Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Ciência da Computação

Tese de Doutorado

Um Modelo para Tarifação Confiável em Computação em Nuvem

Ana Cristina Alves de Oliveira Dantas

Campina Grande, Paraíba, Brasil
Novembro – 2015

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Programa de Pós-Graduação em Ciência da Computação

Um Modelo para Tarifação Confiável em Computação em Nuvem

Ana Cristina Alves de Oliveira Dantas

Tese submetida à Coordenação do Curso de Pós-Graduação em Informática da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sistemas de Computação

Marco Aurélio Spohn e Reinaldo Cezar M. Gomes
(Orientadores)

Campina Grande, Paraíba, Brasil

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCG

D192m Dantas, Ana Cristina Alves de Oliveira.
Um modelo para tarifação confiável em computação em nuvem / Ana Cristina Alves de Oliveira Dantas. – Campina Grande, 2015.
209 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2015.
"Orientação: Prof. Dr. Marco Aurélio Spohn, Reinaldo Cezar M. Gomes".
Referências.

1. Computação em Nuvem. 2. Análise de Tráfego de Rede. 3. Detecção de Anomalias de Tráfego. 4. Modelo de Custo. 5. Tarifação de Serviços. I. Spohn, Marco Aurélio. II. Gomes, Reinaldo Cezar M. III. Título.

CDU 004.7 (043)

"UM MODELO PARA TARIFAÇÃO CONFIÁVEL EM COMPUTAÇÃO EM NUVEM"

Ana Cristina Alves de Oliveira Dantas

ANA CRISTINA ALVES DE OLIVEIRA DANTAS

TESE APROVADA EM 30/11/2015

Marco A. Spohn

MARCO AURELIO SPOHN, Ph.D, UFFS
Orientador(a)

Reinaldo Cézar de Morais Jr.

REINALDO CÉZAR DE MORAIS GOMES, Dr., UFCG
Orientador(a)

JOSÉ ANTÃO BELTRÃO MOURA, Ph.D, UFCG
Examinador(a)

ANDREY ELÍSIO MONTEIRO BRITO, Dr., UFCG
Examinador(a)

DENIO MARIZ TIMÓTEO DE SOUSA, Dr., CEFET-PB
Examinador(a)

STÊNIO FLÁVIO DE LACERDA FERNANDES, Dr., UFPE
Examinador(a)

CAMPINA GRANDE - PB

"Filho, desde tua mocidade aplica-te à disciplina e até com cabelos brancos encontrarás a sabedoria. Como o lavrador e o semeador, cultiva-a, e espera pacientemente seus bons frutos, porque te cansarás um pouco em seu cultivo, mas em breve comerás de seus frutos."

(Eclesiástico 6, 18-20)

Dedicatória

Dedico este trabalho a meus pais, Odete e Severino.

Resumo

A computação em nuvem define uma infraestrutura virtual para prestação de serviços em rede sob demanda. Os clientes contratam serviços em que a infraestrutura primária de *hardware* e *software* encontra-se em centros de dados remotos e não localmente e sobre seu próprio domínio. Há uma necessidade de ferramentas de monitoramento regulatório, que possam operar dentro da infraestrutura do provedor, ou fora dele, deixando os clientes a par do estado atual ou do histórico do desempenho dos serviços contratados.

A computação em nuvem é fortemente dependente das redes computadores e o desempenho dos serviços em nuvem pode ser monitorado via métricas de rede. O conhecimento de métricas de desempenho sobre a execução dos serviços contribui para promover a relação de confiança entre cliente e provedor, bem como fornece subsídios para contestações em faturas, caso necessário.

Um modelo de *tarifação confiável* envolve a disponibilização de métricas de desempenho dos serviços contratados, de modo que o cliente possa aferir as tarifas cobradas. Clientes e provedores podem alternar papeis em diferentes níveis de prestação de serviços de computação em nuvem. Um cliente no nível de infraestrutura pode ser um provedor de dados, por exemplo. Um modelo de tarifação confiável fornece subsídios também ao provedor de serviços para melhorar a alocação de recursos, bem como indicadores para investimentos em infraestrutura que evitem perdas financeiras causadas pelo pagamento de multas por descumprimento de acordos de nível de serviço.

