CÁTIA GARCIA MORAIS

MODELAGEM DE DESEMPENHO DE SERVIDORES WEB EMPREGANDO A TEORIA NETWORK CALCULUS

CURITIBA

2005

CÁTIA GARCIA MORAIS

MODELAGEM DE DESEMPENHO DE SERVIDORES WEB EMPREGANDO A TEORIA NETWORK CALCULUS

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.
Orientadora: Profa. Dra. Cristina Duarte Murta

CURITIBA

2005

Ministerio da Educação
Universidade Federal do Parana
Mestrado em Informatica

PARECER

Nos, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informatica, da aluna *Catia Garcia Morais*, avaliamos o trabalho intitulado, "*MODELAGEM DE DESEMPENHO DE SERVIDORES WEB EMPREGANDO A TEORIA NEWORK CALCULUS* ", cuja defesa foi realizada no dia 22 de março de 2005, as treze e trinta horas, no Auditório do Departamento de Informática do Setor de Ciencias Exatas da Universidade Federal do Parana Apos a avaliação, decidimos pela aprovação da candidata

Curitiba, 22 de março de 2005

Profª Dra Cristina Duarte Murta
DINF/UFPR – Orientadora

Prof Dr Virgílio A Fernandes Almeida
UFMG – Membro Externo

Prof Dr Ricardo Luders
CEFET/PR – Membro Externo

Prof Dr Elias Procópio Duarte Jr
DINF/UFPR – Membro Interno

MINISTÉRIO DA EDUCAÇÃO - UNIVERSIDADE FEDERAL DO PARANÁ - Setor de Ciências Exatas - Coordenação de Pós-Graduação em Informática

Agradecimentos

Agradeço a Deus por conceder esta oportunidade.

Agradeço à professora Cristina Duarte Murta pela competência e dedicação e também pela orientação em todas as fases do desenvolvimento deste trabalho.

Agradeço a meu esposo Paulo Henrique Cayres e a minha filha Paula Morais Cayres pelas manifestações de amor e carinho.

Um agradecimento especial ao professor Alexandre Ibrahim Direne e ao professor Elias Procópio Duarte Jr. pelo apoio em momentos importantes durante o mestrado.

Aos amigos do mestrado e a todos aqueles que de alguma forma contribuíram para a realização deste trabalho.

Resumo

Network Calculus é uma teoria que modela o desempenho de sistemas de filas e permite o cálculo de limites determinísticos de desempenho, quando os fluxos de entrada obedecem a certas restrições. Este trabalho descreve a aplicação desta teoria para a modelagem de desempenho de servidores Web. O desempenho dos servidores Web é vital para o sucesso de muitas organizações. Acompanhar, avaliar e modelar o desempenho dos servidores Web são tarefas fundamentais para prover acesso eficiente e confiável, em especial no caso de servidores populares e eventualmente sobrecarregados. Diferentes técnicas estão disponíveis para a avaliação de desempenho de um sistema. Cada uma apresenta possibilidades, vantagens e limitações e é aplicável em diferentes contextos e com custos diversos. Assim, estudar as possibilidades de aplicação de uma nova teoria para modelagem de desempenho de um sistema como os servidores Web é uma tarefa importante.

Para demonstrar a aplicação da teoria e seus resultados em servidores Web, as funções que descrevem os resultados do Network Calculus, que são funções descritas na álgebra min-plus, foram implementadas e testadas com registros de acesso de alguns servidores Web. Os principais resultados da teoria, a saber, o limite de atraso, o limite do tamanho da fila e o limite do fluxo de saída, foram obtidos e são apresentados para os servidores em questão. Outros aspectos também discutidos são a comparação com a análise operacional, a aplicação da teoria para modelar controle de admissão e a complexidade computacional das funções implementadas.

Abstract

Network Calculus is a theory that models the perfomance of queueing systems. It provides deterministic bounds on performance, when the input flows obey certain restrictions. This work describes the application of this theory for the performance modeling of Web servers. The performance of Web servers is vital for the success of many organizations. To manage, evaluate and model the performance of Web servers are basic tasks to provide efficient and trustworthy access, in special in the case of popular and eventually overloaded servers. Different techniques are available for the performance evaluation of a system. Each one presents possibilities, advantages and limitations. The application of a new theory for modeling of performance of a system such as Web servers is an important task.

To demostrate the application of the theory and its results in Web servers, the functions that describe the results of Network Calculus, which are described in min-plus algebra, have been implemented and tested with traces of some Web servers. The main results of the theory, which are the delay bound, the backlog bound and the output flow had been gotten and are presented for the servers considered. Other aspects of the theory also argued are the comparison with the operational analysis, the application of the theory to model admission control and the analysis of the computational complexity of the implemented functions.

Sumário

AGRADECIMENTOS i

RESUMO ii

ABSTRACT iii

LISTA DE FIGURAS vii

LISTA DE TABELAS viii

LISTA DE ABREVIATURAS ix

1 Introdução 1
- 1.1 Contextualização . . . 2
- 1.2 Motivação . . . 2
- 1.3 Objetivos . . . 3
- 1.4 Organização do Texto . . . 4

2 Avaliação e Modelagem de Desempenho de Sistemas Computacionais 5
- 2.1 Técnicas para Avaliação e Modelagem de Desempenho . . . 6
  - 2.1.1 Experimentação no Sistema Real . . . 6
  - 2.1.2 Modelos de Simulação . . . 7
  - 2.1.3 Modelos Analíticos . . . 8
- 2.2 Métricas para Avaliação de Desempenho . . . 12
- 2.3 Carga de Trabalho . . . 13
- 2.4 Avaliação de Desempenho em Servidores Web . . . 14
  - 2.4.1 Arquitetura do Software Servidor . . . 15
  - 2.4.2 Técnicas de Avaliação de Desempenho de Servidores Web . . . 16
  - 2.4.3 Métricas de Desempenho de Servidores Web . . . 17
  - 2.4.4 Carga de Trabalho . . . 18
- 2.5 Considerações Finais . . . 19

3 A Teoria Network Calculus 20
- 3.1 Álgebra Min-Plus . . . 21
- 3.2 Funções Não Decrescentes e suas Operações . . . 22
- 3.3 Funções de Entrada e de Saída . . . 24
- 3.4 Tamanho da Fila e Atraso Virtual . . . 25
- 3.5 Curva de Chegada e Curva de Serviço . . . 27
- 3.6 Principais Resultados: os Três Limites . . . 29

3.7 Considerações Finais 33

4 Avaliação de Desempenho de Servidores Web com Network Calculus 34
4.1 Considerações Iniciais 35
4.2 Cálculo das Funções de Entrada e de Saída 36
4.3 Cálculo do Tamanho da Fila e do Atraso Virtual 39
4.4 Cálculo da Curva de Chegada Mínima 41
4.5 Cálculo da Curva de Saída 42
4.6 Cálculo dos Três Limites 44
4.7 Aspectos da Implementação do Network Calculus 46
4.7.1 Composição dos Fluxos 47
4.7.2 Implementação das Equações 47
4.8 Considerações Finais 50

5 Outros Experimentos e Aplicações da Teoria Network Calculus em Servidores Web 52
5.1 Comparação com a Lei de Little 53
5.1.1 Derivação da Lei de Little 53
5.1.2 Resultados da Comparação com a Lei de Little 54
5.2 Experimento com Carga Intensa 56
5.2.1 Considerações Iniciais sobre o *Trace* 56
5.2.2 Cálculo da Curva de Saída 58
5.2.3 Cálculo do Tamanho da Fila e do Atraso Virtual 59
5.2.4 Cálculo dos Três Limites 60
5.3 Controle de Admissão com a Teoria Network Calculus 62
5.4 Considerações Finais 64

6 Conclusões 66

REFERÊNCIAS BIBLIOGRÁFICAS 69

Lista de Figuras

2.1 Uma rede de filas. 9
2.2 Variáveis comuns usadas na análise de uma fila. 9
2.3 Visão do sistema como uma caixa preta. 12
2.4 Elementos do servidor Web [28]. 15

3.1 (a) Um circuito simples e (b) Um nó de uma rede de computadores. 23
3.2 Operações de (a) convolução e (b) deconvolução. 24
3.3 Modelo de fluxo de dados do sistema S. 25
3.4 Funções cumulativas $R(t)$ e $R^*(t)$ para um servidor. 25
3.5 Tamanho da fila e atraso virtual, obtidos através das funções cumulativas de entrada e de saída. 26
3.6 Função de entrada e curva de chegada mínima no sistema S, considerando α igual a 2, 3 e 5. 28
3.7 Funções de entrada e saída no sistema S, considerando β igual a 2, 3 e 5. . 29
3.8 Curvas de chegada e de serviço no sistema S, considerando (a) $\alpha = \beta$, (b) $\alpha < \beta$ e (c) $\alpha > \beta$. 31
3.9 Curva α^* que limita o fluxo de saída do sistema S, considerando (a) $\alpha = \beta$, (b) $\alpha < \beta$ e (c) $\alpha > \beta$. 32

4.1 Funções cumulativas $R(t)$ e $R^*(t)$ no servidor Web em (a) 1 minuto e (b) 30 minutos de observação. 37
4.2 Tamanho da fila no servidor Web em (a) 1 minuto e (b) 30 minutos de observação. 39
4.3 Atraso virtual no servidor Web em (a) 1 minuto e (b) 30 minutos de observação. 40
4.4 Curva de chegada mínima no servidor Web em (a) 1 minuto e (b) 30 minutos de observação. 42
4.5 Curvas de entrada e de saída no servidor Web em (a) 1 minuto e (b) 30 minutos de observação, considerando a taxa de serviço de 4 req/s. 43
4.6 Curvas de entrada e de saída no servidor Web em (a) 1 minuto e (b) 30 minutos de observação, considerando a taxa de serviço de 5 req/s. 44
4.7 Curva α^* em (a) 1 minuto e (b) 30 minutos de observação, considerando α como $R(t)$ e β como $R^*(t)$. 45
4.8 Curva α^* em (a) 1 minuto e (b) 30 minutos de observação, considerando α como α mínima e β com taxa de serviço de 5 req/s. 46
4.9 Curva α^* em (a) 1 minuto e (b) 30 minutos de observação, considerando α como $R(t)$ e β com taxa de serviço de 5 req/s. 46
4.10 Código para cálculo do tamanho da fila. 48
4.11 Código para cálculo do atraso virtual. 49

4.12 Código para cálculo da operação $(f \otimes g)(t)$. 50
4.13 Código para cálculo da operação $(f \oslash g)(t)$. 51
4.14 Código para cálculo do limite do atraso virtual. 51

5.1 Chegadas e finalizações do sistema. 54
5.2 Taxa de chegada de requisições no sistema. 58
5.3 $R(t)$ e $R^*(t)$ para o sistema, considerando a taxa de serviço de 400 req/s. . 59
5.4 Tamanho da fila no sistema, considerando a taxa de serviço de 400 req/s. . 60
5.5 Atraso virtual no sistema, considerando a taxa de serviço de 400 req/s. . . 61
5.6 (a) Curva de chegada mínima no sistema, (b) Curvas R experimental, R^*, α mínima e β, considerando a taxa de serviço de 400 req/s, (c) Limite no fluxo de saída no sistema. 62
5.7 Tamanho da fila como função do processo de chegada. 63
5.8 Relação entre o tamanho da fila e a taxa média de serviço do sistema. . . . 64
5.9 CDF do tamanho da fila. 65

Lista de Tabelas

4.1 Estatísticas da taxa de chegada de requisições no servidor Web em 1 minuto e 30 minutos de observação. 37
4.2 Estatísticas da taxa de saída de requisições no servidor Web em 1 minuto e 30 minutos de observação. 38
4.3 Estatísticas do tamanho da fila em cada segundo no servidor Web em 1 minuto e 30 minutos de observação. 39
4.4 Estatísticas do atraso virtual em cada segundo no servidor Web em 1 minuto e 30 minutos de observação. 41
4.5 Limites do tamanho da fila e do atraso virtual para 1 minuto e 30 minutos, para os 3 casos de suposição de α e β. 45

5.1 Tamanho da fila no servidor Web em 1 minuto e 30 minutos de observação, obtido a partir da Lei de Little. 55
5.2 Estatísticas do tamanho da fila em cada segundo no servidor Web em 1 minuto e 30 minutos de observação. 55
5.3 Estatísticas da taxa de chegada de requisições no sistema. 57
5.4 Estatísticas do tamanho da fila no sistema. 60
5.5 Estatísticas do atraso virtual no sistema. 61

Lista de Abreviaturas

CPU	*Central Processing Unit*
FCFS	*First Come First Served*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
LAN	*Local Area Network*
NC	*Network Calculus*
URL	*Uniform Resource Locator*
WWW	*World Wide Web*

Capítulo 1

Introdução

Desde sua introdução, no início dos anos noventa, a *World Wide Web* (WWW) evoluiu muito no que diz respeito à sua utilização. Essa evolução foi conseqüência do aparecimento de diversas aplicações anteriormente inexistentes, tais como sistemas de bibliotecas digitais, educação à distância, áudio e vídeo sob demanda e comércio eletrônico. Essas aplicações ocasionaram um aumento enorme do tráfego na Internet e na WWW [28]. Alguns sítios Web populares recebem milhões de acessos por dia, podendo resultar em tempos de resposta extremamente altos. A demora na exibição de uma página frustra muitos usuários e causa preocupação aos administradores Web. Um dos principais problemas enfrentados por estes é adequar os recursos para atender às exigências dos usuários. Esse desafio exige que os administradores Web estejam aptos a monitorar gargalos, prever a capacidade de seus sítios Web e determinar a melhor maneira de resolver os problemas de desempenho causados pelo aumento da carga de trabalho imposta aos seus sítios Web.

O ambiente Web possui características particulares quando comparado com outros sistemas distribuídos tradicionais. Algumas dessas características provocam um impacto considerável no planejamento dos servidores Web [28]. Primeiro, o número potencial de clientes, além de estar em constante crescimento, pode chegar à ordem de dezenas de milhões [19]. Além disso, há uma grande diversidade de comportamento dos usuários durante a navegação, dificultando uma correta previsão da carga de trabalho imposta a um servidor Web. Em determinados momentos, os servidores Web estão quase inativos e

em outros, o tráfego pode aumentar bastante. Quando há um grande aumento na taxa de requisições aos servidores Web, os tempos de resposta e erros de conexão aumentam significativamente. A sobrecarga pode acontecer devido à saturação do processador ou da memória principal do servidor Web ou, até mesmo, a redução da capacidade de conexão do servidor à rede [28].

1.1 Contextualização

O desempenho dos servidores Web é vital para o sucesso de muitas organizações. É importante que o tempo de resposta no sítio Web seja aceitável, ou então os usuários não utilizarão o serviço e a organização perderá o negócio. Acompanhar, avaliar e modelar o desempenho dos servidores Web são tarefas fundamentais para prover acesso eficiente e confiável, em especial no caso de servidores populares e eventualmente sobrecarregados.

Diferentes técnicas, metodologias e teorias estão disponíveis para a avaliar e modelar o desempenho de um sistema [21, 22, 23, 29]. Cada uma apresenta possibilidades, vantagens e limitações e é aplicável em diferentes contextos e com custo de avaliação diverso. Assim, estudar as possibilidades de aplicação de uma nova teoria para análise e modelagem de desempenho de um sistema como os servidores Web é uma tarefa importante.

Há na literatura uma grande quantidade de trabalhos que aborda os problemas de desempenho dos servidores Web (veja [8, 33, 32, 13] e as referências destes). Existem trabalhos que apresentam propostas de modificação no servidor para prevenir problemas de sobrecarga transiente [20, 39, 12, 36] e também propostas de novas arquiteturas para servidores [41, 40]. Todos estes trabalhos utilizam métodos tradicionais de avaliação de desempenho [22, 23, 21, 28].

1.2 Motivação

Uma nova teoria para modelar o desempenho de sistemas foi proposta em trabalhos recentes [24, 10, 1, 25, 34]. Esta teoria, denominada Network Calculus, vem sendo aplicada no contexto de redes de computadores. A fundamentação matemática desta teoria é a

álgebra min-plus, que é uma álgebra baseada nas operações de mínimo e de adição. Os trabalhos têm mostrado que limites determinísticos de atraso e de tamanho de fila podem ser expressos utilizando esta álgebra. Além disso, com a teoria Network Calculus é possível entender algumas propriedades de redes com serviços integrados, controle de fluxo por janela e escalonamento.

Network Calculus é uma teoria para sistemas de filas que modela problemas de fluxo e estabelece garantias determinísticas para qualidade de serviço em sistemas, quando observadas certas restrições no fluxo de entrada. A partir dos conceitos da teoria Network Calculus é possível derivar limites determinísticos para métricas importantes de desempenho tais como atraso e tamanho da fila e modelar o fluxo de saída a partir de parâmetros da entrada e do sistema. Esta teoria tem sido aplicada em redes de computadores e utilizada, por exemplo, para verificar atrasos, dimensionar tamanho de filas, determinar largura de banda efetiva, modelar escalonadores comuns, entre outros [26].

Não foi encontrado nenhum trabalho que descreva qualquer aplicação prática desta teoria nem trabalhos que aplicam o Network Calculus para analisar desempenho de servidores Web. A teoria Network Calculus é bastante discutida no contexto de redes de computadores, especificamente em roteadores e, embora possa ser aplicada genericamente a qualquer servidor, não foi encontrada nenhuma referência à sua aplicação em servidores Web.

1.3 Objetivos

O objetivo principal deste trabalho é explorar a aplicação da teoria Network Calculus para avaliação de desempenho de servidores Web. A teoria foi interpretada do contexto de redes de computadores para o contexto de servidores Web, considerando o funcionamento e operação destes sistemas. Dentre os objetivos específicos, pode-se destacar:

- aplicação prática da teoria Network Calculus para modelar e avaliar o desempenho de servidores Web;

- comparação entre os resultados da teoria Network Calculus e os resultados da Lei de Little;
- discussão sobre a aplicação da teoria Network Calculus para modelar controle de admissão no contexto dos servidores Web.

1.4 Organização do Texto

Este trabalho está organizado em seis capítulos. No capítulo 2 são apresentados conceitos importantes sobre avaliação de desempenho de sistemas computacionais: técnicas, métricas e carga de trabalho. Este capítulo também descreve o funcionamento de um servidor Web e questões relativas à avaliação de seu desempenho. O capítulo 3 apresenta os conceitos básicos e a notação da álgebra min-plus, necessários ao entendimento da teoria Network Calculus, bem como os fundamentos e os principais resultados dessa teoria. A aplicação da teoria Network Calculus para modelar e avaliar desempenho de servidores Web é mostrada no capítulo 4, onde também são apresentados os experimentos realizados e os resultados obtidos. No capítulo 5 são apresentadas outras aplicações da teoria Network Calculus em servidores Web, que são a comparação com os resultados da Lei de Little e a sua utilização para modelar o controle de admissão, evitando a sobrecarga no servidor. Por fim, o capítulo 6 apresenta as conclusões obtidas com o desenvolvimento do trabalho.

Capítulo 2

Avaliação e Modelagem de Desempenho de Sistemas Computacionais

A melhor maneira de estudar o desempenho de um determinado sistema seria executar a carga de trabalho real na plataforma de hardware e software e medir os resultados [28]. Contudo, muitas vezes essa técnica não é viável, pois a carga pode não ser conhecida ou não estar disponível, é caro instrumentar o sistema ou até mesmo o sistema pode não estar disponível para medição. Diferentes técnicas, metodologias e teorias são conhecidas para avaliação de desempenho de sistemas computacionais. Cada uma apresenta possibilidades, vantagens e limitações, e é aplicável em diferentes contextos e com custo de avaliação diverso.

Este capítulo apresenta conceitos importantes em avaliação de desempenho de sistemas computacionais: técnicas, métricas e carga de trabalho. O funcionamento de um servidor Web e questões relativas à avaliação de seu desempenho também são discutidas. Estes conceitos são importantes para apresentar a terminologia empregada no trabalho e também para contextualizar a teoria Network Calculus no processo de avaliação de desempenho de servidores Web.

2.1 Técnicas para Avaliação e Modelagem de Desempenho

Existem três técnicas para avaliação e modelagem de desempenho: experimentação, simulação e modelagem analítica. A decisão sobre a técnica a ser utilizada requer a análise de várias questões e limitações do ambiente de teste, tais como o estágio do ciclo de vida em que o sistema se encontra, o tempo disponível para a avaliação, as ferramentas disponíveis para a avaliação (linguagens, instrumentos, metodologias), a precisão desejada e também o custo esperado [21]. Algumas vezes é útil utilizar duas ou mais técnicas simultaneamente para validar um resultado ou para distribuir os custos de avaliação. Cada uma destas técnicas é apresentada a seguir.

2.1.1 Experimentação no Sistema Real

A experimentação é uma técnica de medição de desempenho de sistemas reais e consiste em monitorar o sistema enquanto ele está sendo submetido a uma carga em particular. Esta técnica é aplicada em um sistema que está em um estágio de desenvolvimento pós-protótipo. A ferramenta desta técnica é a instrumentação do sistema e apresenta uma precisão variável e um alto custo de implementação [21].

Um monitor é uma ferramenta utilizada para observar as atividades de um sistema. Em geral, os monitores coletam resultados de desempenho, produzem estatísticas e apresentam os resultados. Alguns monitores também identificam áreas com problemas e sugerem correções. Monitores são utilizados não apenas por analistas de desempenho, mas também por programadores e gerentes de sistema. Monitoramento é o primeiro passo em medições de desempenho.

O *benchmarking* é um método tradicional e amplamente utilizado para medir desempenho de um sistema real (hardware e software). Um *benchmark* é um conjunto representativo de programas executado em diferentes computadores e redes, medindo seu desempenho. Grande parte da popularidade dos *benchmarks* vem do fato que eles possuem métricas bem definidas de desempenho e cargas de trabalho padronizadas. Os experimentos podem ser repetidos. Entretanto, a avaliação feita por essas ferramentas ainda pode

apresentar falhas. Isso ocorre porque o ambiente em que o teste é realizado nem sempre é igual, ou mesmo equivalente, ao ambiente real a que o sistema está submetido. Dessa forma pode-se chegar a conclusões sobre o sistema que não condizem com a realidade.

