Shenoy, N.; Patchararungruang, S. (2004). A combined scheduling scheme for absolute and relative differentiated services. *(438) Communication and Computer Networks*.
- Sprunt, B.; Sha, L.; Lehoczky, J. (1989). Aperiodic task scheduling for hard real-time systems. v. 1, p. 27–60, Pittsburg-PA, USA.

- Teixeira, M. M.; Santana, M. J.; Santana, R. C. (2004). Using adaptive priority controls for service differentiation in qos-enabled web servers. v. 3036, p. 537–540, Berlin, Heidelberg. International Conference on Computational Science (ICCS).
- Teixeira, M. M.; Santana, M. J.; Santana, R. C. (2005). Servidor web com diferenciação de serviços: Fornecendo qos para os serviços da internet. Fortaleza-CE, Brasil. XIII Simpósio Brasileiro de Redes de Computadores (SBRC).
- Traldi, O. A.; Barbato, A. K.; Santana, R. H. C. (2006). Service differentiating algorithms for qos-enabled web servers. *WebMedia '06: Proceedings of the 12th Brazilian symposium on Multimedia and the web*, p. 263–272, Natal-RN, Brasil. ACM.
- Vasiliou, N. (2000). Overview of internet qos and web server qos.
- Wu, X.; Li, M.; Wu, J. (2006). Enhanced demand-driven service differentiation algorithm in web clusters. *ICEBE '06: Proceedings of the IEEE International Conference on e-Business Engineering*, p. 386–391, Washington, DC, USA. IEEE Computer Society.
- Ye, N.; Gel, E. S.; Li, X.; Farley, T.; Lai, Y.-C. (2005). Web server qos models: applying scheduling rules from production planning. *Comput. Oper. Res.*, v.32, n.5, p.1147–1164.
- Ye, N.; Li, X.; Farley, T.; Xu, X. (2007). Job scheduling methods for reducing waiting time variance. *Computers & Operations Research*, v.34, n.10, p.3069–3083.
- Zhou, X.; Wei, J.; Xu, C.-Z. (2004). Modeling and analysis of 2d service differentiation on e-commerce servers. *ICDCS '04: Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)*, p. 740–747, Washington, DC, USA. IEEE Computer Society.

Apêndice

A.1 Fórmula da Variância

Com o objetivo de garantir que o desempenho do escalonamento não se degrade com as formulações das equações, necessitou-se desmembrar a fórmula original da variância. Abaixo, os passos do desmembramento.

$$\sigma_u^2 = M_2 - M_1^2$$

$$\dot{\sigma}_u^2 = \dot{M}_2 - \dot{M}_1^2$$

$$M_1 = \frac{\sum x_i}{n}$$

$$S_1 = n_1 \cdot M_1$$

$$\dot{M}_1 = \frac{\sum x_i + y}{n + 1}$$

$$(n + 1) \cdot \dot{M}_1 = S_1 + y$$

$$(n + 1) \cdot \dot{M}_1 = n \cdot M_1 + y$$

$$\dot{M}_1 = \frac{n \cdot M_1 + y}{n + 1}$$

$$M_2 = \frac{\sum x_i^2}{n}$$

$$S_2 = n_2 \cdot M_2$$

$$\dot{M}_2 = \frac{\sum x_i^2 + y^2}{n + 1}$$

$$(n + 1) \cdot \dot{M}_2 = S_2 + y^2$$

$$(n + 1) \cdot \dot{M}_2 = n \cdot M_2 + y^2$$

$$\dot{M}_2 = \frac{n \cdot M_2 + y^2}{n + 1}$$

$$\begin{aligned}
\dot{\sigma}_u^2 &= \frac{n \cdot M_2 + y^2}{n+1} - \frac{(n \cdot M_1 + y)^2}{(n+1)^2} \\
\dot{\sigma}_u^2 &= \frac{n \cdot (n+1) \cdot M_2 + (n+1) \cdot y^2 - n^2 \cdot M_1^2 - 2 \cdot n \cdot M_1 \cdot y - y^2}{(n+1)^2} \\
\dot{\sigma}_u^2 &= \frac{n \cdot (n+1) \cdot M_2 + n \cdot y^2 + y^2 - n^2 \cdot M_1^2 - 2 \cdot n \cdot M_1 \cdot y - y^2}{(n+1)^2} \\
\dot{\sigma}_u^2 &= \frac{n^2 \cdot M_2 + n \cdot M_2 + n \cdot y^2 - (n \cdot M_1)^2 - 2 \cdot n \cdot M_1 \cdot y}{(n+1)^2} \\
\dot{\sigma}_u^2 &= \left(\frac{n}{n+1} \right)^2 \cdot \sigma_u^{2'} + \frac{n}{(n+1)^2} \cdot (M_2 - 2 \cdot M_1 \cdot y + y^2) \\
\dot{\sigma}_u^2 &= \left(\frac{n}{n+1} \right)^2 \cdot \sigma_u^{2'} + \frac{n}{(n+1)^2} \cdot [M_2 - M_1^2 + (M_1^2 - 2 \cdot M_1 \cdot y + y^2)] \\
\dot{\sigma}_u^2 &= \left(\frac{n}{n+1} \right)^2 \cdot \sigma_u^{2'} + \frac{n}{(n+1)^2} \cdot [\sigma_u^{2'} + (M_1 - y)^2] \\
\dot{\sigma}_u^2 &= \frac{n \cdot (n+1)}{(n+1)^2} \cdot \sigma_u^{2'} + n \cdot \left(\frac{M_1 - y}{n+1} \right)^2 \\
\dot{\sigma}_u^2 &= \frac{n^2 + n}{(n+1)^2} \cdot \sigma_u^{2'} + n \cdot \left(\frac{M_1 - y}{n+1} \right)^2 \\
\dot{\sigma}_u^2 &= \frac{(n^2 + n)}{(n+1)^2} \cdot \sigma_u^{2'} + n \cdot \left(\frac{M_1 - y}{n+1} \right)^2 \\
\dot{\sigma}_u^2 &= \frac{n}{(n+1)} \cdot \sigma_u^{2'} + n \cdot \left(\frac{M_1 - y}{n+1} \right)^2
\end{aligned}$$