O objeto desta tese de doutorado é desenvolver um modelo para tarifação confiável de serviços de computação em nuvem que envolva a detecção e notificação de anomalias de tráfego de rede em tempo real que auxilie na estimativa do custo causado por tais anomalias para o modelo de negócio e que contribua para um processo de alocação de recursos capaz de reduzir custos com penalidades financeiras. A validação do modelo foi realizada por meio de escalonamento de recursos baseado em custo. O modelo de tarifação confiável integrado ao mecanismo de escalonamento reduziu custos e perdas financeiras provenientes de violações de acordos de nível de serviço.

Abstract

Cloud computing defines a virtual infrastructure to provide network services on demand. Customers contract services where the primary infrastructure of hardware software is in remote data centers and on the customer own domain. Sharing the same network, or the same physical machine, among various tenants entails some concerns related to information confidentiality, security, troubleshooting, separation of responsibilities for guaranteeing the quality of the technical goals across the different abstraction levels, and how the customer may monitor the use of services and eventual failures. Prior to cloud computing, allowed the service providers dominate the entire chain of information, providing information to enable them to manage the business globally to avoid financial losses and increase profits. With the use of cloud computing services, the customer possesses no control over levels virtualization services that are supporting the level you are operating. A client in infrastructure level can be a data provider, for instance. Thus, it is important to have appropriate tools to keep track of the performance of the contracted services. Cloud computing is heavily dependent on computer networks. In this sense, it is a business differential to provide network performance metrics either for the customers, which is an important non-functional requirement that is sometimes ignored by many cloud service providers. The disposal of real-time performance metrics contributes to promote trust relationship between customer and provider, and to aid the provider to better dimension resources to avoid finacial losses. The object of this doctoral thesis is to develop a model for reliable charging of cloud computing services cloud that accomplishes the network traffic anomaly detection and appropriate notification in real time, as well as enables the estimation of the cost caused by anomalies to business model.

Agradecimentos

Eu vou voltar no tempo e unir os agradecimentos que fiz no mestrado aos que entraram em minha vida desde então...

Agradeço a Deus e a todos a quem Ele concede o dom de cuidar de mim.

Primeiramente, à minha família, mas especialmente aos meus pais Severino e Odete e à minha irmã Karolzinha. Eles sempre me deram exemplos para tudo o que há de bom, incentivando-me e acreditando em mim. Ao meu esposo, Ranyelson, e à minha filha Marianna, porque abriram mão de muitos momentos em que poderíamos estar juntos para que esse trabalho fosse concretizado. Ao casar, eu aumentei a minha família e me sinto muito feliz por isso. Agradeço aos 'Dantas' na pessoa da minha sogra-mãe, Marlene.

Aos meus professores, todos. Amigos e educadores em todos os sentidos. Guiaram-me pela mão, por vezes de olhos vendados, e me ajudaram a chegar até aqui. Especialmente aos professores do IFPB (antiga ETFPB e antigo CEFET-PB) onde estudei o pró-técnico, o curso técnico integrado e a graduação. Lembrança especial aos professores Leônidas e Kamienski, tão exigente e dedicado, que me orientou na vida acadêmica e me abriu muitas oportunidades.

Agradeço ao meu orientador do mestrado, Fubica, pelas dicas e conselhos que fizeram diferença em minha vida profissional e pessoal. Agradeço ao meu orientador Marco Spohn pela oportunidade de entrar no doutorado na UFCG sob sua orientação e por sempre ter estado presente em todas as etapas por que passei. A Reinaldo Cézar, que além do posto de amigo e ex-colega de turma, assumiu o papel de co-orientador, ajudou-me e apoiou em assuntos relacionados à tese, ou não.