A técnica de experimentação tem grande importância prática por identificar problemas correntes, como a necessidade de ajustes de parâmetros, e também por identificar potenciais problemas futuros. A principal vantagem é que o desempenho do sistema real é obtido, e não o desempenho do modelo do sistema, pois as interações que afetam o desempenho do sistema real podem ser difíceis de se captar no modelo do sistema. Como desvantagens pode-se citar a necessidade de ter um sistema em execução e de instrumentar o sistema. Além disso, é difícil estimar o tempo gasto para instrumentar, realizar as medidas e modificar o sistema para estudar o efeito das alterações.

2.1.2 Modelos de Simulação

A simulação pode ser utilizada para avaliar e modelar o desempenho de um sistema computacional. Neste caso, o simulador é um programa de computador que simula o comportamento de desempenho do sistema. Um simulador é construído a partir de um modelo de desempenho do sistema.

Se o sistema está na fase de projeto, a simulação é uma técnica bastante utilizada para predizer o desempenho ou comparar várias alternativas. Além disso, mesmo que o sistema esteja disponível para medição, o modelo de simulação pode ser empregado por fornecer alternativas de comparação com uma variedade de cargas e ambientes. A simulação é uma técnica aplicada em qualquer estágio do ciclo de vida de um sistema e o tempo exigido para sua aplicação é médio. Esta técnica utiliza como ferramenta linguagens de programação e apresenta uma precisão moderada e um médio custo de implementação [21].

Entretanto, modelos de simulação podem falhar e muito tempo pode ser gasto em seu desenvolvimento. Escolher a linguagem é um importante passo no processo de desenvolvimento de um modelo de simulação [21]. Existem quatro opções: linguagens de simulação, linguagens de propósito geral, extensões de linguagens de propósito geral e pacotes de

simulação. Cada uma delas apresenta vantagens e desvantagens, em relação a consumo de tempo, facilidades embutidas na linguagem, até a familiariedade do programador e do analista com a linguagem.

A simulação é uma ferramenta versátil, poderosa e extremamente útil na avaliação de desempenho. As principais vantagens são que os modelos de simulação podem ser construídos com níveis arbitrários de detalhes e que permitem simular situações complexas que são analiticamente intratáveis. Como desvantagens pode-se citar a complexidade no desenvolvimento do simulador e o tempo de execução da simulação.

Um simulador pode utilizar números aleatórios para gerar variáveis aleatórias, que representam tempos de chegada e de serviços no sistema, de acordo com distribuições de probabilidade. Com base nesses valores, questões de desempenho podem ser respondidas utilizando-se técnicas estatísticas para fornecer valores estimados. Para avaliar o desempenho de um sistema, uma vez construído o modelo probabilístico, isto é, asserções são feitas sobre o processo de chegada e o tempo de serviço de tarefas no sistema, as respostas às questões de desempenho podem, em teoria, ser analiticamente determinadas. Entretanto, na prática, estas questões são muito difíceis de serem determinadas analiticamente e as respostas a elas podem ser realizadas por um estudo de simulação [35]. A simulação é uma alternativa aos modelos analíticos apresentados a seguir.

2.1.3 Modelos Analíticos

Em sistemas computacionais, muitas tarefas compartilham recursos tais como CPU, discos e outros dispositivos. Como, normalmente, somente uma tarefa por vez pode utilizar o recurso, todas as outras tarefas ficam esperando em filas por aquele recurso. O conjunto de recursos dá origem a uma rede de filas, como mostra a Figura 2.1.

Uma das formas mais conhecidas para a construção de modelos analíticos de filas é empregando a **teoria de filas**. O sistema pode ser descrito por equações matemáticas cujas soluções consistem na solução do modelo. A teoria de filas é uma ferramenta matemática empregada para realizar análise de desempenho de um sistema que pode ser modelado como uma rede de filas. Ela ajuda a determinar, por exemplo, o tempo que as tarefas

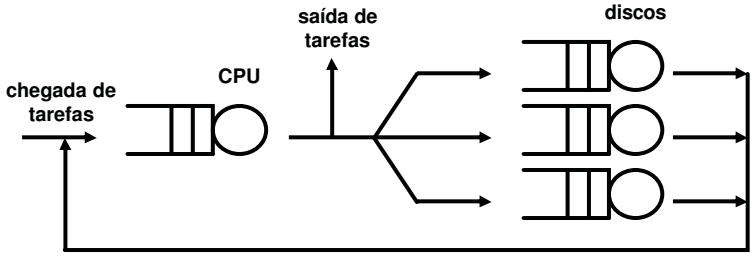


Figura 2.1: Uma rede de filas.

gastaram em várias filas dentro do sistema computacional. Estes tempos podem então ser combinados para predizer o tempo de resposta, que é basicamente o tempo total que a tarefa gastou dentro do sistema, incluindo o tempo de serviço.

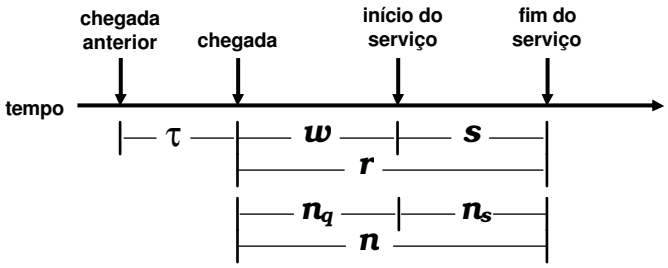


Figura 2.2: Variáveis comuns usadas na análise de uma fila.

Algumas das variáveis mais utilizadas na análise de filas estão ilustradas na Figura 2.2 e são descritas a seguir. Existe um grande número de relacionamentos entre estas variáveis, que são aplicadas no estudo de filas e serão empregadas neste trabalho.

- τ é o tempo entre chegadas, isto é, tempo entre duas chegadas sucessivas. A média dos tempos entre chegadas é representada por $E[\tau]$ e a taxa de chegada é igual a $\lambda = 1/E[\tau]$.

- w é o tempo de espera na fila, isto é, o intervalo de tempo entre o instante de chegada e o instante que o serviço começa.

- s é o tempo de serviço por tarefa. A média dos tempos de serviço é representada por $E[s]$ e a taxa de serviço por servidor é igual a $\mu = 1/E[s]$.

- r é o tempo de resposta. Esta variável também é conhecida como atraso do sistema e inclui o tempo de espera pelo serviço e o tempo recebendo o serviço ($r = w + s$).

- n_q é o número de tarefas esperando por atendimento.

- n_s é o número de tarefas recebendo atendimento.

- n é o número total de tarefas no sistema. Esta variável também é conhecida como tamanho da fila e inclui as tarefas sendo atendidas bem como as tarefas que estão esperando atendimento ($n = n_q + n_s$).

Um modelo analítico pode ser aplicado em qualquer estágio do ciclo de vida de um sistema e o tempo exigido para sua aplicação é pequeno. Esta técnica utiliza como ferramenta a teoria de análise e apresenta pequena exatidão e um baixo custo de implementação. A principal vantagem desta técnica é ser barata, pois consiste na solução de equações matemáticas. Como desvantagens pode-se citar que é uma técnica que aproxima a realidade por um modelo, as suposições simplificam o modelo para que as equações sejam tratáveis e, devido a isso, pode perder exatidão. Normalmente é aplicada como auxílio no projeto preliminar de sistemas.

Os modelos analíticos são classificados como probabilísticos ou determinísticos. A seguir é apresentada uma discussão sobre os dois tipos de modelos, apontando as entradas, os resultados e as possibilidades de cada um.

Modelos Probabilísticos

Os sistemas de filas probabilísticos são caracterizados por seis parâmetros: processo de chegada, distribuição do tempo de serviço, número de servidores, tamanho máximo da fila, tamanho da fonte ou população e política de escalonamento da fila. O processo de chegada é caracterizado pela distribuição do tempo entre as chegadas, que define a taxa de chegada de tarefas no sistema. O tempo de serviço representa o tempo gasto no servidor

para executar uma determinada tarefa e define a taxa de serviço do sistema. O tamanho máximo da fila, também denominada tamanho do *buffer*, representa a quantidade de tarefas que pode permanecer no sistema devido a limitações de espaço e inclui as tarefas na fila e em serviço. É comum assumir que o tamanho do *buffer* é infinito; neste caso, nenhuma tarefa é perdida. O número total de tarefas que podem chegar ao sistema é o tamanho da fonte ou população. Na maioria dos sistemas reais, o tamanho da população é finito, mas quando este tamanho é muito grande, é comum que ele seja definido como infinito para simplificar a análise. A política de escalonamento define a ordem na qual as tarefas são atendidas (por exemplo, ordem de chegada, prioridade, atendimento aleatório e outros).

Para especificar um sistema de fila, é necessário especificar estes seis parâmetros. Os analistas utilizam a **notação de Kendall**, na forma $A/S/m/B/K/SD$, onde cada letra corresponde a um parâmetro [21]. Assim, A é a distribuição dos tempos entre chegadas, S é a distribuição dos tempos de serviço, m é o número de servidores, B é a capacidade do sistema, K é o tamanho da população e SD é a disciplina de serviço.

Este tipo de modelo de filas utiliza a probabilidade para obtenção dos resultados. O modelo de filas mais simples é aquele que tem uma única fila. Tal modelo pode ser empregado para analisar, por exemplo, recursos individuais de um sistema computacional. O modelo de fila mais conhecido é $M/M/1$, onde os tempos entre chegadas e os tempos de serviço são exponencialmente distribuídos e existe somente um servidor, não há limitação no tamanho do *buffer* nem da população e a disciplina de serviço é *First Come First Served* (FCFS). Para analisar este tipo de fila, é necessário conhecer somente λ e μ.

Modelos Determinísticos

Um grande número de problemas em análise de desempenho de sistemas computacionais pode ser resolvido usando a **análise operacional** [23], que não requer qualquer asserção sobre o processo de chegada ou sobre a distribuição dos tempos de serviço. A análise operacional consiste em um conjunto de leis denominadas leis operacionais. Estas leis estabelecem relações simples entre quantidades de desempenho diretamente

mensuráveis em um sistema computacional e são observadas em qualquer sistema, são diretamente testáveis e podem ser verificadas através de medições.

As quantidades medidas são denominadas quantidades operacionais e são medidas em um sistema durante um período finito de observação. Por exemplo, considere o sistema como uma caixa preta, conforme mostra a Figura 2.3. Se durante um tempo de observação T são verificadas A chegadas de tarefas em um sistema, C tarefas completadas, e B é o tempo em que o sistema esteve ocupado, pode-se calcular a taxa de chegada no sistema A/T, a taxa de serviço C/T, a utilização B/T e o tempo médio de serviço B/C.

Figura 2.3: Visão do sistema como uma caixa preta.

Várias leis operacionais são conhecidas, por exemplo, a lei da utilização, a lei do fluxo forçado, a lei do tempo de resposta e a lei da demanda de serviço [23, 21, 28]. A lei operacional mais conhecida é a **lei de Little**, que é de particular interesse para este trabalho.

A lei de Little estabelece a relação entre o número médio de tarefas em um dado sistema e o tempo médio de resposta do sistema, $E[n] = \lambda \times E[r]$. Este relacionamento pode ser aplicado a qualquer sistema onde o número de tarefas entrando no sistema é igual ao número de tarefas completadas, isto é, nenhuma nova tarefa é criada no sistema e nenhuma tarefa é perdida dentro do sistema. Esta asserção é conhecida como balanço de fluxo [21]. A lei de Little está baseada na visão do sistema como uma caixa preta.

2.2 Métricas para Avaliação de Desempenho

Quando um sistema é submetido a um estudo de desempenho, vários critérios ou métricas devem ser escolhidas para avaliação. Uma forma de selecionar a métrica apropriada é verificar os serviços que são oferecidos pelo sistema e quais são as possibilidades de resposta [21]. Geralmente as respostas são classificadas em três categorias: correta, incorreta ou

não realizada. Por exemplo, um roteador em uma rede pode repassar o pacote para o destino certo, para um destino errado ou não repassar.

Se o sistema realiza o serviço corretamente, seu desempenho é medido pelo tempo gasto para realizar o serviço, a taxa na qual o serviço é realizado, e a quantidade de recurso consumida enquanto realizava o serviço. Estas três métricas são denominadas tempo de resposta, taxa de processamento (*throughput*) e utilização, respectivamente. O tempo de resposta compreende o tempo entre a chegada da tarefa e a resposta do sistema, a taxa de processamento é a quantidade de tarefas que são executadas pelo sistema por unidade de tempo e a utilização representa o percentual de tempo que o sistema ficou ocupado realizando a tarefa.

Se o sistema realiza o serviço incorretamente, um erro ocorreu. É interessante classificar os erros para determinar a probabilidade em que ocorrem e apresentar o intervalo de tempo entre erros. Se o sistema não realiza um serviço, ele está indisponível. É interessante classificar as falhas para determinar a duração de um evento e o intervalo de tempo entre falhas.

As métricas associadas aos três tipos de resposta do sistema (correta, incorreta ou não realizada) também são chamadas de métricas de desempenho, confiabilidade e disponibilidade. Para cada serviço oferecido pelo sistema, um número variado de métricas de desempenho, confiabilidade e disponibilidade pode ser aplicado. Para a escolha das métricas, deve-se considerar métricas com pequena variabilidade, evitar redundância (eliminar métricas similares) e escolher um conjunto completo que avalie todas as possíveis saídas do sistema.

2.3 Carga de Trabalho

Uma carga é o conjunto de todas as tarefas (requisições de serviço) submetidas a um sistema durante um período de tempo. O desempenho de um sistema depende fortemente das características da carga. Para que as medidas de desempenho sejam significativas, a carga deve ser cuidadosamente selecionada.

A escolha da carga é uma parte muito importante do processo de avaliação de desempenho, pois as conclusões obtidas do estudo podem ser incorretas ou inúteis, caso a escolha da carga não seja adequada. No processo de selecionar a carga, quatro parâmetros devem ser considerados [21]: os serviços que serão executados pelo sistema; o nível de detalhe desejado sobre os serviços executados, por exemplo, a freqüência de um pedido ou a média de demanda de um recurso; a representatividade frente uma aplicação real e a atualização da carga, uma vez que os usuários de sistema mudam com bastante freqüência seu padrão de comportamento.

Um modelo de carga é uma representação que imita a carga de trabalho real em estudo [28]. Ele pode ser um conjunto de programas escritos e implementados com o objetivo de testar artificialmente um sistema em um ambiente controlado. Um modelo de carga também pode ser o conjunto de dados de uma distribuição, que serve como entrada para um modelo analítico de um sistema.

Os modelos podem ser reais ou sintéticos [21]. Uma carga real é aquela observada em um sistema durante sua operação normal, por exemplo, registros das operações de um servidor Web observado por 30 minutos. Uma carga sintética ou artificial é classificada em executável, por exemplo, geradores de carga sintética; ou não-executável, por exemplo, valores médios de parâmetros descritivos.

Certamente lidar com cargas de trabalho reais com um grande número de elementos é uma tarefa difícil. Portanto, para trabalhar com problemas práticos, é preciso reduzir e resumir as informações necessárias para descrever a carga de trabalho. Os modelos devem ser compactos, porém representativos. É preciso criar um modelo de carga de trabalho que preserve as características mais relevantes da carga de trabalho real.

2.4 Avaliação de Desempenho em Servidores Web

Um servidor Web é uma combinação de uma plataforma de hardware, sistema operacional, software servidor e conteúdo (páginas a serem servidas), conforme ilustra a Figura 2.4 [28]. Todos esses componentes têm influência no desempenho dos servidores Web.

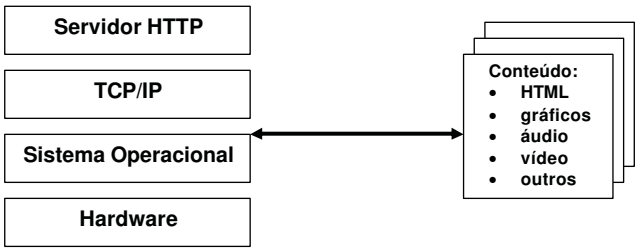


Figura 2.4: Elementos do servidor Web [28].

Nesta seção é apresentada a descrição da arquitetura do servidor Web e também são discutidas as questões de análise de desempenho desses servidores: técnicas, métricas e carga de trabalho.

2.4.1 Arquitetura do Software Servidor

O software do servidor Web, também conhecido como servidor HTTP ou *daemon* HTTP (*HyperText Transfer Protocol*), é um programa que controla o fluxo de dados recebidos e emitidos em um computador conectado a uma intranet ou à Internet. Basicamente, um software de servidor Web "escuta" os pedidos HTTP que chegam dos clientes pela rede. O programa servidor estabelece a conexão solicitada entre ele e o cliente, envia o arquivo solicitado e retorna ao modo de escuta [28].

Como um servidor pode receber requisições de muitos clientes, uma fila de requisições é formada no servidor. Se apenas uma requisição for atendida de cada vez, os recursos na máquina servidora podem ser subutilizados, a taxa de processamento do servidor pode ser baixa e o tempo de resposta aos clientes aumentará com o aumento da carga no servidor [28]. Para agilizar o atendimento aos pedidos, os servidores Web tratam de mais de uma requisição de cada vez.

O protocolo implementado entre clientes e servidores é um protocolo requisição-resposta (GET-RESPONSE), em que clientes enviam requisições e os servidores respondem às requisições do cliente. Esta interação pode ser vista como uma "transação Web" [28]. Uma requisição HTTP inclui várias partes: o método que especifica a ação a ser realizada, por

exemplo, GET; o *Uniform Resource Locator* (URL) que identifica o nome do localizador do recurso solicitado; e outras informações, por exemplo, autenticação.

Quando chega uma requisição do cliente, o servidor analisa o pedido, de acordo com o protocolo HTTP. O servidor executa o método solicitado e, no caso de um GET, o servidor consulta o arquivo em sua árvore de documentos utilizando o sistema de arquivos, pois o arquivo pode estar na memória cache ou nos discos. Então, o servidor lê o conteúdo do arquivo e o escreve na porta da rede. Depois que o arquivo foi completamente enviado, o servidor fecha a conexão, se o protocolo HTTP não persistente for usado. O tempo de residência no servidor é o tempo gasto na execução de um pedido HTTP, e inclui o tempo de serviço e o tempo de espera nos diversos componentes do servidor, como processador, disco e placa de rede.

2.4.2 Técnicas de Avaliação de Desempenho de Servidores Web

A Web é caraterizada por uma diversidade de componentes: diferentes navegadores e servidores executando em uma série de plataformas com diferentes capacidades de processamento. A variedade de componentes torna mais complexo o problema de monitorar e coletar dados de desempenho [28]. Os usuários Web podem experimentar atrasos longos, variáveis e imprevisíveis na rede, que dependem da largura de banda da conexão e do congestionamento da rede, entre outros fatores.

Entender e medir desempenho de serviços fim-a-fim é uma tarefa desafiadora. As técnicas atuais incluem máquinas que enviam requisições a servidores e coletam dados de desempenho (*benchmark*) ou a instrumentação de páginas Web com código que relata questões de desempenho.

Existem *benchmarks* específicos para medir desempenho de servidores Web. Estes programas simulam a ação de um navegador e geram requisições ao servidor, recebem as respostas e coletam dados. Os *benchmarks* de servidores Web mais utilizados são Webstone [30], SPECweb [37], SURGE [9], HTTPerf [31] e TPC-W [38]. É importante observar que os *benchmarks* de servidores Web normalmente são executados em LANs pequenas e isoladas, quase sem erros de transmissão. Além disso, as latências são muito

mais altas nos ambientes do mundo real do que nas redes locais usadas nos laboratórios de *benchmarking* [28]. Assim, os resultados da análise do *benchmark* da Web devem levar em consideração estas observações.

O trabalho [13] propõe uma nova abordagem para medir desempenho de sítios Web. O sistema proposto é um monitor, denominado EtE, que passivamente coleta registros de pacotes de um sítio servidor para determinar características de desempenho de serviço. Com estes registros, a composição de uma página individual é reconstruída e características de desempenho são integradas aos acessos de todos os clientes. Como vantagens dessa abordagem pode-se citar que é obtida informação completa de todos os acessos, não apenas de uma amostra; é possível quantificar os benefícios da cache de navegadores e rede para o desempenho de servidores Web; é possível identificar características de acessos que são interrompidos pelo cliente, entre outros. Como limitações do monitor EtE pode-se citar que não trata informação HTTP de conexões criptografadas; é aplicado em um único servidor Web, ou em um grupo de servidores Web, com um único ponto de entrada ou saída, onde o monitor pode capturar todo o tráfego; em caminhos com servidores proxy, o monitor pode somente medir o tempo de resposta do proxy em vez dos clientes reais.

2.4.3 Métricas de Desempenho de Servidores Web

Uma requisição Web utiliza diversos recursos, incluindo processadores e discos, no cliente, no servidor, nas redes, nos roteadores e sistemas intermediários [28]. O tempo de resposta total de uma requisição é composto pelo tempo gasto na rede, tempo gasto no servidor Web e tempo gasto na máquina do cliente. O tempo gasto na rede é composto pelo tempo de latência e tempo de transmissão. O tempo gasto no servidor Web é a soma do tempo de serviço e tempo de fila. O tempo de serviço é o tempo que uma requisição passa recebendo atendimento em qualquer um dos recursos do servidor, por exemplo, realizando uma operação de entrada/saída no servidor de arquivos. O tempo de fila é o tempo gasto esperando até que um recurso esteja disponível, por exemplo, esperando que o processador esteja disponível ou esperando acesso a disco.

A taxa em que as requisições HTTP são atendidas é uma métrica comum de processamento em um servidor Web e é expressa em operações HTTP por segundo (HTTPops/s) [28]. Devido à grande variabilidade no tamanho dos objetos Web solicitados, a taxa de processamento é também medida em bits por segundo (b/s) ou bytes por segundo.

O tempo de resposta e a taxa de processamento são as duas métricas de desempenho mais importantes para sistemas Web [28]. Enquanto o tempo de resposta é a medida de desempenho de maior interesse para usuários, a taxa de processamento é mais interessante para os administradores de sistemas.

2.4.4 Carga de Trabalho

O desempenho de um sistema distribuído com muitos clientes, servidores e redes depende bastante das características de sua carga. Considere um servidor Web que foi observado durante 30 minutos e 180.000 requisições foram concluídas [28]. A carga de trabalho do servidor Web durante esse período de 30 minutos é o conjunto de 180.000 requisições. As características da carga de trabalho são representadas por um conjunto de informações, por exemplo, tempo de chegada e de conclusão da requisição, tempo de CPU, tamanho do objeto solicitado, para cada uma das 180.000 requisições da Web.

Existe um conjunto considerável de trabalhos sobre caracterização da carga de trabalho do tráfego da Web [4, 3, 14]. Algumas das características consideradas tratam das distribuições de tamanho de arquivo, distribuição da popularidade do arquivo, auto-similaridade no tráfego da Web, localidade de referência e padrões de requisição do usuário. É importante também saber que outras propriedades básicas foram encontradas na análise do tráfego real da Web, uma delas é a tendência de se observar o tráfego em rajadas [2]. Uma inspeção visual do número de requisições que chegam a um servidor Web em diferentes escalas de tempo, ou seja, em intervalos de tempo de tamanho variável, pode mostrar as rajadas ou a alta variabilidade do processo de chegada. Este tipo de processo de chegada pode degradar o desempenho se não for levado em consideração.

Os *benchmarks* utilizados para medir desempenho de servidores Web normalmente empregam uma carga de trabalho padrão, que inclui páginas HTML geradas estática e

dinamicamente. As características de cada conjunto de páginas, ou seja, tamanhos de arquivo e freqüências de acesso, podem ser modeladas através de parâmetros de entrada. As características da carga de trabalho do SPECweb [37], por exemplo, foram coletadas de registros (*logs*) de vários servidores populares da Internet e alguns sítios Web menores. Assim, a carga do SPECweb tenta imitar padrões de acesso aos documentos de um servidor Web típico, onde 70% das requisições são de páginas estáticas e 50% dos arquivos requisitados são de tamanho até 10KB.

2.5 Considerações Finais

Este capítulo mostrou que diferentes abordagens podem ser utilizadas para modelar e avaliar o desempenho de sistemas computacionais. A escolha da técnica apropriada depende de várias questões tais como o custo esperado, o tempo disponível para avaliação e a disponibilidade do sistema. Os modelos analíticos determinísticos são de especial interesse deste trabalho. No próximo capítulo será apresentada a teoria que foi utilizada para modelar o desempenho de servidores Web.

As métricas normalmente utilizadas em avaliação de desempenho foram apresentadas, bem como questões sobre a escolha da carga de trabalho no processo de avaliação de desempenho. Técnicas, métricas e modelos de carga também foram discutidos no contexto de servidores Web, que é o objeto de estudo deste trabalho.

Capítulo 3

A Teoria Network Calculus

A teoria Network Calculus (NC) consiste em um conjunto de desenvolvimentos recentes em modelagem matemática de redes. Network Calculus pode ser vista como uma teoria de sistemas aplicada a redes de computadores. A fundamentação matemática desta teoria é a álgebra min-plus [7].

Network Calculus, descrito em [15, 16] e completamente desenvolvido na última década [10, 24, 1], fornece expressões concisas para limites de atraso e tamanho de fila experimentados por um fluxo individual em um ou mais nós de uma rede. Esta abordagem considera que o fluxo do tráfego é desconhecido, mas apresenta restrições de regularidade, diferentemente de outras metodologias que tratam o tráfego como processo estocástico [27].

Enquanto a teoria de filas tradicional trata de processos estocásticos e distribuições de probabilidade, Network Calculus assume que o tráfego é desconhecido mas satisfaz certas restrições na chegada de pacotes e gera implicações na taxa de serviço do sistema. Estas restrições permitem que limites nos atrasos de pacotes e no tamanho da fila possam ser derivados, os quais podem ser imediatamente utilizados para quantificar o comportamento de uma rede em tempo real. A teoria de filas tradicional, por outro lado, normalmente produz valores médios e distribuições para os resultados. A derivação de limites é com freqüência difícil e limites superiores de atrasos fim-a-fim podem não existir ou podem ser impossíveis de serem calculados. O processo de chegada de pacotes no Network Calculus

é descrito pela "curva de chegada", que quantifica restrições no número de pacotes ou no número de bits de um fluxo em um intervalo de tempo para um determinado nó da rede.

Este capítulo apresenta uma revisão dos conceitos básicos e da notação da álgebra min-plus que são utilizados na teoria NC, bem como alguns conceitos e os principais resultados do Network Calculus. Neste capítulo estão apresentados apenas os conceitos básicos da teoria necessários ao entendimento deste trabalho. Este texto não representa um resumo da teoria Network Calculus, para o qual ficam sugeridas as referências [27] e [11] para uma discussão completa.

3.1 Álgebra Min-Plus

Na álgebra convencional, a estrutura algébrica $(\mathbb{R}, +, \times)$ envolve elementos reais e os operadores de adição e multiplicação. Para a construção da álgebra min-plus, as operações de adição e de multiplicação são substituídas, respectivamente, pelas operações de mínimo e de adição. A estrutura algébrica resultante é $(\mathbb{R} \cup \{+\infty\}, \wedge, +)$.

Para exemplificar, considere a propriedade distributiva na álgebra convencional $a \times (b + c) = (a \times b) + (a \times c)$ e a mesma propriedade na álgebra min-plus $a + (b \wedge c) = (a + b) \wedge (a + c)$. Quando esta propriedade é aplicada respectivamente na álgebra convencional e na álgebra min-plus, sendo $a = 2, b = 3$ e $c = 5$, os seguintes valores são obtidos:

$2 \times (3 + 5) = (2 \times 3) + (2 \times 5) = 6 + 10 = 16$

$2 + (3 \wedge 5) = (2 + 3) \wedge (2 + 5) = 5 \wedge 7 = 5$

A notação $\wedge$ representa o mínimo, isto é, $a \wedge b = \texttt{min}[a..b]$. Note que o mínimo de um conjunto nem sempre existe. Uma generalização do conceito de mínimo é o ínfimo. Por exemplo, o intervalo aberto (a, b) não tem mínimo pois $a \notin (a, b)$, e o ínfimo é a. Por outro lado, se o mínimo de um conjunto existe, ele é idêntico ao seu ínfimo, isto é, $\texttt{min}[a..b] = \texttt{inf}[a..b] = a$. A operação de ínfimo é representada pelo operador `inf`.

Outra definição importante na álgebra min-plus é o conceito de supremo. As mesmas considerações podem ser feitas para distingüir o supremo do máximo. A notação $\vee$ representa o máximo, isto é, $a \vee b = \texttt{max}[a..b]$. O intervalo aberto (a, b) não tem máximo pois $b \notin (a, b)$, e o supremo é b. Por outro lado, se o máximo de um conjunto existe, ele

é idêntico ao seu supremo, isto é, $\texttt{max}[a..b] = \texttt{sup}[a..b] = b$. A operação de supremo é representada pelo operador `sup`.

As operações de mínimo, ínfimo, supremo e máximo são representadas, respectivamente, pelos operadores `min`, `inf`, `sup` e `max`.

Seja S um subconjunto não vazio de $\mathbb{R}$. Se f é uma função de S em $\mathbb{R}$, então $f(S)$ será denotada por $f(\mathrm{S}) = \{t \text{ tal que } t = f(s) \text{ para algum } s \in S\}$. O ínfimo deste conjunto tem duas notações equivalentes:

$$\inf f(S) = \inf_{s \in S}\{f(s)\}.$$

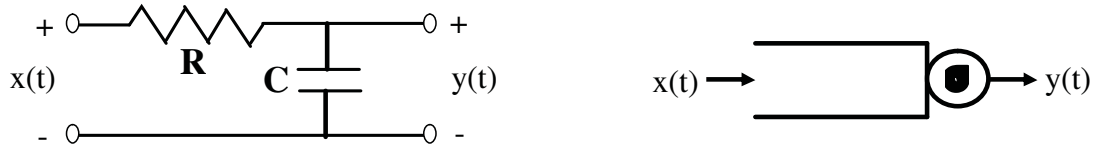
3.2 Funções Não Decrescentes e suas Operações

Uma função f é não decrescente se e somente se $f(s) \leq f(t)$ para todo $s \leq t$. Seja G um conjunto de funções não decrescentes, e F o conjunto de funções não decrescentes com $f(t) = 0$ para $t < 0$. Na álgebra min-plus, a operação de mínimo entre duas funções é definida por $(f \wedge g)(t) = f(t) \wedge g(t)$ e a operação de adição é definida por $(f + g)(t) = f(t) + g(t)$.

As operações de convolução e deconvolução conhecidas da álgebra convencional também estão definidas na álgebra min-plus. A convolução é uma operação que descreve a saída (resposta) de um sistema cuja operação é descrita por uma dada função g, quando este sistema recebe uma entrada f. Então, a saída h do sistema é representada por $f \otimes g$. A deconvolução é um processo utilizado para reverter os efeitos da convolução em um conjunto de dados, por exemplo, tem-se h e g e deseja-se obter f.

Para exemplificar a operação de convolução na álgebra min-plus, considere o circuito apresentado na Figura 3.1 (a) [26]. Se o sinal de entrada é a tensão $x(t)$, então a saída $y(t)$ deste circuito simples pode ser obtida pela convolução de x com a resposta ao impulso deste circuito, dada por $h(t) = exp(-t/RC)/RC$ para $t \geq 0$, onde R é a resistência e C é a capacitância. Então:

$$y(t) = (h \otimes x)(t) = \int_0^t h(t-s)x(s)ds.$$



(a) (b)

Figura 3.1: (a) Um circuito simples e (b) Um nó de uma rede de computadores.

Considere agora um nó de uma rede de computadores, conforme mostra a Figura 3.1 (b) [26]. Este nó é um dispositivo (*shaper*) que força um fluxo de entrada $x(t)$ a ter uma saída $y(t)$ de acordo com uma determinada taxa σ, mesmo que isto gere atraso de alguns bits na fila do dispositivo. Aqui os "sinais" de entrada e de saída são fluxos cumulativos, definidos como o número de bits vistos no fluxo durante o intervalo $[0..t]$. Utilizando a álgebra min-plus, x e y formam a relação:

$$y(t) = (\sigma \otimes x)(t) = \inf_{0 \leq s \leq t} \{\sigma(t-s) + x(s)\}.$$

Sejam f e g duas funções de F. A operação de convolução min-plus de f por g é definida pela Equação 3.1 e a operação de deconvolução é definida pela Equação 3.2. Convolução é a soma que expressa a quantidade de sobreposição de uma função f quando sobre ela é deslocada outra função g. A convolução expressa o resultado da combinação de duas funções. A deconvolução é o processo usado para reverter os efeitos da convolução nos dados registrados [42].

$$(f \otimes g)(t) = \inf_{0 \leq s \leq t} \{f(t-s) + g(s)\} \tag{3.1}$$

$$(f \oslash g)(t) = \sup_{u \geq 0} \{f(t+u) - g(u)\} \tag{3.2}$$

Se $f(t)$ e $g(t)$ são infinitas para algum t, então a Equação 3.2 não está definida. Ao contrário da convolução min-plus, a função $(f \oslash g)(t)$ não é necessariamente 0 para $t \leq 0$. Para uma discussão detalhada sobre as propriedades da álgebra min-plus e as propriedades dos operadores de convolução e deconvolução fica sugerida a referência [27].

A operação de convolução está ilustrada na Figura 3.2 (a) e a operação de deconvolução está ilustrada na Figura 3.2 (b). A função f foi obtida a partir de uma carga real de um servidor Web [18] e a função g foi gerada como sendo um fluxo cumulativo de 5 unidades de serviço por unidade de tempo. A função $(f \otimes g)(t)$ da Figura 3.2 (a) é o resultado da aplicação da entrada f em um sistema que opera segundo a função g. É possível notar que a saída $(f \otimes g)$ é a combinação das duas funções e é igual ou inferior a $f(t)$.

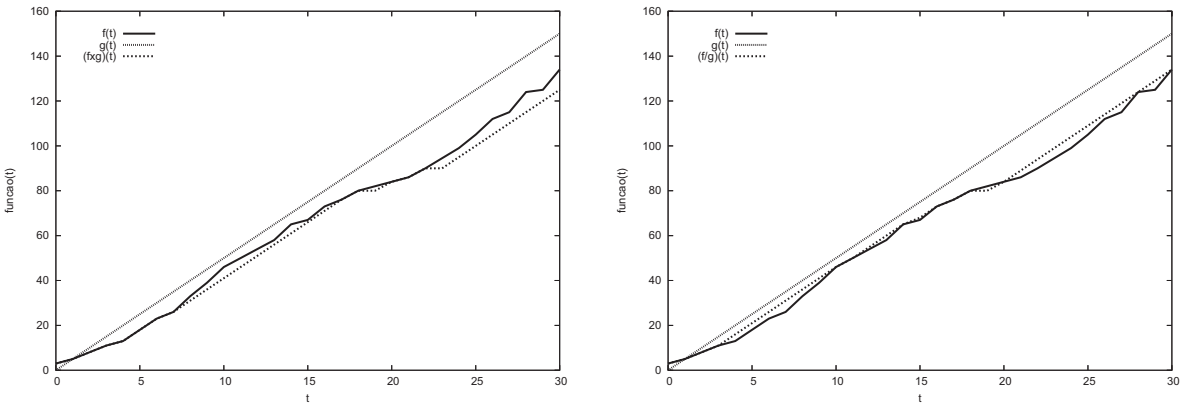


Figura 3.2: Operações de (a) convolução e (b) deconvolução.

3.3 Funções de Entrada e de Saída

Considere um sistema S, o qual pode ser visto como uma caixa preta. O sistema S pode ser, por exemplo, um *buffer*, um roteador em uma rede, uma subrede, um servidor ou ainda uma rede completa. A entrada do sistema S é vista como um fluxo, que é descrito por uma função não decrescente. Fluxos de dados são representados por funções cumulativas do número de objetos visto no fluxo no intervalo $[0..t]$. Os objetos podem ser bits, bytes, pacotes, células, palavras, requisições, ou qualquer unidade que representa a granulação mínima de serviço do sistema. Pode-se utilizar um modelo de tempo discreto ou contínuo. O modelo de fluxo de dados está ilustrado na Figura 3.3.

S recebe dados de entrada, descritos pela função cumulativa $R(t)$ e os entrega depois de um atraso variável. Considere $R^*(t)$ como sendo a função cumulativa de saída do sistema S. As funções de entrada e de saída, respectivamente $R(t)$ e $R^*(t)$, estão apresentadas na

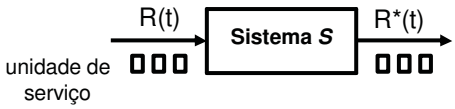


Figura 3.3: Modelo de fluxo de dados do sistema S.

Figura 3.4. Neste sistema cada pacote de tamanho 1Kbyte demora exatamente 3 unidades de tempo para ser servido.

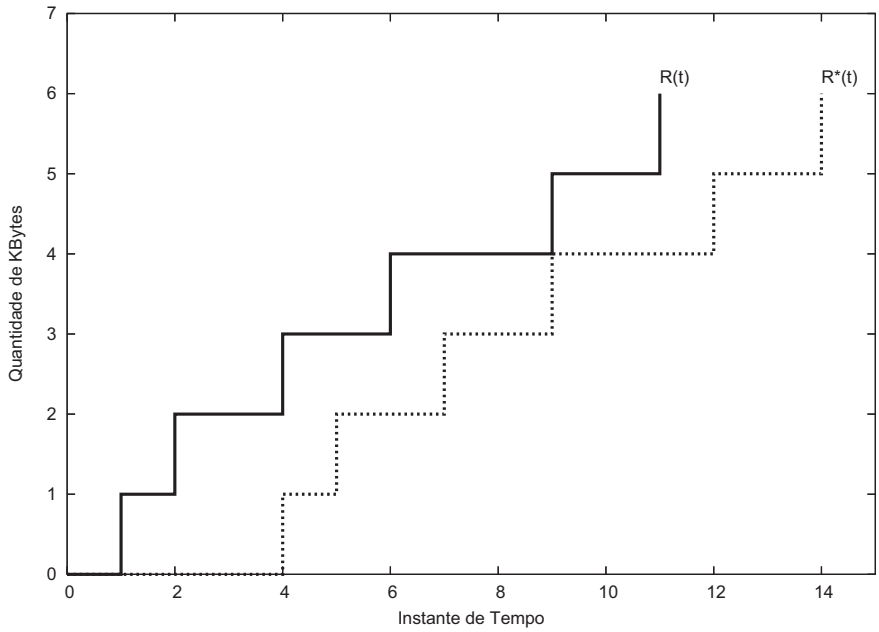


Figura 3.4: Funções cumulativas $R(t)$ e $R^*(t)$ para um servidor.

3.4 Tamanho da Fila e Atraso Virtual

A partir das funções cumulativas de entrada e de saída são derivadas duas métricas de interesse: tamanho da fila (*backlog*) e atraso virtual (*virtual delay*), conforme mostra a Figura 3.5. Para a derivação destes resultados, o pressuposto do balanço de fluxo se aplica.

O tamanho da fila é a quantidade de unidades do fluxo (bits, bytes, pacotes, etc.) que está presente no sistema em cada instante de tempo t, definido pela Equação 3.3. Assim, o tamanho da fila representa a distância vertical entre as funções de entrada e de saída.

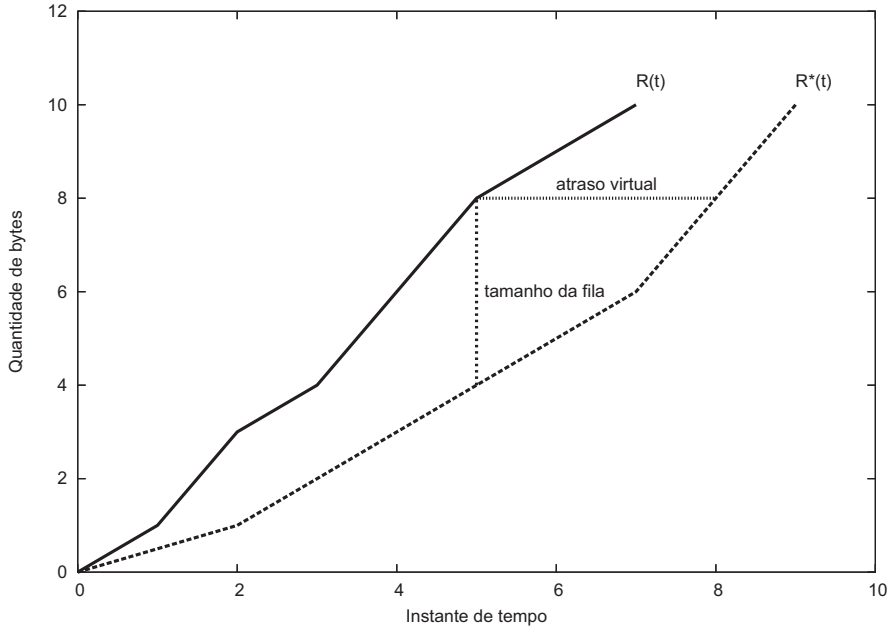


Figura 3.5: Tamanho da fila e atraso virtual, obtidos através das funções cumulativas de entrada e de saída.

$$w(t) = R(t) - R^*(t) \tag{3.3}$$

O número total de tarefas no sistema, definido no capítulo 2, pode ser redefinido como a quantidade de unidades do fluxo presente no sistema, ou seja, o *backlog*, que será denominado **tamanho da fila** no restante deste trabalho. Assim, o tamanho da fila representará o número de tarefas em atendimento bem como as tarefas que estão esperando atendimento.

O atraso virtual no instante t é o atraso que pode ser sentido por uma unidade de serviço que chegou no instante t, se todas as unidades recebidas anteriormente forem servidas antes desta. O atraso virtual é definido pela Equação 3.4. Assim, o atraso virtual representa a distância horizontal entre as funções de entrada e de saída.

$$d(t) = \texttt{inf}\{\tau \geq 0 \mid R(t) \leq R^*(t+\tau)\} \tag{3.4}$$

O tempo gasto no sistema por uma tarefa, definido no capítulo 2, pode ser redefinido como o atraso que pode ser sentido por uma unidade de serviço que chegou no instante

t, ou seja, o *virtual delay*, que será denominado **atraso virtual** no restante deste trabalho. Assim, o atraso representará a soma do tempo de serviço (tempo que uma tarefa passa recebendo atendimento) e do tempo de fila (tempo que uma tarefa gasta esperando atendimento).

3.5 Curva de Chegada e Curva de Serviço

Para prover garantia de serviço, a rede deve oferecer algum mecanismo específico aos fluxos de dados como, por exemplo, reserva de largura de banda. Por outro lado, o tráfego enviado pelas origens precisa ser limitado para não sobrecarregar o sistema e, consequentemente, diminuir seu desempenho. Isto pode ser feito utilizando o conceito de curva de chegada.

Dada uma função não decrescente α definida para $t \geq 0$, pode-se dizer que um fluxo $R(t)$ é limitado por α se e somente se, para todo $s \leq t$:

$$R(t) - R(s) \leq \alpha(t - s).$$

A definição pode ser reconstruída, utilizando a convolução min-plus, conforme a Equação 3.5.

$$R(t) \leq \inf_{0 \leq s \leq t} \{\alpha(t - s) + R(s)\} = (\alpha \otimes R)(t) \tag{3.5}$$

A função α é chamada curva de chegada para o fluxo R. A curva de chegada mínima para o fluxo $R(t)$ é a função $(R \oslash R)(t)$ (ver Teorema 1.2.2 em [27]).