Sempre serei grata à professora Judith Kelner e ao professor Djamel Sadok pela oportunidade de trabalhar durante 2 anos como pesquisadora do Grupo de Pesquisa em Redes e Telecomunicações (GPRT/CIn/UFPE) e de ainda poder optar entre fazer o doutorado na UFPE. No GPRT eu tive a oportunidade de trabalhar ao lado de pessoas que tanto admiro, como os "chefes" Kamienski, Stênio e Josy, e os amigos fiz que fiz por lá: Manuela, Arthur, Breno, Rafael, Patricia, Thiago, Alysson, Rodrigo, Felipe, Ernani, Ricardo, Juliane, Chico, Joma, Tarciana, Fabrício, Moisés... Em Recife também conheci pessoas que fazem parte da

minha vida até hoje, como Andrezza, Marly, Marcella, que dividiram apartamento comigo e outros colegas do CIn, como Marcinho, Mario, Marcelo Siqueira e Fabrício.

Aos colegas de "colégio" e "faculdade" do tempo da ETFPB e do CEFET-PB que, mesmo não estudando mais juntos, nunca perdemos o contato e o carinho. Hoje, como professora, eu agradeço aos colegas de trabalho do IFPB e aos alunos. Alguns colegas eu pude reencontrar na UFCG, como Nelson, Robson, Carol, Nazareno, Daniela e Gustavo. Aos professores e alunos do tempo em que ensinei na Universidade Salgado de Oliveira (UNIVERSO) e na Faculdade dos Guararapes. Adquirir conhecimento só faz sentido, quando podemos compartilhá-lo.

Aos amigos que fiz na UFCG, em especial no LSD e no LATEC. Companheiros de comemorações super engraçadas e trabalhos árduos que nos aproximaram muito. Tenho que ressaltar aqueles que dividiram sala ou projeto comigo e que guardarei sempre com muito carinho: Pryscilla, Elayne, Cleide, Aninha, Kêka, Vera, Lili (Josenita), Ayla, Andrey, Esther, Lívia, Raquel, Marcus, Osorinho, Bárbara, William, Walfredo, Zane, Milena, Marcelo, Cadu, Rostand, Marcelo Iury, Matheus, Nigini, Fireman, Elizeu, Lauro, Bruno, Léo, Flávio, Ramon, Dimas, Gustavo, Petrônio, Anderson, Cezár, Nailson, Gildo e Thaciana.

Aos membros do Grupo de Pesquisa em Redes Convergentes (GPRC/IFPB), especialmente ao professor Rhavy e aos orientandos que me ajudaram a instalar o laboratório, Flávio, Henryson, Aleciano, Eduardo e Adrielly.

Aos colegas do Grupo de Pesquisa em Engenharia de Sistemas (SE Group) da Universidade Técnica de Dresden pelos agradáveis e enriquecedores momentos que vivenciei durante o doutorado sanduíche, especialmente ao professor Christof Fetzer, Thordis, Thomas, Florian, Do, Rasha, Sergey, André, Diogo, Sebastian, Martin, Vesna e Stephan.

Aos irmãos que fiz na comunidade católica de que participei durante oito anos. Contribuiram para que eu procurasse conhecer-me melhor, a buscar ir além dos limites que vejo e tentar ajudar as pessoas de uma forma mais concreta.

Aos que trabalharam comigo na Coteminas ou na Springs Global US, especialmente a Bill Fields, Mike Sherril, John Dykstra, Judy Davenport, Charlie, Daniel, Helga, Cláudia, Rosemary e Douglas, que me apoiaram na decisão de mudar do cargo de analista de TI para investir no doutorado e na carreira acadêmica.

Aos que estão lendo esta tese, pois fazem este trabalho valer a pena. Muito obrigada.