Considere um sistema S que foi observado por 30 unidades de tempo. A Figura 3.6 mostra a função de entrada $R(t)$, a curva de chegada mínima $(R \oslash R)(t)$ e a função $(\alpha \otimes R)(t)$ para valores de α constantes e iguais a 2, 3 ou 5 unidades de serviço por unidade de tempo. Como pode ser observado na Figura 3.6, a função $(\alpha \otimes R)(t)$ para $\alpha = 2$ mostra que este valor limita fortemente o fluxo de entrada R. O mesmo ocorre

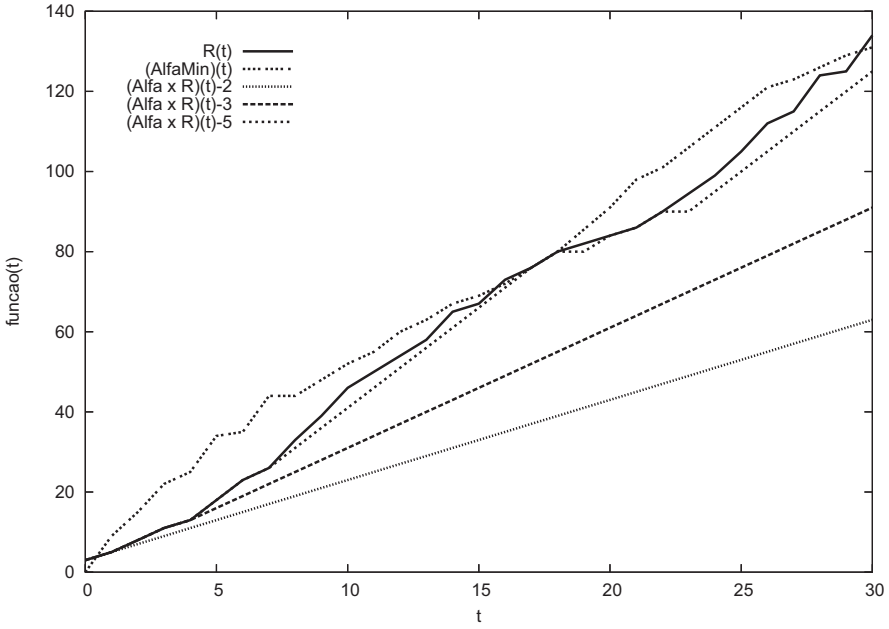


Figura 3.6: Função de entrada e curva de chegada mínima no sistema S, considerando α igual a 2, 3 e 5.

para $\alpha = 3$, mas para $\alpha = 5$ a função $(\alpha \otimes R)(t)$ é mais próxima do fluxo de entrada R. A função $(R \oslash R)(t)$ representa o valor mínimo de α para abrigar sem restrições o fluxo R.

A curva de serviço é uma abstração que define o modelo de serviço de um sistema. Os detalhes do escalonamento das unidades de serviço são simplificados utilizando o conceito de curva de serviço.

Considere um sistema S e um fluxo atravessando S com as funções de entrada e de saída $R(t)$ e $R^*(t)$. Pode-se dizer que S oferece para o fluxo uma curva de serviço β, se e somente se, para todo $t \geq 0$, existe algum $0 \leq s \leq t$, tal que

$$R^*(t) - R(s) \geq \beta(t - s).$$

A definição pode ser reconstruída, utilizando a convolução min-plus, conforme a Equação 3.6.

$$R^*(t) \geq \inf_{0 \leq s \leq t} \{\beta(t - s) + R(s)\} = (\beta \otimes R)(t) \tag{3.6}$$

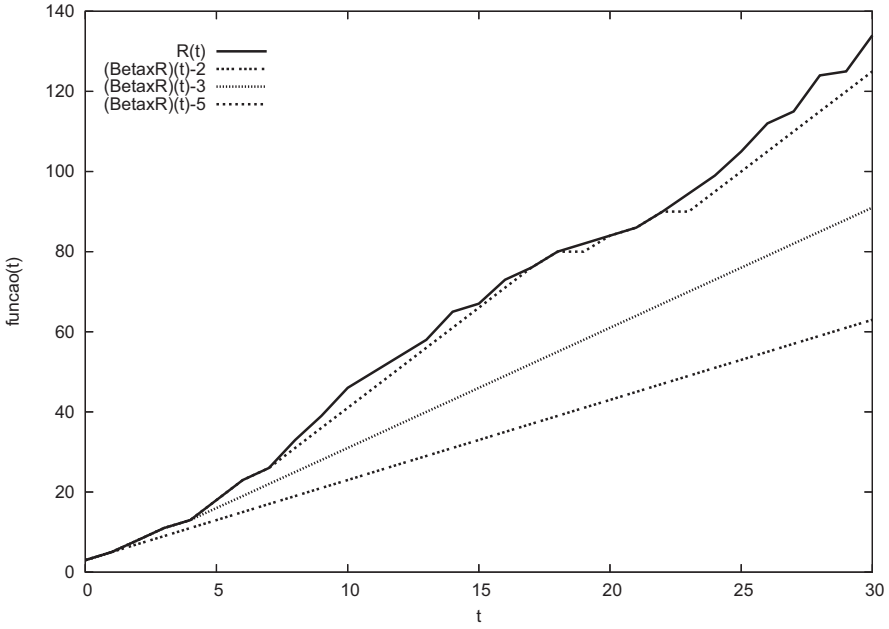


Figura 3.7: Funções de entrada e saída no sistema S, considerando β igual a 2, 3 e 5.

Considere um sistema S que foi observado por 30 unidades de tempo. A Figura 3.7 mostra a função de entrada $R(t)$ e a função $(\beta \otimes R)(t)$ para valores de β constantes e iguais a 2, 3 ou 5 unidades de serviço por unidade de tempo. Como pode ser observado na Figura 3.7, a função $(\beta \otimes R)(t)$ representa a função de saída R^* do sistema S, quando ele recebe a entrada R e opera a uma taxa de serviço β. A função $(\beta \otimes R)(t)$ para $\beta = 2$ mostra que este valor limita fortemente o fluxo de saída. O mesmo ocorre para $\beta = 3$, mas para $\beta = 5$ a função de saída é mais próxima do fluxo de entrada R.

3.6 Principais Resultados: os Três Limites

Os principais resultados da teoria Network Calculus são o limite do tamanho da fila, do atraso e do fluxo de saída. Os três limites estão descritos a seguir e se aplicam para sistemas sem perdas, mas também podem ser descritos para sistemas com perdas [27]. As provas são aplicações diretas das definições de curva de chegada e de curva de serviço e podem ser encontradas em [27].

Limite do tamanho da fila

Assuma um fluxo, limitado pela curva de chegada α, que atravessa um sistema que oferece uma curva de serviço β. O tamanho da fila $w(t)$, para todo t, satisfaz:

$$R(t) - R^*(t) \leq \sup_{s \geq 0}\{\alpha(s) - \beta(s)\}.$$

Isto significa que o tamanho da fila é função da taxa de chegada e da taxa de serviço. Se α e β são constantes e $\alpha = \beta$ ou $\beta > \alpha$, todas as tarefas serão atendidas na mesma taxa β e não haverá acúmulo de tarefas no sistema. Se $\alpha > \beta$, então algumas tarefas não conseguirão atendimento imediato e ficarão aguardando na fila, que estará limitada à maior distância vertical entre estas curvas.

Limite do atraso

Assuma um fluxo, limitado pela curva de chegada α, que atravessa um sistema que oferece uma curva de serviço β. O atraso $d(t)$, para todo t, satisfaz:

$$d(t) \leq h(\alpha, \beta).$$

O desvio horizontal h entre duas funções f e g de F é definido como:

$$h(f, g) = \sup_{s \geq 0}[\inf\{\tau \geq 0 \mid f(s) \leq g(s + \tau)\}].$$

Isto significa que o atraso é função da taxa de chegada e da taxa de serviço. Se α e β são constantes e $\alpha = \beta$ ou $\beta > \alpha$, todas as tarefas serão atendidas na mesma taxa β e não haverá atraso no sistema. Se $\alpha > \beta$, então algumas tarefas não conseguirão atendimento imediato e sofrerão atraso, que estará limitado a maior distância horizontal entre estas curvas.

Limite do fluxo de saída

Assuma um fluxo, limitado pela curva de chegada α, que atravessa um sistema que oferece uma curva de serviço β. O fluxo de saída é limitado pela curva de chegada α^*, onde:

$$\alpha^* = \alpha \oslash \beta.$$

Para ilustrar, considere um sistema S que foi observado por 30 unidades de tempo. Suponha que este sistema apresenta uma curva de chegada α com taxa constante e igual à curva de serviço β, conforme mostra a Figura 3.8 (a). O valor do limite do tamanho da fila e do atraso é igual a 0, pois não há acúmulo de tarefas no sistema nem atrasos. Se for considerado $\alpha < \beta$, conforme mostra a Figura 3.8 (b), o valor do limite do tamanho da fila e do atraso também é igual a 0, pois a taxa de serviço consegue atender satisfatoriamente todas as tarefas que chegam no sistema.

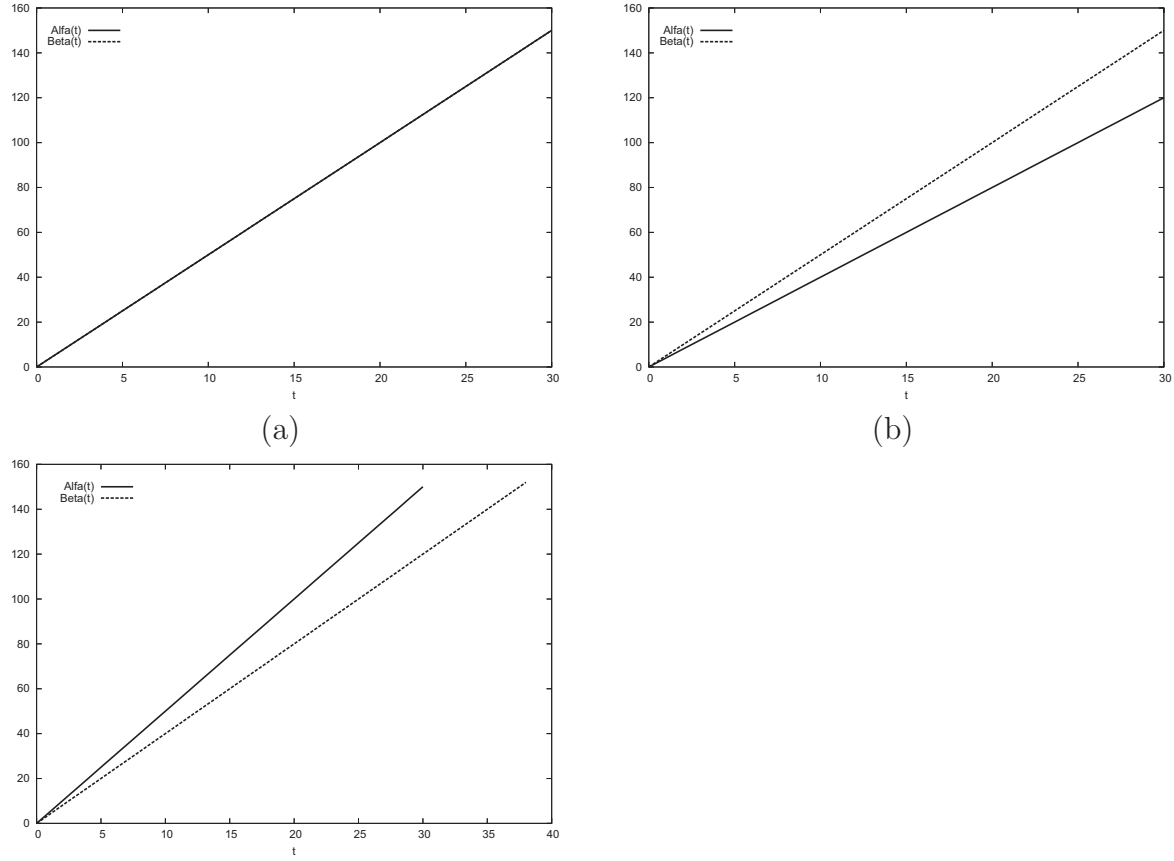


Figura 3.8: Curvas de chegada e de serviço no sistema S, considerando (a) $\alpha = \beta$, (b) $\alpha < \beta$ e (c) $\alpha > \beta$.

Suponha que este sistema apresenta uma curva de chegada com taxa constante igual a 5 e uma curva de serviço β com taxa constante igual a 4, conforme mostra a Figura 3.8 (c). Quando calculado o limite do tamanho da fila para este sistema no intervalo considerado, o valor de 30 unidades de serviço foi encontrado. O limite do atraso no intervalo é igual a 8 unidades de tempo.

A Figura 3.9 apresenta a curva α^* que limita o fluxo de saída, obtida pela operação $\alpha \oslash \beta$, considerando em (a) $\alpha = \beta = 4$, em (b) $\alpha = 4 < \beta = 5$ e em (c) $\alpha = 5 > \beta = 4$. Estes exemplos representam curvas simples de chegada e de serviço e são apenas ilustrativos. Na prática, as curvas são mais complexas e podem depender de parâmetros da carga e do sistema. No entanto, os resultados se aplicam da mesma forma. A ampla aplicabilidade da teoria e a exclusividade dos resultados providos demonstram sua robustez e importância.

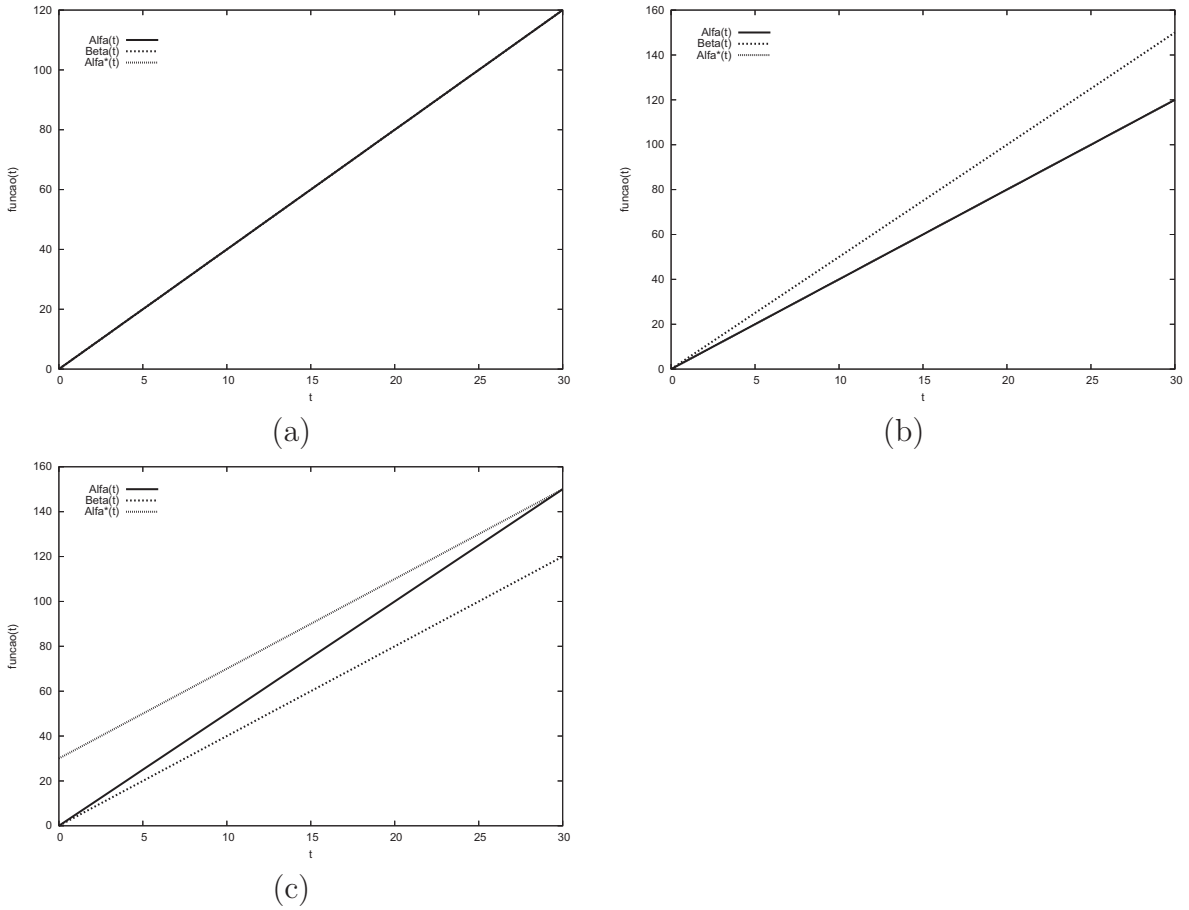


Figura 3.9: Curva α^* que limita o fluxo de saída do sistema S, considerando (a) $\alpha = \beta$, (b) $\alpha < \beta$ e (c) $\alpha > \beta$.

3.7 Considerações Finais

Este capítulo apresentou a teoria Network Calculus, utilizada para modelar o desempenho de um ou mais nós de uma rede de computadores. Esta teoria está fundamentada na álgebra min-plus e considera o sistema como uma caixa preta. Os fluxos de dados de entrada e de saída são vistos como funções cumulativas do número de objetos visto nos fluxos no intervalo $[0..t]$. A partir destas funções são derivados o tamanho da fila e o atraso virtual do sistema.

Network Calculus estabelece relações de entrada e saída a partir da premissa que o fluxo de entrada e o tempo de serviço satisfazem certas restrições. Estas restrições permitem que limites superiores no atraso de pacotes, no tamanho da fila do sistema e no fluxo de saída possam ser derivados.

Capítulo 4

Avaliação de Desempenho de Servidores Web com Network Calculus

O objetivo principal deste trabalho é aplicar os conceitos da teoria Network Calculus para a avaliação de desempenho de servidores Web. Para isto, o sistema S foi modelado como um servidor Web. O fluxo de entrada do servidor é constituído por requisições HTTP e o fluxo de saída é constituído de respostas a essas requisições. Informações registradas em *traces* reais de servidores Web foram utilizadas para a realização dos experimentos. Os fluxos de entrada e de saída foram compostos com base nestes registros. A partir dos fluxos, os resultados de desempenho do servidor, providos pela referida teoria, foram calculados.

Este capítulo apresenta um modelo determinístico para avaliação de desempenho de sistemas servidores Web, baseado nos conceitos fundamentais e nos resultados da teoria Network Calculus. Os resultados de desempenho do servidor, obtidos através do NC, são apresentados. Além disso, aspectos da implementação dos resultados da teoria também são discutidos. Para validar este modelo de desempenho, foi desenvolvido um simulador do servidor Web e implementado em AWK. Este simulador gerou resultados de desempenho, tais como tamanho da fila e atraso.

4.1 Considerações Iniciais

Considere um servidor Web S, o qual pode ser visto como uma caixa preta. S recebe requisições HTTP e as responde depois de um atraso variável. A saída do servidor S é representada pelas respostas a estas requisições. O servidor S é considerado sem perdas. Portanto, assume-se que todas as requisições são atendidas. Observe que na teoria NC o sistema é avaliado por meio de suas funções de entrada, de saída e de serviço, e não é relevante para a avaliação a política de escalonamento do servidor.

Para aplicação dos conceitos do Network Calculus foi utilizado um conjunto de dados consistindo de registros (*traces*) do servidor HTTP da Universidade da Califórnia - campus de Berkeley [18]. Para cada requisição foram registradas várias informações, sendo de interesse deste trabalho as seguintes: o tempo que o cliente fez a requisição (em segundos) e o tempo em que o primeiro byte foi visto na resposta do servidor (em segundos). Este conjunto de dados foi utilizado porque contém as informações necessárias para a geração das funções de entrada e de saída do servidor.

O *trace* Web foi coletado durante um período de 18 dias e armazenado em quatro arquivos codificados e compactados, para eficiência de armazenamento. Há também um conjunto de ferramentas que permite tratar e manipular os arquivos [18]. Neste trabalho foi utilizada uma parte do conjunto de dados que corresponde a um período de 4 horas de duração do dia 17 de novembro de 1996, das 16:47:06 até às 20:47:06, com um total de 95.768 requisições. Os resultados da aplicação do NC são apresentados para o primeiro minuto e para os 30 primeiros minutos registrados no *trace*. Experimentos com intervalos maiores foram realizados, por exemplo, 1 hora e 4 horas e o mesmo comportamento descrito para os primeiros minutos foi observado nos períodos mais longos.

Para conversão dos dados do *trace*, originalmente apresentados no formato binário, foi utilizada a ferramenta *showtrace* disponível em [18]. As informações sobre tempo de chegada da requisição e tempo de início da resposta, necessárias para o desenvolvimento deste trabalho, foram filtradas do *trace* e estavam no formato 848278028:829593. Os tempos foram convertidos para o formato DD/MM/AAAA HH:MM:SS pela ferramenta *timeconvert*, também disponível em [18]. Foi verificado que em algumas requisições a data

não pertencia ao intervalo de tempo indicado para o *trace*. Estas requisições representavam 2,85% do conjunto e foram descartadas. Utilizando um programa desenvolvido em JAVA, o conjunto de dados resultante foi formado de uma linha para cada requisição, onde a primeira coluna indicava o instante de tempo (em segundos) em que a requisição chegou no servidor e a segunda coluna indicava o instante de tempo (em segundos) em que a resposta à requisição foi iniciada.

4.2 Cálculo das Funções de Entrada e de Saída

Seja $R(t)$ a função cumulativa de entrada do servidor definida como o número de requisições HTTP observado em um intervalo de tempo arbitrário $[0..t]$. Considere $R^*(t)$ como sendo a função cumulativa de saída do servidor, representando as respostas às requisições.

Para tratar o conjunto de dados inicial, formado por duas colunas representando o instante de tempo de chegada da requisição e de sua resposta, foi desenvolvido um programa em AWK que gerou, para cada instante de tempo (ordenado), o número total de requisições que chegaram e o número total de requisições que saíram do servidor Web. Com estes dados, foi possível verificar a taxa de chegada de requisições no servidor e também a taxa de saída de respostas. A partir destes conjuntos de dados, foram gerados dois arquivos que representavam, respectivamente, $R(t)$ e $R^*(t)$, isto é, para cada instante de tempo, a partir do tempo 0, o número acumulado de requisições que chegaram e que saíram do servidor Web (as funções de entrada e de saída). A Figura 4.1 mostra as funções de entrada e de saída no servidor Web em intervalos de 1 minuto e de 30 minutos, respectivamente.

Conforme visto no capítulo anterior, a distância horizontal entre as curvas representa o atraso virtual e a distância vertical representa o tamanho da fila. Observe que para o período mais longo (30 minutos) as curvas ficam visualmente muito próximas, pois a distância entre elas se apresenta cada vez menor quando observada em um intervalo maior. Para períodos ainda mais longos de observação, as curvas se sobrepõem.

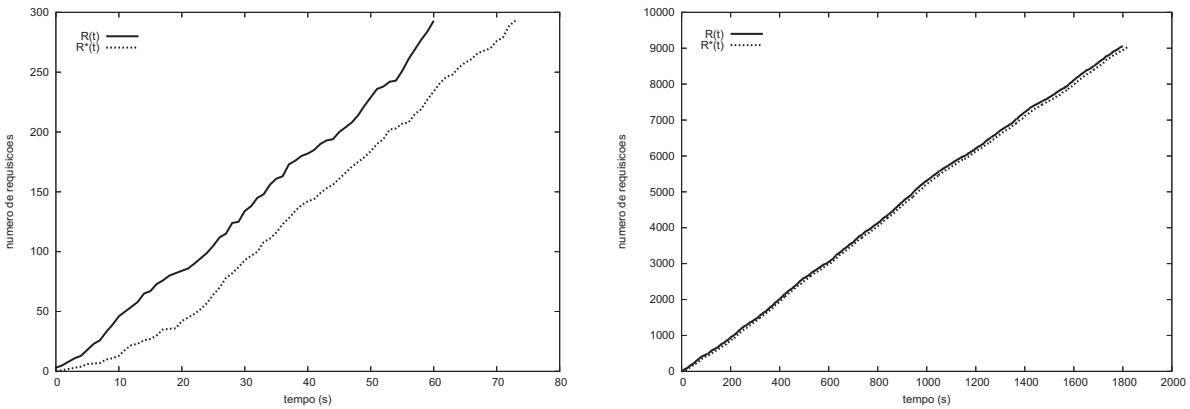


Figura 4.1: Funções cumulativas $R(t)$ e $R^*(t)$ no servidor Web em (a) 1 minuto e (b) 30 minutos de observação.

A Tabela 4.1 apresenta as estatísticas para a taxa de chegada de requisições no servidor no primeiro minuto e nos 30 primeiros minutos de observação. A Tabela 4.2 apresenta as estatísticas da taxa de saída para os mesmos intervalos. As taxas foram calculadas em períodos de um minuto. Por exemplo, chegaram, em média, 5,09 requisições por segundo em 30 minutos de observação, enquanto, nesse mesmo intervalo, foram finalizadas 5,02 requisições por segundo em média. Os valores para o coeficiente de variação indicam pouca variabilidade nas taxas de chegada e de saída no período observado.

Tabela 4.1: Estatísticas da taxa de chegada de requisições no servidor Web em 1 minuto e 30 minutos de observação.

ÍNDICE	TAXA DE CHEGADA (req/s) em 1 minuto	TAXA DE CHEGADA (req/s) em 30 minutos
Menor	1	1
Maior	10	15
Média	4,97	5,09
Mediana	4	5
Coef. de Variação	0,496	0,502
1^o quartil	3	3
3^o quartil	7	7

A estatística descritiva permite resumir uma grande quantidade de dados a partir de certas características que emergem do conjunto. As medidas de dispersão são medidas da variabilidade ou dispersão de um conjunto de dados em torno do seu valor central. Elas permitem identificar até que ponto os resultados se concentram ou não ao redor da

Tabela 4.2: Estatísticas da taxa de saída de requisições no servidor Web em 1 minuto e 30 minutos de observação.

ÍNDICE	TAXA DE SAÍDA (req/s) em 1 minuto	TAXA DE SAÍDA (req/s) em 30 minutos
Menor	1	1
Maior	10	16
Média	4,39	5,02
Mediana	4	5
Coef. de Variação	0,469	0,476
1^o quartil	3	3
3^o quartil	6	7

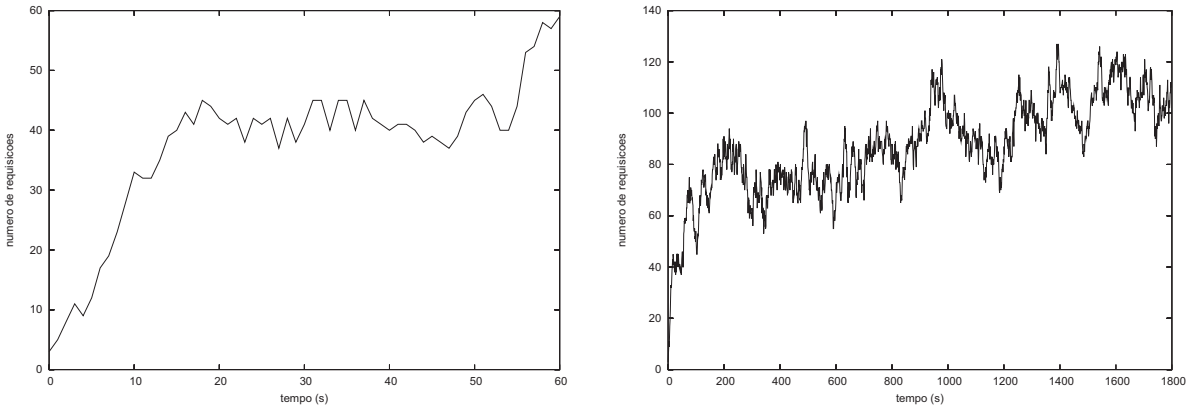
tendência central de um conjunto de observações. O coeficiente de variação (COV = $\frac{\sigma}{\mu}$), que é o desvio padrão dividido pela média, indica a variabilidade da amostra em relação à média. O desvio padrão (σ) é a raiz quadrada da variância, que é definida como o somatório do quadrado dos desvios (diferença entre i-ésimo valor e a média), dividido pelo número de elementos do conjunto. A variância (σ^2) caracteriza a dispersão dos pontos de uma amostra potencializando as diferenças.

Os fluxos de entrada e de saída observados no servidor em questão crescem a taxas aproximadamente constantes e não sofreram oscilações bruscas, o que pode ser verificado na Figura 4.1 e comprovado pelo baixo coeficiente de variação apresentado nas Tabelas 4.1 e 4.2.

Umas das vantagens do NC em relação às demais teorias de avaliação de desempenho é a apresentação dos fluxos de entrada e de saída como funções cumulativas. Assim, é possível visualizar e avaliar melhor o comportamento da chegada e da saída do sistema em relação às oscilações bruscas ocorridas tanto no processo de chegada como, por exemplo, chegadas em rajadas, quanto no processo de saída, devido a, por exemplo, ocorrências que possam degradar o desempenho do servidor. A suavização provida pelas curvas cumulativas produz este benefício.

4.3 Cálculo do Tamanho da Fila e do Atraso Virtual

Das funções cumulativas de entrada e de saída são derivadas duas métricas de interesse: tamanho da fila (*backlog*) e atraso virtual (*virtual delay*). O tamanho da fila no instante t é a quantidade de requisições observada dentro do sistema em cada instante. O atraso virtual no instante t é o atraso que pode ser sentido por uma requisição que chegou ao sistema neste instante, se todas as requisições recebidas anteriormente forem servidas antes desta.



(a) (b)

Figura 4.2: Tamanho da fila no servidor Web em (a) 1 minuto e (b) 30 minutos de observação.

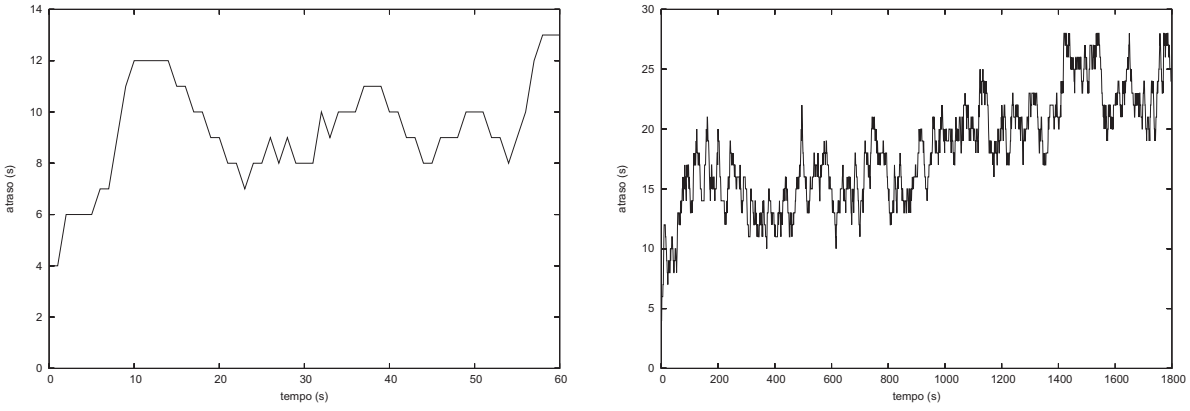
Tabela 4.3: Estatísticas do tamanho da fila em cada segundo no servidor Web em 1 minuto e 30 minutos de observação.

ÍNDICE	TAMANHO DA FILA (núm. de req.) em 1 minuto	TAMANHO DA FILA (núm. de req.) em 30 minutos
Menor	3	3
Maior	59	127
Média	37,52	87,06
Mediana	41	88
Coef. de Variação	0,326	0,213
1^o quartil	37	75
3^o quartil	44	101

O tamanho da fila no instante t é definido pela Equação 3.3. Para calcular o tamanho da fila, a função $w(t)$ foi implementada e executada com os dados de entrada do *trace*. A Figura 4.2 mostra, respectivamente, o tamanho da fila por segundo no servidor Web

observado por 1 minuto e por 30 minutos. É possível verificar que o tamanho da fila cresce lentamente durante os 30 minutos de observação. Os dados estatísticos do tamanho da fila são apresentados na Tabela 4.3. Os valores para o coeficiente de variação indicam pouca variabilidade no tamanho da fila nos períodos observados. É possível verificar que a média é maior para o intervalo de observação de 30 minutos.

O atraso virtual no instante t é definido pela Equação 3.4. Para calcular o atraso virtual, a função $d(t)$ foi implementada e executada com os dados de entrada do *trace*. A Figura 4.3 mostra, respectivamente, o atraso virtual no servidor Web observado por 1 minuto e por 30 minutos. Pode-se observar que o atraso virtual também é crescente no intervalo, conforme ocorreu com o tamanho da fila. Os dados estatísticos do atraso virtual nos intervalos considerados são apresentados na Tabela 4.4. Os valores para o coeficiente de variação indicam pouca variabilidade no atraso virtual nos períodos observados. É possível verificar que a média do atraso virtual é maior para o intervalo de observação de 30 minutos.



(a) (b)

Figura 4.3: Atraso virtual no servidor Web em (a) 1 minuto e (b) 30 minutos de observação.

A simulação do tamanho da fila realizada para validar este cálculo produziu exatamente os mesmos resultados apresentados na Tabela 4.3, comprovando a correção da implementação. A teoria Network Calculus considera o sistema como uma caixa preta e não faz restrições quanto à política de atendimento no servidor. Assim, o atraso virtual definido pelo NC é o atraso experimentado pela requisição que chega ao sistema no tempo t se todas as requisições que chegaram antes dela são servidas antes dela. Portanto, o

Tabela 4.4: Estatísticas do atraso virtual em cada segundo no servidor Web em 1 minuto e 30 minutos de observação.

ÍNDICE	ATRASO VIRTUAL (s) em 1 minuto	ATRASO VIRTUAL (s) em 30 minutos
Menor	4	4
Maior	13	28
Média	9,25	18,18
Mediana	9	18
Coef. de Variação	0,216	0,245
1^o quartil	8	15
3^o quartil	10,5	21

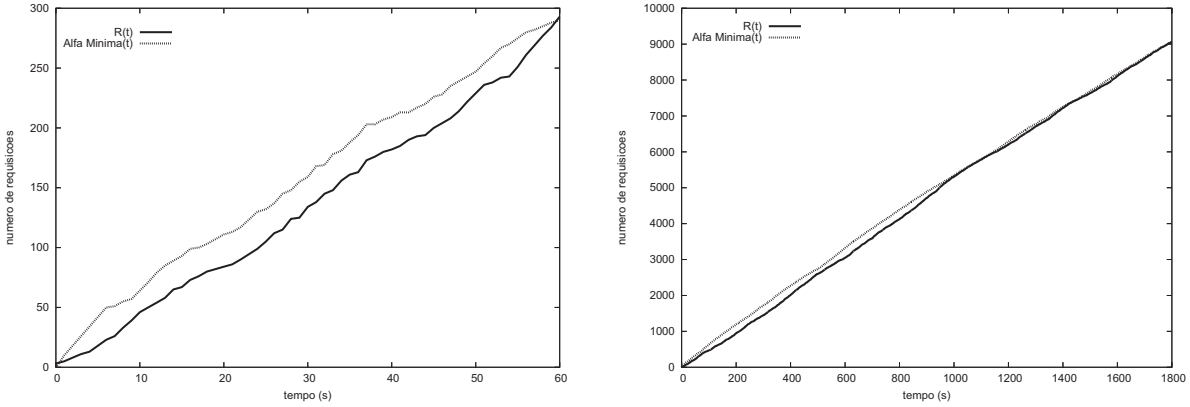
atraso obtido pelo simulador não produziu os mesmos resultados, pois considerou o atraso real de cada requisição.

4.4 Cálculo da Curva de Chegada Mínima

A função não-descrescente α, definida na Equação 3.5, é a curva de chegada para o fluxo. Este conceito é empregado na negociação entre clientes e administradores de redes que implementam mecanismos para prover qualidade de serviço. Para um servidor Web, não é comum restringir a taxa de chegada de requisições a um determinado valor máximo. No entanto, para implementação de mecanismos para prover qualidade de serviço, este tipo de acordo poderia ocorrer entre o servidor e seus clientes preferenciais. Neste caso, estes clientes se comprometeriam a enviar requisições a uma dada taxa máxima, que seria representada por α.

No caso do servidor Web utilizado neste experimento não há um valor conhecido de α, e acredita-se que, como na maioria dos servidores Web, não exista um limite predefinido para a curva de chegada. Quando não há conhecimento da curva α, pode-se encontrar a curva de chegada mínima através da função $R(t)$, aplicando a operação de deconvolução. Esta solução é comumente empregada quando $R(t)$ é obtido por medição [27]. A curva de chegada mínima é a função $(R \oslash R)(t)$, conforme operação definida na Equação 3.2.

A Figura 4.4 mostra a curva de chegada mínima no servidor Web observado por 1 minuto e por 30 minutos. A curva de chegada mínima representa a menor curva que pode abrigar o fluxo R.



(a) (b)

Figura 4.4: Curva de chegada mínima no servidor Web em (a) 1 minuto e (b) 30 minutos de observação.

4.5 Cálculo da Curva de Saída

Considere o servidor S e um fluxo atravessando S com as funções de entrada e de saída $R(t)$ e $R^*(t)$. Pode-se dizer que S oferece para o fluxo de entrada uma curva de serviço β, conforme definido na Equação 3.6.

Todo servidor tem sua curva de serviço β. A função que representa esta curva depende da implementação do servidor e pode ser complexa. Por exemplo, esta função pode ser dependente da carga e do tamanho da fila no servidor num determinado instante. A curva de serviço pode também ser utilizada no servidor para modelar o controle de admissão.

No caso do servidor Web utilizado neste experimento não há um valor conhecido de β. Então, foi assumido que o servidor tem uma taxa de serviço constante r, independente da carga. Assim, $\beta = rt$. Este é o modelo mais simples para β mas outros modelos para a curva de serviço podem ser utilizados.

Considerando $r = 4$ req/s, a Figura 4.5 mostra, respectivamente, a curva de saída no servidor Web observado por 1 minuto e por 30 minutos. O valor de r foi escolhido com base nos valores médios da taxa de saída apresentados na Tabela 4.2. Estes gráficos

apresentam três curvas: a função β, a curva de chegada $R(t)$ e a curva de saída, que é a função $(\beta \otimes R)(t)$. A saída do sistema é, portanto, a convolução da sua taxa de serviço β com a sua função de chegada $R(t)$, ou seja, a combinação das duas funções. $(\beta \otimes R)(t)$ é o limite de serviço do sistema, a sua função resposta.

Observe na Figura 4.5 (a) que a curva de saída é muito próxima da função β, mostrando que β limita a saída do sistema. Para um tempo maior de observação, conforme ilustrado na Figura 4.5 (b), as curvas estão sobrepostas. Observe também que o tamanho da fila (distância vertical entre as curvas) e o atraso virtual (distância horizontal) vão aumentando. Este comportamento é esperado para este valor de β, pois a taxa de chegada é maior que a taxa de serviço do sistema. Este resultado é coerente com a teoria e demonstra a correção da implementação das funções.

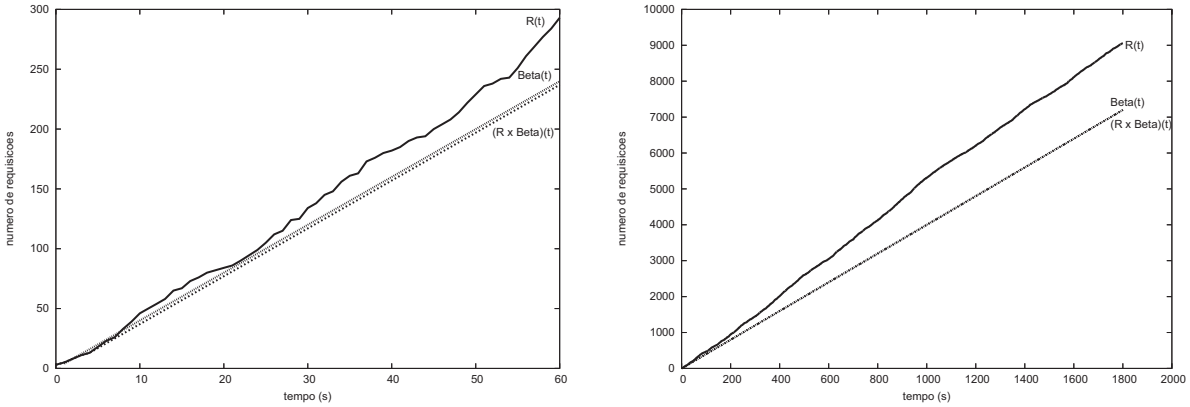


Figura 4.5: Curvas de entrada e de saída no servidor Web em (a) 1 minuto e (b) 30 minutos de observação, considerando a taxa de serviço de 4 req/s.

A curva de serviço β, as funções de entrada e de saída $R(t)$ e $R^*(t)$ e a convolução entre a função β e a função de entrada $R(t)$ estão ilustradas na Figura 4.6, quando a taxa de serviço para o sistema é $\beta = rt$, com $r = 5$ req/s. Essas curvas são praticamente indistingüíveis no intervalo maior, conforme mostra a Figura 4.6 (b), mas pode-se observar na Figura 4.6 (a) que a curva de saída real $R^*(t)$ é inferior a $(\beta \otimes R)(t)$. Isto indica que a curva β escolhida pode não ser representativa do sistema real. A curva de serviço pode ser, por exemplo, dependente da carga, ou do tipo $\beta = a+rt$, onde a é uma constante. Modelar curvas de serviço de servidores reais a partir deste cenário é uma tarefa potencialmente

relevante e, embora não seja objetivo deste trabalho, é uma das sugestões para a sua continuação.

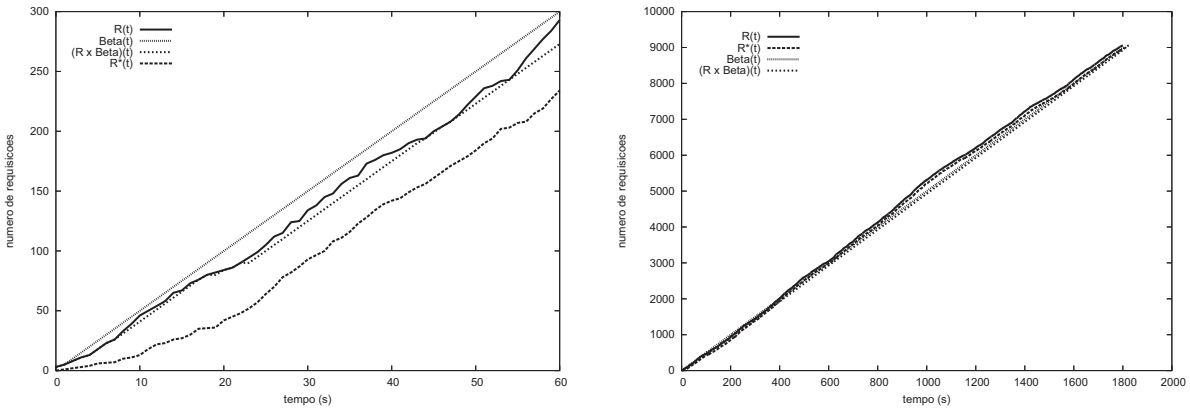


Figura 4.6: Curvas de entrada e de saída no servidor Web em (a) 1 minuto e (b) 30 minutos de observação, considerando a taxa de serviço de 5 req/s.

4.6 Cálculo dos Três Limites

Os três limites são calculados a partir das curvas α e β, conforme descrito na seção 3.6. Considerando que as funções α e β não são conhecidas para o servidor Web em estudo, inicialmente foi suposto que as funções de entrada e de saída $R(t)$ e $R^*(t)$ representavam as curvas de chegada e de serviço do servidor.

Os valores obtidos para o limite do tamanho da fila e para o limite do atraso virtual foram exatamente os maiores valores para as respectivas métricas apresentadas nas Tabelas 4.3 e 4.4. Este resultado mostra a correção da implementação das funções do NC e das operações da álgebra min-plus, pois o limite superior do tamanho da fila é a maior distância vertical entre as curvas α e β, agora sendo consideradas como $R(t)$ e $R^*(t)$. O limite superior do atraso virtual é a maior distância horizontal entre as curvas α e β, agora também sendo consideradas como $R(t)$ e $R^*(t)$. O limite do tamanho da fila e do atraso virtual estão apresentados na Tabela 4.5. O terceiro resultado do Network Calculus é o limite do fluxo de saída. A Figura 4.7 apresenta um limite para o fluxo de saída dado, neste caso, por $\alpha^* = R \oslash R^*$, para um minuto e 30 minutos de observação.