Conteúdo

1 Introdução **1**

1.1 Contextualização 3

1.2 Motivação 4

1.3 Problematização 6

1.3.1 Problema de Negócio 7

1.3.2 Problema Técnico 7

1.4 Objetivos 8

1.4.1 Objetivo Geral 8

1.4.2 Objetivos Específicos 8

1.5 Solução Proposta 9

1.6 Metodologia 12

1.6.1 Pesquisa Bibliográfica 12

1.6.2 Pesquisa Experimental 13

1.7 Contribuições e Publicações 14

1.8 Estrutura do Documento 16

2 Fundamentação Teórica **17**

2.1 Computação em Nuvem 17

2.1.1 Arquitetura de Sistemas de Computação em Nuvem 21

2.1.2 Modelos de Negócio de Computação em Nuvem 23

2.1.3 Classificação quanto ao Domínio Administrativo 30

2.1.4 Federação de Nuvens 32

2.2 Gerência de Tráfego em Redes de Computadores 33

2.2.1 Medições e Monitoramento de Redes 34

2.2.2 Análise de Tráfego de Rede 35
2.3 Gerência de Acordo de Nível de Serviço 37
2.4 Detecção de Anomalias de Tráfego de Rede 41
2.4.1 Técnicas Baseadas em Casamento de Padrões 42
2.4.2 Técnicas Baseadas no Comportamento do Tráfego 42
2.5 Validação de Técnicas de Detecção de Anomalias 44
2.6 Conclusões 46
2.6.1 Análise de Tráfego de Serviços de Computação em Nuvem 46
2.6.2 Acordos de Níveis de Serviço para Computação em Nuvem 46
2.6.3 Detecção de Anomalias de Tráfego de Rede de Serviços de Computação em Nuvem 47

3 TADE: Arquitetura para Detecção e Gerenciamento de Anomalias de Tráfego para Serviços de Computação em Nuvem 49
3.1 Funcionalidades 50
3.2 Produtos de *Software* 52
3.2.1 Interações entre os Produtos de *Software* 53
3.2.2 *Anomaly Detector* 55
3.2.3 *Anomaly Detector Agent* 55
3.2.4 Aplicação *Web* 55
3.3 Implantação dos Produtos de *Software* 56
3.3.1 Verificação 58
3.3.2 Ambiente de Testes 58
3.4 Validação 59
3.5 Conclusões 62

4 Mecanismo para Detecção de Anomalias de Tráfego baseado em Entropia 64
4.1 Implementação da Técnica de Deteção de Anomalias Baseada em Entropia ... 65
4.2 Construção de Séries Temporais de Entropia 65
4.3 Processamento das Séries Temporais de Entropia 69
4.4 Avaliação de Desempenho 69
4.4.1 Metodologia 69

4.4.2 Resultados Obtidos 71
4.4.3 Estatística Inferencial 73
4.5 Conclusões 76

5 EMATADE: Arquitetura de Detecção de Anomalias baseada em Entropia e Aprendizagem de Máquina 79
5.1 EMATADE: Detecção Híbrida de Anomalias 80
5.2 Detecção de Anomalias por Aprendizagem de Máquina 81
5.3 Metodologia 82
5.3.1 Simulação de Ataque de Negação de Serviço à Nuvem 83
5.3.2 Ataque Real de Negação de Serviço Distribuído a um Provedor de Telecomunicações 86
5.4 Conclusões 89

6 Escalonamento baseado em Custo 91
6.1 Modelo de Custo para Dados-como-Serviço 91
6.1.1 Modelo do Sistema 92
6.1.2 Modelo de Custo 93
6.2 Escalonamento baseado em Custo 96
6.2.1 Formalização do Problema 97
6.2.2 Formalização da Solução 99
6.2.3 Algoritmo de Escalonamento 101
6.3 Estudos Preliminares 103
6.3.1 Escalonamento por Alternância Circular 103
6.3.2 Simulação 104
6.3.3 Resultados Preliminares 105
6.4 Conclusões 106

7 Modelo para Tarifação Confiável em Computação em Nuvem 108
7.1 Definição Inicial de Custos para os Serviços 111
7.2 Definição dos SLAs 112
7.3 Experimento para Definição de Variáveis 113

7.3.1 Caracterização do Tráfego de Rede 113
7.3.2 Estimativa de Variáveis do Modelo 115
7.4 Definição de Penalidades 115
7.5 Injeção de Anomalias de Tráfego 119
7.6 Escalonamento baseado em Custo Integrado ao Mecanismo de Detecção de Anomalias 119
7.7 Impacto do Custo de Anomalias 120
7.8 Conclusões 126

8 Trabalhos Relacionados **128**
8.1 Detecção de Violações de SLAs 129
8.2 Detecção de Anomalias baseada em Entropia 130
8.3 Tarifação de Serviços de Computação em Nuvem 132