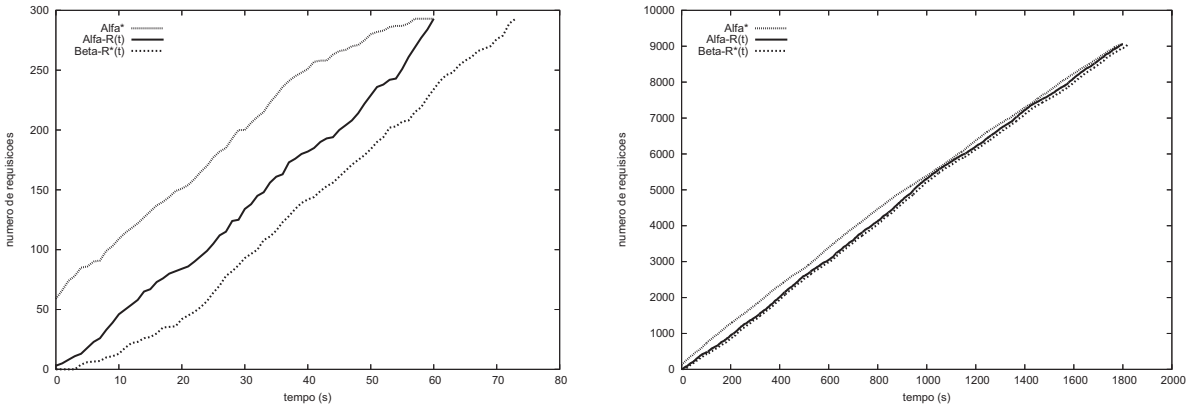


Figura 4.7: Curva α^* em (a) 1 minuto e (b) 30 minutos de observação, considerando α como $R(t)$ e β como $R^*(t)$.

Tabela 4.5: Limites do tamanho da fila e do atraso virtual para 1 minuto e 30 minutos, para os 3 casos de suposição de α e β.

CASOS	$\alpha = R$ $\beta = R^*$		$\alpha = R \oslash R$ $\beta = 5$ req/s		$\alpha = R$ $\beta = 5$ req/s	
INTERVALOS	1 min.	30 min.	1 min.	30 min.	1 min.	30 min.
Limite do tamanho da fila	59	127	20	393	3	320
Limite do atraso virtual	13	28	4	79	1	64

Outro experimento foi realizado para calcular os três limites, onde a curva de chegada mínima definida por $(R \oslash R)(t)$ foi utilizada como α e a curva de serviço empregada foi $\beta = rt$, onde r igual a 5 req/s. O limite do tamanho da fila e do atraso virtual estão apresentados na Tabela 4.5. O terceiro resultado do Network Calculus, que é o limite do fluxo de saída, está apresentado na Figura 4.8, para os intervalos de 1 minuto e de 30 minutos.

Um terceiro experimento foi realizado para calcular os três limites, onde a curva de entrada $R(t)$ foi utilizada como α e a curva de serviço empregada foi $\beta = rt$, onde r é igual a 5 req/s. O limite do tamanho da fila e do atraso virtual estão apresentados na Tabela 4.5. A Figura 4.9 (a) mostra que a taxa de serviço β sempre é maior que a taxa de chegada α, representada pela curva R, exceto pelo primeiro segundo de observação, onde chegaram 3 requisições no tempo 0, e as saídas só começaram no tempo 1. Por isso, nesta situação, o tamanho máximo da fila é de 3 requisições e o atraso virtual é de 1 s. O

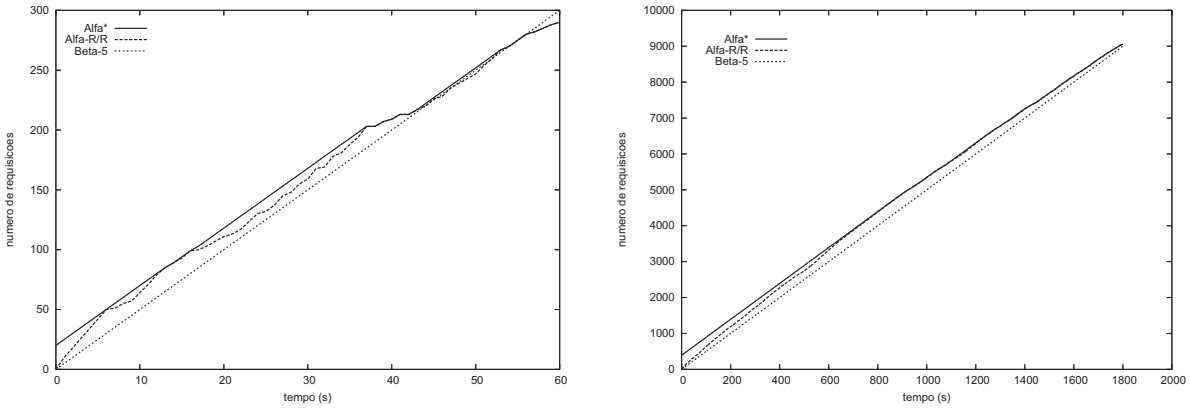


Figura 4.8: Curva α^* em (a) 1 minuto e (b) 30 minutos de observação, considerando α como α mínima e β com taxa de serviço de 5 req/s.

terceiro resultado da teoria, que é o limite do fluxo de saída, está apresentado na Figura 4.9, para os intervalos de 1 minuto e de 30 minutos.

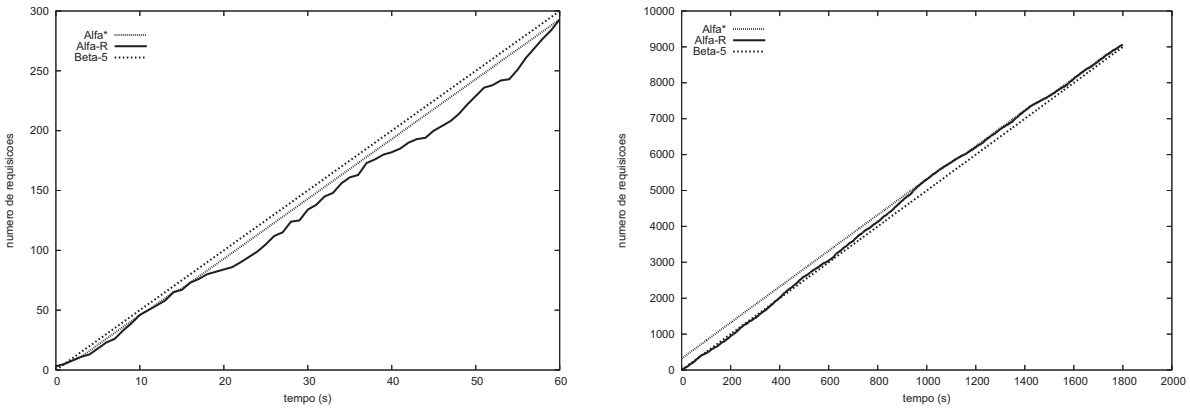


Figura 4.9: Curva α^* em (a) 1 minuto e (b) 30 minutos de observação, considerando α como $R(t)$ e β com taxa de serviço de 5 req/s.

4.7 Aspectos da Implementação do Network Calculus

Nesta seção são descritas as duas principais etapas da implementação do Network Calculus, a saber, a composição dos fluxos de entrada e de saída e a implementação de funções para cálculo dos principais resultados da teoria.

4.7.1 Composição dos Fluxos

O conjunto de dados resultante do *trace* foi formado em duas colunas, onde a primeira indicava o instante de tempo (em segundos) em que a requisição chegou no servidor e a segunda indicava o instante de tempo (em segundos) que resposta à requisição foi iniciada. A partir deste conjunto de dados e utilizando programas em AWK, foram gerados dois arquivos que representavam, respectivamente, $R(t)$ e $R^*(t)$, isto é, para cada instante de tempo, a partir do tempo 0, o número acumulado de requisições que chegaram e saíram do servidor Web (as funções de entrada e saída).

4.7.2 Implementação das Equações

As funções providas pela teoria Network Calculus foram implementadas em rotinas AWK, respeitando as equações apresentadas no Capítulo 3.

A Figura 4.10 apresenta o código utilizado para calcular o tamanho da fila no sistema para cada instante t, dado por $w(t) = R(t) - R^*(t)$. Esta rotina recebe como parâmetros o intervalo de tempo para o qual o tamanho da fila deve ser calculado (`ini` e `fim`), bem como os arquivos que contêm as funções de entrada e de saída $R(t)$ e $R^*(t)$. Como em um determinado instante de tempo o sistema pode ter ficado sem receber requisições e/ou ter ficado sem produzir respostas às requisições, o código repete o valor da função cumulativa até que, para um determinado instante de tempo, um novo valor seja obtido dos dados de entrada.

A Figura 4.11 apresenta o código utilizado para calcular o atraso virtual para cada instante t. O atraso virtual é definido como $d(t) = \texttt{inf}\{\tau \geq 0 \mid R(t) \leq R^*(t+\tau)\}$. Esta rotina recebe como parâmetros o intervalo de tempo para o qual o atraso deve ser calculado (`ini` e `fim`), bem como os arquivos que contêm as funções de entrada e de saída $R(t)$ e $R^*(t)$. Como em um determinado instante de tempo o sistema pode ter ficado sem receber requisições e/ou ter ficado sem produzir respostas às requisições, o código repete o valor da função cumulativa até que, para um determinado instante de tempo, um novo valor seja obtido dos dados de entrada.

```
END{
     valorx = 0 ;
     valory = 0 ;
     for( t = ini ; t <= fim ; t++ )
     {
          if( rtx[ t ] == -1 )
               rtx[ t ] = valorx ;
          else
               valorx = rtx[ t ] ;

          if( rty[ t ] == -1 )
               rty[ t ] = valory ;
          else
               valory = rty[ t ] ;

          print t, rtx[ t ] - rty[ t ]
     }
}
```

Figura 4.10: Código para cálculo do tamanho da fila.

Outro aspecto a considerar é que o sistema é observado durante um intervalo de tempo $[0..t]$. Na prática foi verificado que, para o cálculo do atraso virtual, é necessário conhecer a função $R^*(t)$ até o instante de tempo em que todas as respostas são produzidas, ou seja, até o instante de tempo em que número de respostas vistas no sistema é igual ao número de requisições que chegaram no sistema. Este conhecimento da função $R^*(t)$, além do instante t, é necessário para o cálculo do atraso virtual no intervalo $[0..t]$.

Uma dificuldade encontrada foi identificar o limite de τ para o cálculo do atraso virtual. Foi utilizado como limite o valor em que o instante $(t + \tau)$ não pertencia ao intervalo de observação, ou seja, onde a função $R^*(t)$ não estaria definida. Este valor também foi fornecido como parâmetro para a rotina na variável `fimy`.

A Figura 4.12 apresenta o código utilizado para calcular a convolução entre duas funções não decrescentes f e g e a Figura 4.13 apresenta o código para calcular a deconvolução entre duas funções. Estas rotinas recebem como parâmetros o intervalo de tempo para o qual a operação deve ser calculada (`ini` e `fim`), bem como os arquivos que contêm as funções f e g. Também houve dificuldade em identificar o limite de u para a operação de deconvolução, definida na Equação 3.2, e necessário ao cálculo da curva de chegada mínima $(R \oslash R)(t)$ e também da curva $\alpha^* = \alpha \oslash \beta$ (limite do fluxo de saída). Foi utilizado como limite o instante de tempo onde as funções envolvidas no cálculo estão definidas.

```
END{
     # completa Rt? com valor do Rt anterior,
     # qdo no t não está definido
     valorx = 0 ;
     valory = 0 ;
     for( t = ini ; t <= fimy ; t++ )
     {
          if( rtx[ t ] == -1 )
               rtx[ t ] = valorx ;
          else
               valorx = rtx[ t ] ;

          if( rty[ t ] == -1 )
               rty[ t ] = valory ;
          else
               valory = rty[ t ] ;
     }

     for( t = ini ; t <= fim ; t++ )
     {
          tau = 0
          menor = fim
          while( tau <= fim && (t + tau) <= fimy )
          {
               if( rtx[ t ] <= rty[ t + tau ] )
                    if( tau <= menor )
                         menor = tau ;
               tau++
          }
          print t, menor
     }
}
```

Figura 4.11: Código para cálculo do atraso virtual.

A complexidade computacional das operações é outro ponto a considerar. Se for analisado, por exemplo, o cálculo do limite do atraso (definido na seção 3.6), onde, para cada instante t do intervalo de observação, o valor de s deve variar de 0 até t e, para cada s, o valor de τ deve ser variado de 0 até $(s + \tau) \leq t$, tem-se um algoritmo com complexidade $O(n^3)$, onde n é o número de amostras no intervalo $[0..t]$. A análise detalhada da complexidade das operações é fundamental para a avaliação das possibilidades de aplicação da teoria nos diversos contextos, seja para aplicações em tempo real ou não [17]. A Figura 4.14 apresenta o código utilizado para calcular o limite do atraso virtual. Esta rotina recebe como parâmetros o intervalo de tempo para o qual o limite deve ser calculado (`ini` e `fim`), bem como os arquivos que contêm as funções α e β.

```
END{
     valorx = 0 ;
     valory = 0 ;
     for( i = ini ; i <= fim ; i++ )
     {
          if( f[ i ] == -1 )
               f[ i ] = valorx ;
          else
               valorx = f[ i ] ;

          if( g[ i ] == -1 )
               g[ i ] = valory ;
          else
               valory = g[ i ] ;
     }

     for( t = ini ; t <= fim ; t++ )
     {
          s = 0 ;
          menor = f[ t - s ] + g[ s ] ;
          while( s <= t && (t - s) <= fim )
          {
               if( (f[ t - s ] + g[ s ]) < menor )
                    menor = f[ t - s ] + g[ s ] ;
               s++ ;
          }
          print t, menor ;
     }
}
```

Figura 4.12: Código para cálculo da operação $(f \otimes g)(t)$.

4.8 Considerações Finais

Neste capítulo foi apresentada a modelagem e a avaliação de desempenho de servidores Web utilizando a teoria Network Calculus. Esta teoria permite o cálculo de limites para métricas importantes de desempenho como tamanho da fila e atraso virtual no servidor Web, bem como apresenta como resultado o limite do fluxo de saída do sistema. Os resultados podem ser facilmente obtidos a partir da curva de chegada α e da curva de serviço β aplicando as equações que modelam o sistema. Além disso, é possível identificar curvas de serviço para os servidores e expressar seus limites. Os resultados de desempenho obtidos foram validados através de simulação.

A aplicação da teoria demonstrada neste capítulo é baseada em registro de um sistema real. No entanto, é importante ressaltar que a teoria pode modelar servidores mesmo em fase de projeto, uma vez conhecidas ou pelo menos supostas suas curvas de chegada e de serviço. A caracterização de curvas de chegada e de serviço de sistemas reais, para posterior utilização em modelagem de sistemas, é uma proposta para trabalhos futuros.

```
END{
     valorx = 0 ;
     valory = 0 ;
     for( t = ini ; t <= fim ; t++ )
     {
          if( f[ t ] == -1 )
               f[ t ] = valorx ;
          else
               valorx = f[ t ] ;

          if( g[ t ] == -1 )
               g[ t ] = valory ;
          else
               valory = g[ t ] ;
     }

     for( t = ini ; t <= fim ; t++ )
     {
          u = 0 ;
          maior = f[ t + u ] - g[ u ] ;
          while( (t+u) <= fim && u <= fim )
          {
               if( (f[ t + u ] - g[ u ]) > maior )
                    maior = f[ t + u ] - g[ u ] ;
               u++ ;
          }
          print t, maior ;
     }
}
```

Figura 4.13: Código para cálculo da operação $(f \oslash g)(t)$.

```
. . .
END{
     . . .
     for( t = ini ; t <= fim ; t++ )
     {
          maior = -1 ;
          for( s = 0 ; s <= fim ; s++ )
          {
               tau = 0 ;
               menor = fimy ;
               while( (s + tau) <= fimy )
               {
                    if( a[ s ] <= b[ s + tau ] )
                         if( tau < menor )
                              menor = tau ;
                    tau++ ;
               }
               if( menor > maior )
                    maior = menor ;
          }
          print t, maior ;
     }
}
```

Figura 4.14: Código para cálculo do limite do atraso virtual.

Capítulo 5

Outros Experimentos e Aplicações da Teoria Network Calculus em Servidores Web

A partir do modelo de desempenho de servidores Web obtido com a aplicação da teoria Network Calculus, apresentado no capítulo 4, foi realizada uma análise da viabilidade de seu emprego no processo de avaliação de desempenho destes servidores. Inicialmente, foi realizada uma análise comparativa dos resultados de desempenho da teoria Network Calculus com os resultados oferecidos pela Lei de Little. O resultado desta comparação é apresentado neste capítulo.

Este capítulo também mostra um estudo sobre o desempenho dos servidores Web da Copa do Mundo de 1998, utilizando o modelo proposto no capítulo 4. Os resultados desta avaliação são apresentados. Foi verificado que estes servidores Web enfrentaram aumentos repentinos na taxa de chegada de requisições. Assim, um estudo sobre controle de admissão de tarefas no sistema foi realizado, empregando os resultados de desempenho providos pela teoria Network Calculus.

5.1 Comparação com a Lei de Little

No capítulo 2, na seção sobre técnicas para avaliação e modelagem de desempenho de sistemas computacionais, foi apresentada a análise operacional como uma técnica analítica determinística. A análise operacional consiste de leis operacionais e a lei operacional mais conhecida é a lei de Little, que é de particular interesse para este trabalho.

A lei de Little estabelece a relação entre o número médio de tarefas presentes em um dado sistema com o tempo médio de resposta do sistema, $E[n] = \lambda \times E[r]$. Este relacionamento pode ser aplicado a qualquer sistema onde o número de tarefas que chegam ao sistema é igual ao número de serviços completados [21].

A teoria Network Calculus fornece a métrica tamanho da fila, número de tarefas presentes no sistema, a partir das funções cumulativas de entrada e de saída, respectivamente $R(t)$ e $R^*(t)$. O tamanho da fila, definido como $w(t) = R(t) - R^*(t)$, é a distância vertical entre as duas funções, para cada instante de tempo t, dentro do intervalo de observação.

As métricas $E[n]$ e $w(t)$ apresentam semelhanças. Assim, foi realizado um estudo para comparar o resultado provido pela lei de Little e pela teoria Network Calculus.

5.1.1 Derivação da Lei de Little

Devido à semelhança na definição da lei de Little e do tamanho da fila no NC, a idéia foi derivar a lei de Little a partir do conceito de funções de entrada e de saída, definidas na teoria NC. Estas funções são necessárias para o cálculo do tamanho da fila $w(t)$.

Considere um gráfico de operação do sistema que apresenta o número total de tarefas que chegam e que são finalizadas no sistema no tempo, conforme ilustra a Figura 5.1. Cada degrau na função de chegada representa a ocorrência da chegada de uma ou mais tarefas naquele instante e cada degrau na função de tarefas finalizadas representa a saída de uma ou mais tarefas [23]. Em qualquer instante, a distância vertical entre as funções representa o número de tarefas presente no sistema. Em qualquer intervalo de tempo, a área entre as funções representa o tempo acumulado no sistema durante o intervalo, medido em tarefas `x` tempo. Por exemplo, a área preenchida da Figura 5.1 representa

2 tarefas no sistema durante um período de 2 segundos, o que indica que o sistema gastou um total de 4 segundos de execução para estas 2 tarefas neste intervalo. O tempo acumulado no sistema será denotado por J. Assim, o número médio de tarefas no sistema é $N = \frac{J}{T} = \frac{4}{2} = 2$.

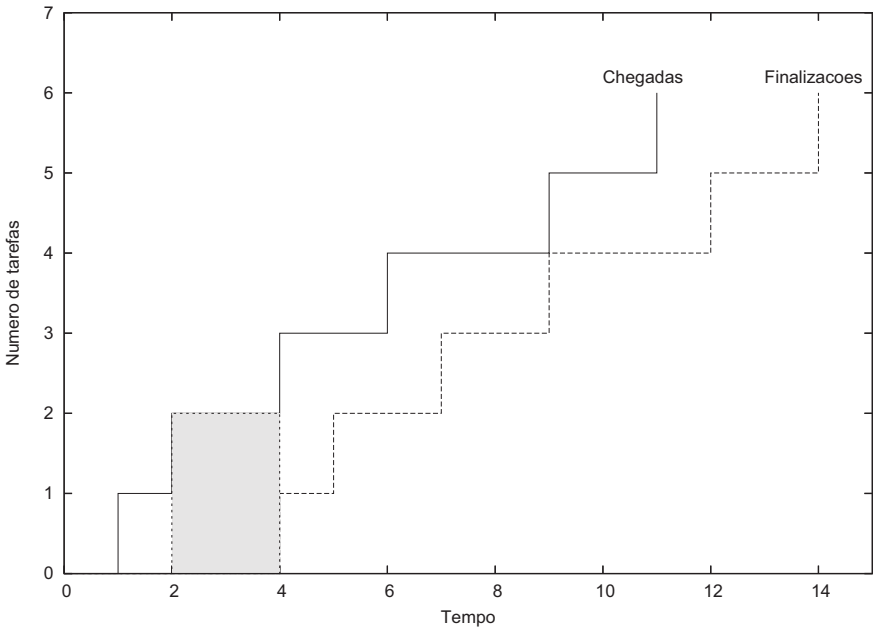


Figura 5.1: Chegadas e finalizações do sistema.

5.1.2 Resultados da Comparação com a Lei de Little

A apresentação da lei de Little através das funções de chegada e de tarefas finalizadas, sugerida em [23], representa exatamente as funções de entrada e de saída do sistema, respectivamente $R(t)$ e $R^*(t)$, definidas pela teoria Network Calculus. Isto pode ser verificado pela semelhança entre os gráficos das Figuras 3.4 e 5.1.

A comparação da lei de Little com os resultados obtidos pela teoria Network Calculus serve para validar o tamanho da fila no NC. Entretanto, o próprio resultado da lei de Little não é discutido no NC. Uma diferença importante é que a lei de Little provê valores médios para os resultados de desempenho, NC busca limites de pior caso.

Para ilustrar esta comparação, considere um servidor Web que foi monitorado por um intervalo de tempo T e que um registro dos tempos de chegada e de resposta de cada requisição foi obtido. Seja N o número de requisições neste intervalo. Se um gráfico for

construído mostrando a quantidade de tarefas no sistema para cada instante de tempo t, a área representará o total de tarefas que enfrentaram fila no sistema. Seja J esta área. O número médio de tarefas no sistema pode ser definido como $E[n] = J/T$. Para o *trace* de exemplo, utilizado no Capítulo 4 [18], a Tabela 5.1 mostra os resultados com a aplicação da Lei de Little. Estes resultados são idênticos àqueles obtidos com a aplicação da teoria Network Calculus para cálculo do tamanho da fila, conforme a média apresentada na Tabela 5.2, que é a mesma tabela apresentada na seção 4.3.

Tabela 5.1: Tamanho da fila no servidor Web em 1 minuto e 30 minutos de observação, obtido a partir da Lei de Little.

ÍNDICE	1 minuto	30 minutos
J	2289	156807
T	61	1801
E[n]	**37,52**	**87,06**

Tabela 5.2: Estatísticas do tamanho da fila em cada segundo no servidor Web em 1 minuto e 30 minutos de observação.

ÍNDICE	TAMANHO DA FILA (núm. de req.) em 1 minuto	TAMANHO DA FILA (núm. de req.) em 30 minutos
Menor	3	3
Maior	59	127
Média	**37,52**	**87,06**
Mediana	41	88

Se um gráfico for construído mostrando para cada requisição o tempo total gasto no servidor, a área representará o tempo total gasto no sistema por todas as tarefas. Porém, o atraso médio por tarefa calculado a partir da área do gráfico, dado por $E[r] = \frac{J}{N}$, não corresponde ao atraso virtual dado pelo NC. Na verdade, são métricas distintas. O atraso médio por tarefa é uma métrica que representa o tempo médio de permanência no sistema por tarefa do conjunto de tarefas atendido. O atraso virtual $d(t)$ definido na teoria NC é função do tempo de chegada e indica o tempo que uma tarefa que chegou no tempo t pode permanecer no sistema.

5.2 Experimento com Carga Intensa

O objetivo desta seção é apresentar os resultados do NC para um *trace* de servidores Web no qual foi registrada uma carga bastante intensa. Analisar os modelos de desempenho em condições extremas é útil para verificar limites de sua aplicação.

O *trace* do sítio da Copa do Mundo de 1998 foi utilizado nos experimentos por estar disponível na Internet e por apresentar uma carga alta com picos de acesso ao servidor. A idéia inicial foi verificar o comportamento dos resultados de desempenho de servidores Web quando a teoria NC fosse aplicada, considerando uma carga com variabilidade no processo de chegada.

O artigo [5] apresenta um estudo sobre a caracterização da carga no sítio Web da Copa do Mundo de 1998. Os registros de acesso a este sítio Web foram coletados durante um período de três meses. Durante este período de tempo, o sítio Web recebeu mais de 1,35 bilhões de requisições.

A caracterização da carga de trabalho tem um papel importante no projeto de sistemas e também no projeto de novos componentes de sistemas. Além disso, a carga extremamente alta do sítio Web da Copa do Mundo de 1998 permitiu prever o comportamento de cargas futuras de servidores Web e assim ser possível realizar um planejamento de acordo.

5.2.1 Considerações Iniciais sobre o *Trace*

Para aplicar o modelo NC de avaliação de desempenho em servidores Web, foi utilizado um conjunto de dados consistindo de registros (*traces*) dos servidores HTTP do sítio da Copa do Mundo de 1998 [6]. Foram utilizados 30 servidores durante o campeonato, distribuídos em 4 localidades: França e Estados Unidos (Virginia, Texas e Califórnia). Um conjunto de balanceadores de carga foi utilizado para distribuir as requisições entre estas quatro localidades e entre os servidores de cada localização. O *trace* disponível em [6] é a carga completa, sem distinção de servidor ou localização. Neste trabalho o sítio Web foi considerado como o sistema S definido pela teoria, isto é, uma caixa preta.

O *trace* Web foi coletado entre 30 de abril de 1998 e 26 de julho de 1998. Durante este período de tempo o sítio Web recebeu 1.352.804.107 requisições. Os arquivos de dados de todos os servidores foram combinados em uma única seqüência de requisições, ordenada pelo *timestamp* de cada requisição. Por causa do volume de dados, o arquivo (codificado) foi separado em intervalos que representam um dia. Para manter o tamanho do arquivo do *trace* abaixo de 50 MB, alguns arquivos foram divididos em subintervalos. Foram gerados 249 arquivos codificados durante os 92 dias de coleta de dados.

Neste experimento foi utilizada uma parte do *trace*, que representa o conjunto de dados dos dias 24, 25 e 26 de junho de 1998, com duração de quase 35 horas. Este período foi escolhido por apresentar variabilidade no processo de chegada, de acordo com o objetivo deste estudo. Para cada requisição foram registradas várias informações, sendo de interesse deste trabalho o tempo de chegada da requisição (em segundos) no servidor. Um arquivo contendo o instante de tempo (ordenado) e o número total de requisições que chegaram no servidor Web foi gerado. Com estes dados, foi possível verificar a taxa de chegada de requisições no sistema, apresentada na Figura 5.2. É possível verificar dois instantes de pico na taxa de chegada de requisições no sistema. A Tabela 5.3 apresenta as estatísticas para a taxa de chegada de requisições no sistema. A mediana indica que em 50% do tempo de observação chegaram até 282 req/s e o maior valor indica 2669 requisições por segundo, representando o maior pico da rajada.

Tabela 5.3: Estatísticas da taxa de chegada de requisições no sistema.

ÍNDICE	TAXA DE CHEGADA (req/s)
Menor	14
Maior	2669
Média	394,7
Mediana	282
Coef. de Variação	0,727
1^o quartil	240
3^o quartil	407

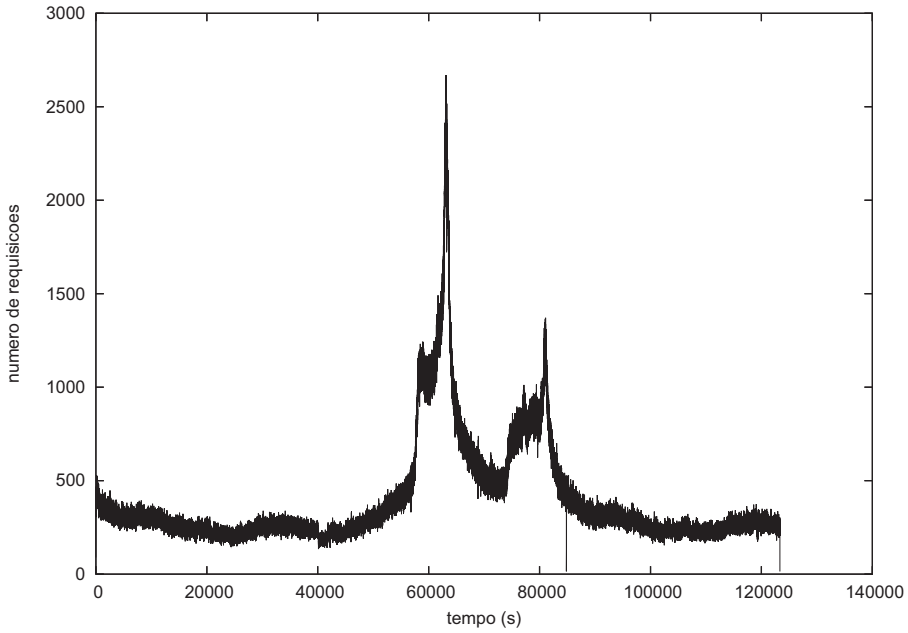


Figura 5.2: Taxa de chegada de requisições no sistema.

5.2.2 Cálculo da Curva de Saída

Com a informação do tempo de chegada da requisição no sistema, disponível no *trace*, foi gerada a função $R(t)$ correspondente à função de entrada. A Figura 5.3 mostra a curva $R(t)$ para o sistema no período observado. Pode-se verificar duas inclinações mais fortes na curva $R(t)$ (a primeira mais acentuada), nos mesmos instantes em que ocorreram os picos de chegada.

No *trace* do sítio Web da Copa do Mundo de 1998 não há registro do tempo de resposta da requisição. Assim, a função $R^*(t)$ não é conhecida. No caso do servidor Web utilizado neste experimento não há um valor conhecido de β. Então, foi assumido que o servidor tem uma taxa de serviço constante r, independente da carga. Assim, foi definido $\beta = rt$, com $r = 400$ req/s. Este valor foi escolhido com base na taxa de chegada média de requisições. A seguir, foi calculada a função $(\beta \otimes R)(t)$, que representará a função de saída do sistema $R^*(t)$ quando a curva de serviço é β, conforme apresentada na Figura 5.3.

Observe na Figura 5.3 que as distâncias vertical e horizontal entre as curvas $R(t)$ e $R^*(t)$ vão aumentando a partir do instante onde ocorre o primeiro pico na taxa de

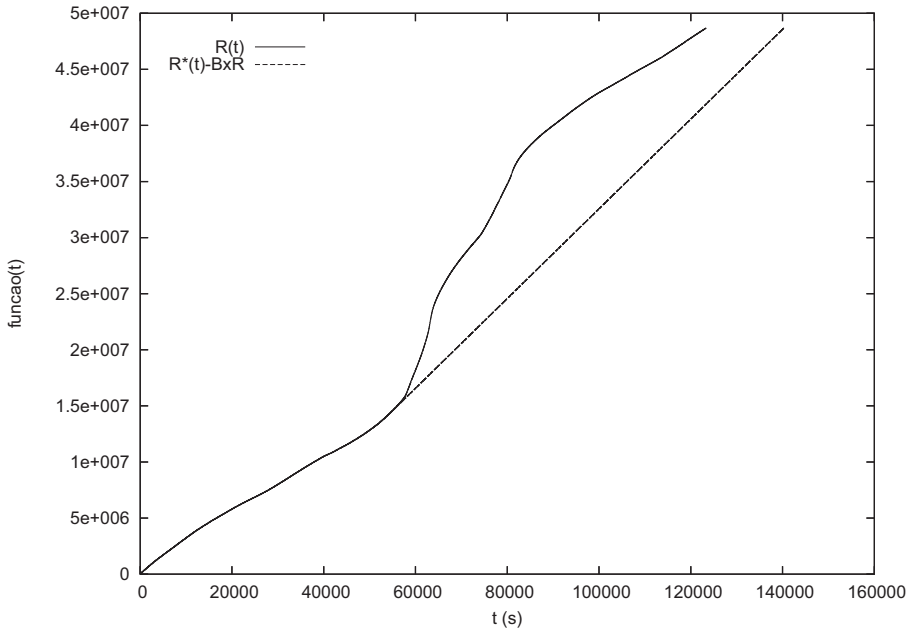


Figura 5.3: $R(t)$ e $R^*(t)$ para o sistema, considerando a taxa de serviço de 400 req/s.

chegada. Este comportamento é esperado para este valor de β, pois a taxa de chegada neste instante é maior que a taxa de serviço no sistema.

5.2.3 Cálculo do Tamanho da Fila e do Atraso Virtual

Considerando as curvas de chegada e de saída apresentadas na subseção 5.2.2, o tamanho da fila no sistema para o intervalo de observação está mostrado na Figura 5.4. É possível verificar que o tamanho da fila cresce rapidamente durante os instantes da rajada. Este comportamento é esperado, pois a taxa de chegada é bem maior que a taxa de serviço do sistema nesses instantes. Os dados estatísticos do tamanho da fila estão apresentados na Tabela 5.4. É possível verificar que, em grande parte do tempo, não houve requisição acumulada no sistema. Este comportamento também é esperado, pois a taxa de serviço é capaz de atender a taxa de chegada de requisições em grande parte do tempo, evitando que as requisições ficassem aguardando na fila.

O atraso virtual no sistema para o intervalo de observação está mostrado na Figura 5.5. Para este cálculo foi utilizada a curva de saída definida na subseção 5.2.2. É possível verificar que o atraso virtual aumenta rapidamente durante os instantes das rajadas. Este comportamento é esperado pois a taxa de chegada é bem maior que a taxa de serviço

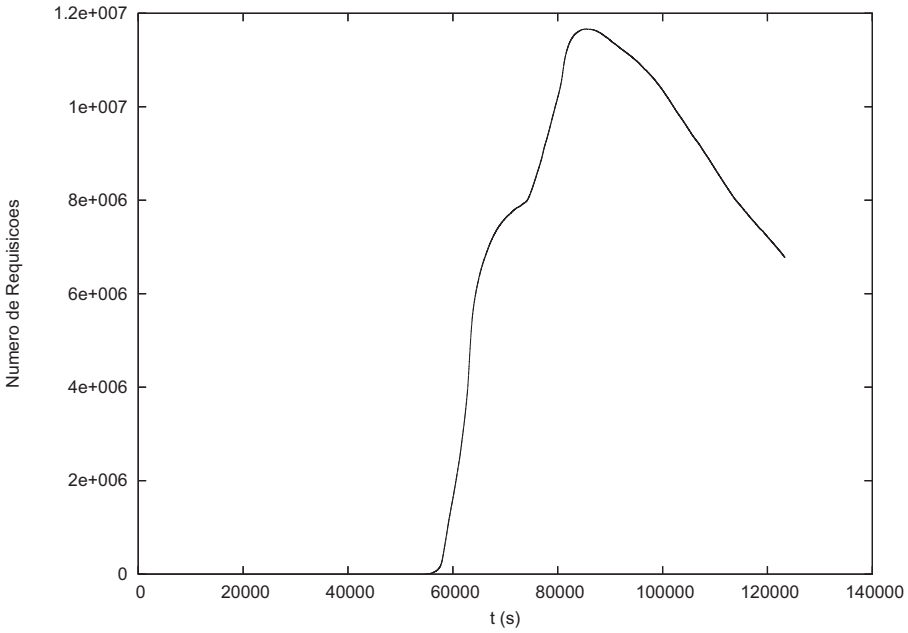


Figura 5.4: Tamanho da fila no sistema, considerando a taxa de serviço de 400 req/s.

Tabela 5.4: Estatísticas do tamanho da fila no sistema.

ÍNDICE	TAMANHO DA FILA (núm. req.)
Menor	0
Maior	11.663.367
Média	4.566.960
Mediana	2.792.736
Coef. de Variação	1,025
1^o quartil	0
3^o quartil	9.013.773

do sistema nesses instantes. A sobrecarga no sistema gerou atrasos maiores. Os dados estatísticos do atraso virtual estão apresentados na Tabela 5.5. É possível verificar que em grande parte do tempo não houve atraso do sistema. Este comportamento também é esperado pois a taxa de serviço é capaz de atender a taxa de chegada de requisições em grande parte do tempo, evitando que as requisições ficassem aguardando na fila e por conseqüência evitando um aumento no tempo de resposta.

5.2.4 Cálculo dos Três Limites

Para calcular os três limites, são necessárias a curva de chegada α e a curva de serviço β. Estes valores não são conhecidos para o servidor Web em estudo. Para este cálculo

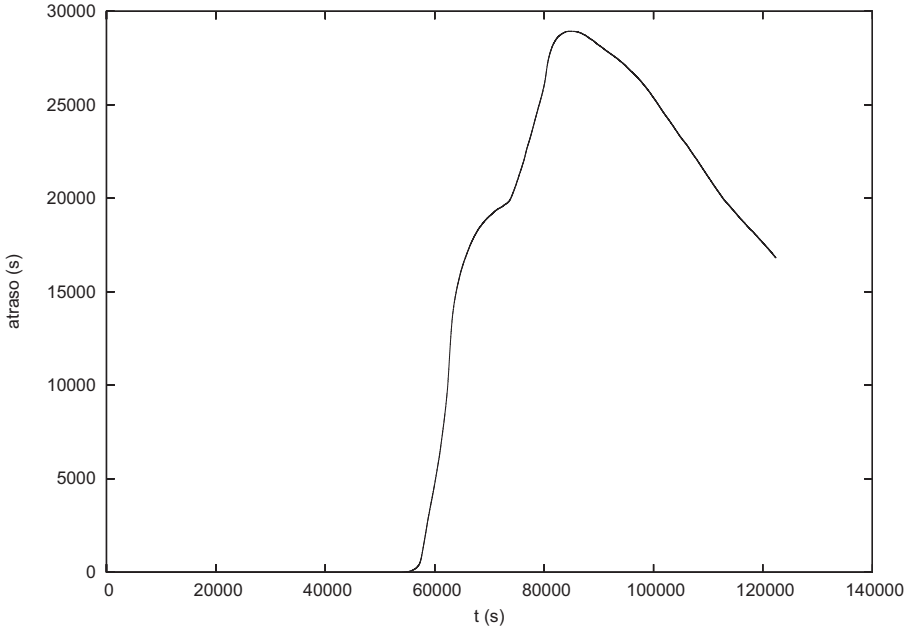


Figura 5.5: Atraso virtual no sistema, considerando a taxa de serviço de 400 req/s.

Tabela 5.5: Estatísticas do atraso virtual no sistema.

ÍNDICE	ATRASO VIRTUAL (s)
Menor	0
Maior	29.159
Média	11.417,7
Mediana	6.982
Coef. de Variação	1,025
1^o quartil	0
3^o quartil	22.535

foi considerada $\alpha(t) = (R \oslash R)(t)$, isto é, a curva de chegada mínima, conforme está apresentada na Figura 5.6 (a). A curva de serviço considerada foi $\beta(t) = rt$, com r igual a 400 req/s. A Figura 5.6 (b) apresenta a curva $R(t)$ experimental e as curvas $R^*(t)$, $\beta(t)$ e $\alpha(t)$, considerando esta situação.

O limite do tamanho da fila encontrado foi de 11.663.367 requisições, que é igual ao máximo do tamanho da fila. O limite do atraso virtual foi de 29.159 segundos, que é igual ao máximo do atraso virtual no intervalo de observação. O limite do fluxo de saída está apresentado na Figura 5.6 (c), que mostra que o fluxo de saída está limitado pela curva α^*.

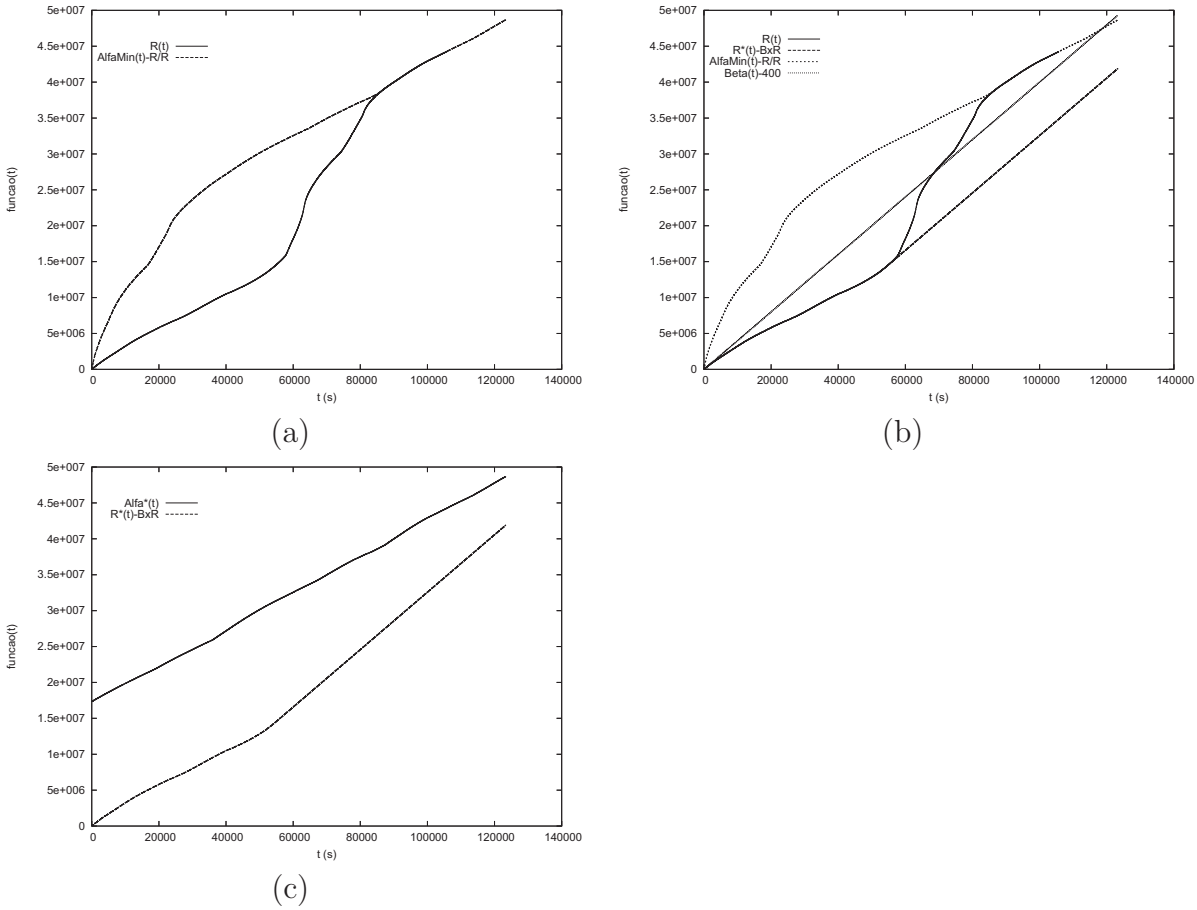


Figura 5.6: (a) Curva de chegada mínima no sistema, (b) Curvas R experimental, R^*, α mínima e β, considerando a taxa de serviço de 400 req/s, (c) Limite no fluxo de saída no sistema.

5.3 Controle de Admissão com a Teoria Network Calculus

Servidores Web estão sujeitos a variações de demanda, o que pode levá-los a uma situação de sobrecarga. Rajadas ou grande variabilidade do processo de chegada podem degradar o desempenho e a taxa de processamento, se não forem levadas em consideração [2].

Para evitar a sobrecarga no sistema e conseqüente a degradação no desempenho, a idéia foi simular um mecanismo para controle de sobrecarga em servidores Web, aplicando os conceitos da teoria Network Calculus. Esta teoria modela o comportamento de desempenho de um sistema medindo seus fluxos de entrada e de saída. Esta abordagem permite caracterizar sobrecargas em servidores e configurar parâmetros que ajudem no controle e detecção de sobrecarga.

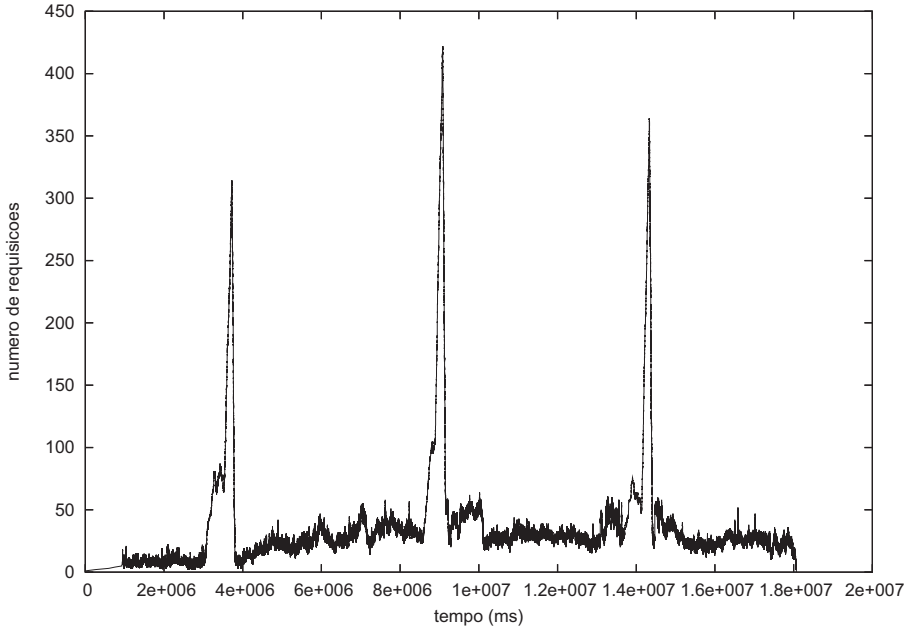


Figura 5.7: Tamanho da fila como função do processo de chegada.

Para exemplificar, suponha um servidor Web observado por um intervalo de tempo. A Figura 5.7 mostra o tamanho da fila, calculado como o número de requisições no servidor no instante de tempo que cada requisição chegou no sistema. Este cálculo é feito a partir do registro do tempo de entrada e do tempo de permanência de cada requisição no servidor. O tamanho da fila médio é de 31,9 requisições durante o intervalo de observação e a mediana é 26. Este gráfico mostra três picos na carga, revelando que o servidor experimentou tamanhos de filas maiores que 300 requisições (cerca de dez vezes a média). O maior tamanho de fila é de 421 requisições.

Dado este cenário, suponha que o servidor Web esteja sobrecarregado durante os períodos de pico na carga. A idéia foi verificar a relação entre o tamanho da fila e a taxa de serviço do sistema. A Figura 5.8 mostra a taxa de serviço média para todas as requisições que experimentaram o mesmo tamanho de fila. É possível verificar que a taxa de serviço é estável quando o tamanho da fila é menor que aproximadamente 60. O sistema parece estar em operação normal, produzindo uma taxa de serviço média aproximadamente constante, quando o tamanho da fila é menor que 60. O comportamento do sistema é mais instável e variado quando o tamanho da fila é maior que 60. Para o tamanho da fila muito grande, o desempenho pode ser comprometido.

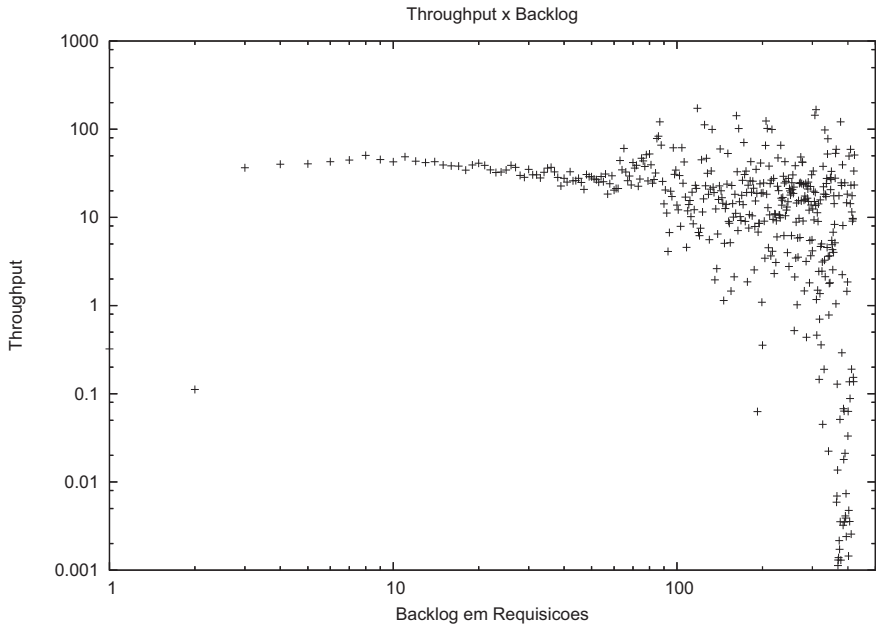


Figura 5.8: Relação entre o tamanho da fila e a taxa média de serviço do sistema.

Para verificar a freqüência da sobrecarga, a Figura 5.9 mostra a função de distribuição acumulada (CDF) para o tamanho da fila. Em 96% do tempo o tamanho da fila é menor que 60.

Assim, observando o número de requisições que chegam no servidor e as suas respostas, e utilizando as funções $R(t)$ e $R^*(t)$, é possível identificar as rajadas que o sistema pode receber e aquelas que degradarão seu desempenho. Para este último caso, um alarme pode ser implementado para indicar a necessidade de controle de admissão ou de outros mecanismos para controle de sobrecarga.

O tráfego enviado pelas origens para o servidor Web precisa ser limitado para não sobrecarregar o sistema e, conseqüentemente, diminuir seu desempenho. O controle de admissão é modelado no NC pela curva α. Um fluxo de entrada deve ser menor ou igual a $(\alpha \otimes R)(t)$.

5.4 Considerações Finais

Neste capítulo foi apresentada uma comparação da lei de Little com o resultado do tamanho da fila definido pela teoria Network Calculus. O valor do número médio de tarefas no sistema obtido pela lei de Little é o mesmo quando comparado com o valor

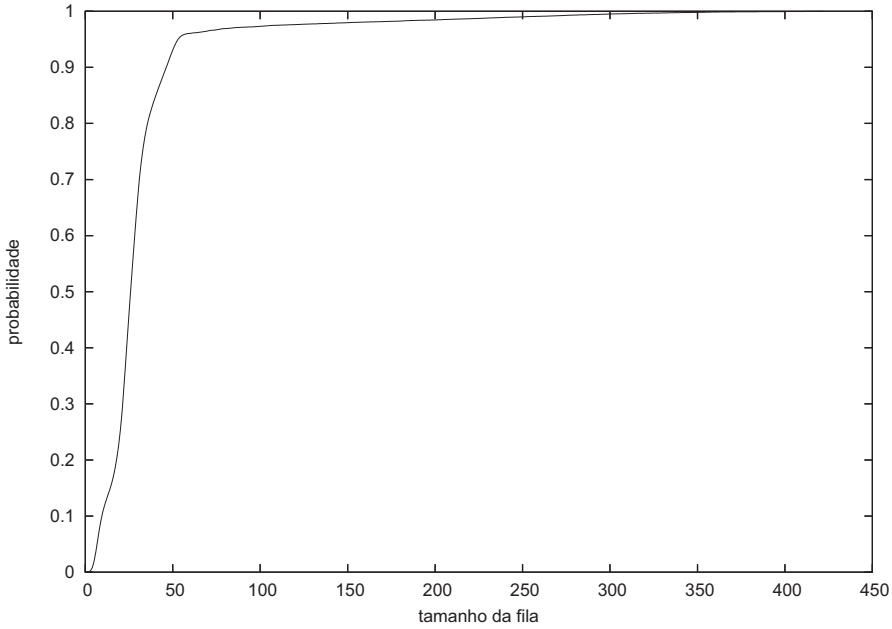


Figura 5.9: CDF do tamanho da fila.

médio do tamanho da fila definido pela teoria NC. No entanto, os objetivos dos dois modelos são diferentes. Enquanto que a lei de Little provê valores médios para os resultados de desempenho, NC busca limites de pior caso.

Umas das vantagens da teoria Network Calculus em relação às demais teorias de avaliação de desempenho é a apresentação dos fluxos de entrada e de saída como funções cumulativas. Assim, é possível visualizar e avaliar melhor o comportamento da chegada no sistema em relação às oscilações bruscas. Isto foi verificado no experimento com o *trace* da Copa do Mundo de 1998.

A teoria Network Calculus pode ser empregada para revelar sobrecarga em servidores Web. É possível identificar as rajadas que degradarão o desempenho ao acompanhar os fluxos de entrada e de saída. A teoria provê a definição da curva de chegada α para modelar o controle de admissão. Assim, um mecanismo para controle de admissão pode ser ativado e seus parâmetros podem ser modificados dinamicamente.

Capítulo 6

Conclusões

Network Calculus é uma teoria que permite a modelagem de sistemas de filas e define limites máximos determinísticos para algumas métricas de desempenho, quando observadas certas restrições no fluxo de entrada. Este trabalho mostrou a aplicação desta teoria para modelar o desempenho de servidores Web. O fluxo de entrada no servidor é constituído por requisições HTTP e o fluxo de saída é constituído de respostas a essas requisições. Registros de acesso de servidores Web reais foram utilizados como dados de entrada. Para obter os resultados, curvas de chegada e de serviço foram simuladas, tendo como base as informações dos registros de acesso. É importante ressaltar que embora a aplicação do Network Calculus tenha sido feita com os registros de acesso, é possível aplicar a teoria da mesma forma para qualquer função conhecida de entrada. Devido a isso, é possível modelar e prever o desempenho futuro do sistema, uma vez conhecidos os parâmetros de entrada do modelo.

Não é conhecida nenhuma teoria de sistemas de filas que ofereça como resultados limites superiores determinísticos de desempenho. Por isso foi considerado que esta teoria é importante e pode ser bastante útil para modelar o desempenho dos sistemas. Os modelos probabilísticos da teoria de filas tradicional provêem resultados probabilísticos, apresentados como distribuições de probabilidade, assim como devem ser os parâmetros de entrada. Os modelos determinísticos estabelecem relações simples entre quantidades de desempenho diretamente mensuráveis em um sistema computacional e provêem valores

médios para os resultados. A utilização de funções cumulativas em toda a teoria constitui um aspecto interessante, pois assim é possível visualizar melhor o comportamento dos fluxos de chegada e de saída do sistema em relação a oscilações bruscas ocorridas tanto no processo de chegada quanto no processo de saída.

Este trabalho apresenta algumas contribuições. Inicialmente, ele apresenta uma aplicação prática da teoria Network Calculus. Não foi encontrado nenhum trabalho que descreva qualquer aplicação prática desta teoria, nem trabalhos que aplicam o Network Calculus para analisar desempenho de servidores Web. Como o primeiro trabalho nesta área, maior ênfase foi dada à interpretação da teoria no contexto de servidores Web e à implementação e demonstração da correção das funções implementadas. A validação foi feita de algumas formas, incluindo a comparação com a Lei de Little e vários experimentos e análises manuais de resultados. A aplicação da teoria em servidores com carga intensa foi também mostrada. A utilização da teoria para modelar o controle de admissão foi discutida. A teoria Network Calculus é bastante adequada para modelar controle de admissão, pois define a curva de chegada com este objetivo.

Diversos prosseguimentos podem ser identificados para este trabalho. A partir do contexto e das definições da teoria, seria bastante interessante caracterizar curvas de chegada e de serviço de servidores Web reais, o que contribuiria para a construção de modelos novos e mais genéricos. Em particular, identificar curvas de serviço características de servidores Web pode auxiliar a modelagem destes sistemas. A dependência da curva de serviço em função da carga pode também ser analisada. Todo servidor tem sua curva de serviço. A função que representa esta curva depende da implementação do servidor e pode ser complexa. Esta função pode, por exemplo, ser dependente da carga, do tamanho da fila no servidor e da política de escalonamento.

Outro caminho natural é a modelagem de controle de admissão empregando esta teoria e a implementação de uma ferramenta de monitoração e previsão de desempenho de servidores Web baseada nas funções e nos resultados do Network Calculus. Diversas

curvas de chegada podem ser testadas em conjunto com as curvas de serviço para obter melhores limites de desempenho. A análise detalhada da complexidade computacional das operações implementadas é importante para a implementação proposta.

Referências Bibliográficas

[1] AGRAWAL, R.; CRUZ, R. L.; OKINO, C.; RAJAN, R. **Performance Bounds for Flow Control Protocols**. *IEEE/ACM Transactions on Networking*, volume 7, pp. 310–323, junho de 1999.

[2] ALMEIDA, V.; ARLITT, M.; ROLIA, J. **Analyzing a Web-Based System's Performance Measures at Multiple Time Scales**. *ACM SIGMETRICS Performance Evaluation Review*, 30(2):3–9, 2002.

[3] ALMEIDA, VIRGÍLIO; BESTAVROS, AZER; CROVELLA, MARK; OLIVEIRA, ADRIANA DE. **Characterizing reference locality in the WWW**. *Proceedings of the IEEE Conference on Parallel and Distributed Information Systems)*, Miami Beach, FL, 1996.

[4] ARLITT, M. F.; WILLIAMSON, C. L. **Web Server Workload Characterization: The Search for Invariants**. *Measurement and Modeling of Computer Systems*, pp. 126–137, 1996.

[5] ARLITT, M.; JIN, T. **Workload Characterization of the 1998 World Cup Web Site**. *IEEE Network*, volume 14, pp. 30–37, maio/junho de 2000.

[6] ARLITT, M.; JIN, T. **1998 World Cup Web Site Access Logs**, agosto de 1998. Acessado em 2004, disponível em `http://www.acm.org/sigcomm/ITA/`.

[7] BACCELLI, F.; COHEN, G.; OLSDER, G.; QUADRAT, J-P. ***Synchronization and Linearity: An Algebra for Discrete Event Systems***. Wiley: versão on-line do livro ISBN 0 471 93609 X, 2001. Acessado em 2004. Disponível em `http://www-rocq.inria.fr/metalau/cohen/SED/book-online.html`.

[8] BANGA, G.; DRUSCHEL, P. **Measuring the Capacity of a Web Server**. *USENIX Symposium on Internet Technologies and Systems*, 1997.

[9] BARFORD, PAUL; CROVELLA, MARK. **Generating Representative Web Workloads for Network and Server Performance Evaluation**. *Measurement and Modeling of Computer Systems*, pp. 151–160, 1998.

[10] CHANG, CHENG-SHANG. **On Deterministic Traffic Regulation and Service Guarantees: a systematic approach by filtering**. *IEEE/ACM Transactions on Information Theory*, volume 44, pp. 1097–1110, maio de 1998.

[11] CHANG, CHENG-SHANG. ***Performance Guarantees in Communication Networks***. Springer-Verlag, 2000.

[12] CHEN, H.; MOHAPATRA, P. **Session-Based Overload Control in QoS-aware Web Servers**, In *Proceedings of IEEE INFOCOM. New York*, junho de 2002.

[13] CHERKASOVA, LUDMILA; FU, YUN; TANG, WENTING; VAHDAT, AMIN. **Measuring and Characterizing End-to-End Internet Service Performance**. *ACM Transactions on Internet Technology*, 3(4):347–391, 2003.

[14] CROVELLA, MARK; BESTAVROS, AZER. **Self-Similarity in World Wide Web Traffic: Evidence and Possible Causes**. *IEEE/ACM Transactions on Networking*, 5(6):835–846, 1997.

[15] CRUZ, R. L. **A Calculus for Network Delay, Part I: Network Elements in Isolation**. *IEEE Transactions on Information Theory*, volume 37, pp. 114–131, janeiro de 1991.

[16] CRUZ, R. L. **A Calculus for Network Delay, Part II: Network Analysis**. *IEEE Transactions on Information Theory*, volume 37, pp. 132–141, janeiro de 1991.

[17] FIROIU, V.; LE BOUDEC, J.-Y.; TOWSLEY, D.; ZHANG, Z.-L. **Theories and Models for Internet Quality of Service**. *Proceedings of the IEEE*, volume 90, pp. 1565–1591, setembro de 2002.

[18] GRIBBLE, STEVEN D. **The Internet Traffic Archive**, julho de 1997. Acessado em 2004, disponível em `http://www.acm.org/sigcomm/ITA/`.

[19] INTERNET WORLD STATS. **Internet Usage Statistics - The Big Picture**, dezembro de 2004. Acessado em 2005, disponível em `http://www.internetworldstats.com/stats.htm`.

[20] IYER, R.; TEWARI, V.; KANT, K. **Overload Control Mechanisms for Web Servers**, In *Workshop on Performance and QoS of Next Generation Networks. Nagoya, Japan*, novembro de 2000.

[21] JAIN, RAJ. ***The Art Of Computer Systems Performance Analysis***. John Wiley & Sons Inc, 1991.

[22] KLEINROCK, L. ***Queuing Systems.*** *Vol. I: Theory.* John Wiley & Sons Inc, 1975.

[23] LAZOWSKA, E. D.; ZAHORJAN, J.; GRAHAM, G. S.; SEVCIK, K. C. ***Quantitative System Performance: Computer System Analysis Using Queueing Network Models***. Prentice Hall, 1984. Acessado em 2004. Disponível em `http://www.cs.washington.edu/homes/lazowska/qsp/`.

[24] LE BOUDEC, J. Y. **Application of Network Calculus to Guaranteed Service Networks**. *IEEE/ACM Transactions on Information Theory*, volume 44, pp. 1087–1097, maio de 1998.

[25] LE BOUDEC, J. Y.; CHARNY, A. **Packet Scale Rate Guarantee for Non-FIFO Nodes**. *IEEE/ACM Transactions on Networking*, volume 11, pp. 810–820, outubro de 2003.

[26] LE BOUDEC, J. Y.; THIRAN, P. **A Short Tutorial on Network Calculus I: Fundamental Bounds in Communication Networks**. *ISCAS'2000*, maio de 2000.

[27] LE BOUDEC, J. Y.; THIRAN, P. ***Network Calculus: A Theory of Deterministic Queuing Systems for the Internet***. Versão on-line do livro Springer Verlag - LNCS 2050, 2001. Acessado em 2004. Disponível em `http://ica1www.epfl.ch/PS_files/NetCal.htm`.

[28] MENASCÉ, DANIEL A.; ALMEIDA, VIRGÍLIO A. F. ***Planejamento de Capacidade para Serviços na Web: Métricas, modelos e métodos***. Editora Campus, 2002.

[29] MENASCÉ, DANIEL A.; ALMEIDA, VIRGÍLIO A. F.; DOWDY, LARRY W. ***Performance by Design: Computer Capacity Planning By Example***. Prentice Hall, 2004.

[30] MINDCRAFT INC. **WebStone: The Benchmark for Web Servers**, março de 2002. Acessado em 2004, disponível em `http://www.mindcraft.com/webstone/`.

[31] MOSBERGER, DAVID; JIN, TAI. **HTTPerf - A Tool for Measuring Web Server Performance**, 1999. Acessado em 2004, disponível em `http://www.hpl.hp.com/personal/David_Mosberger/httperf.html`.

[32] NAHUM, E. M.; BARZILAI, T. P.; KANDLUR D. D. **Performance Issues in WWW Servers**. *IEEE/ACM Transactions on Networking*, volume 10, pp. 2–11. ACM Press, fevereiro de 2002.

[33] NAHUM, E. M.; ROSU, M.; SESHAN, S.; ALMEIDA, J. **The Effects of Wide-Area Conditions on WWW Server Performance**. *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pp. 257–267. ACM Press, 2001.

[34] PANDIT, H.; SCHMITT, J.; STEINMETZ, R. **Network Calculus Meets Queueing Theory - A Simulation Based Approach to Bounded Queues**. *Twelfth IEEE International Workshop on Quality of Service, Montreal, Canada*, junho de 2004.

[35] ROSS, SHELDON M. ***Simulation***. Academic Press, 2002.

[36] SCHROEDER, B.; HARCHOL-BALTER, M. **Web Servers under Overload: How Scheduling Can Help**, In *18th International Teletraffic Congress. Berlin, Germany.* Original: Technical Report CMU-CS-02-143, Carnegie-Mellon University, setembro de 2003.

[37] STANDARD PERFORMANCE EVALUATION CORPORATION. **SPECWeb 99**, 2000. Acessado em 2004, disponível em `http://www.spec.org/benchmarks.html`.

[38] TRANSACTION PROCESSING PERFORMANCE COUNCIL. **TPC-W**, 2001. Acessado em 2004, disponível em `http://www.tpc.org/tpcw/default.asp`.

[39] VOIGT, T.; TEWARI, R.; FREIMUTH, D.; MEHRA, A. **Kernel Mechanisms for Service Differentiation in Overloaded Web Servers**, In *Proceedings of the 2001 USENIX Annual Technical Conference, Boston*, junho de 2001.

[40] WELSH, M.; CULLER, D. **Adaptive Overload Control for Busy Internet Servers**, In *Proceedings of the 4th USENIX Conference on Internet Technologies and Systems*, março de 2003.

[41] WELSH, M.; CULLER, D. **Overload Management as a Fundamental Service Design Primitive**, In *Proceedings of the Tenth ACM SIGOPS European Workshop. Saint-Emilion, France*, setembro de 2002.

[42] WOLFRAM RESEARCH, INC. **Mathworld: Calculus and Analysis, Integral Transforms, Convolution**, 1999. Acessado em 2004, disponível em `http://mathworld.wolfram.com/`.