9 Considerações Finais **134**

A Convenções Adotadas **153**

B Caracterização da Utilização de Serviços de Computação em Nuvem **155**
B.1 Objetivos da Pesquisa 156
B.2 Projeto Experimental 156
B.3 Estatística Descritiva 159
B.4 Estatística Inferencial 167
B.5 Validação 173
B.6 Conclusões 175

C Levantamento de Requisitos da TADE **177**

D Casos de Uso da TADE **180**
D.1 Planejamento de Versões 180
D.2 Coletar Pacotes de Rede (TADE_UC-1) 181
D.3 Processar Pacotes de Rede (TADE_UC-2) 185
D.4 Notificar Anomalias (TADE_UC-3) 188

E Ambiente de Testes da TADE **190**

F Dados Suplementares do Capítulo 7 **195**

Lista de Símbolos

AAA – Autenticação, Autorização e Contabilidade (*Autentication, Authorization, Accounting*)

AMQP – Protocolo de Enfileiramento de Mensagens Avançado (*Advanced Message Queuing Protocol*)

API – Interface para Programação de Aplicação (*Application Programming Interface*)

CA – Teste α de Cronbach (*Cronbach's* α)

CCOA – Arquitetura Aberta de Computação em Nuvem (*Cloud Computing Open Architecture*)

CEP – Processamento de Eventos Complexos (*Complex Event Processing*)

CTIC – Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação

DaaS – Dados-como-Serviço (*Data-as-a-Service*)

DC – Centro de Dados (*Data Center*)

DoS – Ataque de negação de serviço (*Denial of Service*)

DDoS – Ataque de negação de serviço distribuído (*Distributed Denial of Service*)

DEMO – Projeto $\&$ Metodologia de Engenharia para Organizações (*Design* $\&$ *Engineering Methodology for Organizations*)

DPI – Inspeção Profunda de Pacotes (*Deep Packet Inspection*)

EbAT – Detecção de Anomalias baseada em Entropia (*Entropy-based Anomaly Detection*)

EMATADE – Detecção de Anomalias baseada em Entropia e Aprendizagem de Máquina (*Entropy and Machine Learning-based Anomaly Detection*)

E/S – Entrada e Saída

EO – Ontologia Empresarial (*Enterprise Ontology*)

ERP – Planejamento de Recursos de Empresas (*Enterprise Resource Planning*)

FCAPS – Falha, Configuração, Contabilidade, Desempenho e Segurança (*Failure, Configuration, Accounting, Performance and Security*)

FPGA – *Field Programable Gateway Array*

GQM – Objetivo, Questão e Métrica (*Goal, Question, Metric*) HP – Empresa *Hewlett Packard*

IaaS – Infraestrutura-como-Serviço (*Infrastructure-as-a-Service*)

ICMP – Protocolo de Mensagens de Controle da Internet (*Internet Control Message Protocol*)

IDS – Sistema de Detecção de Intrusão (*Intrusion Detection System*)

IPS – Sistema de Prevenção de Intrusão (*Intrusion Prevention System*)

ISP – Provedor de Serviços de Internet (*Internet Service Provider*)

JiT – Imediatamente pronto (*Just-in-Time*)

KL – *Kullback-Leibler*

MILP – Programação Linear Inteira Mista (*Mixed Integer Linear Programming*)

MPLS – Comutação de Rótulos Multiprotocolo (*Multiprotocol Label Switching*)

MOS – Média dos Pontos de Opinião (*Mean Opinion Score*)

NaaS – Rede-como-Serviço (*Network-as-a-Service*)

NRNE – Entropia de Rede Relativa Normalizada (*Normalized Relative Network Entropy*)

OSI – Interconexão de Sistemas Abertos (*Open Systems Interconnection*)

PaaS – Plataforma-como-Serviço (*Platform-as-a-Service*)

PC – Computador Pessoal (*Personal Computer*)

PCA – Análise de Componentes Principais (*Principal Component Analysis*

P2P – Par-a-Par, ou Entre-Pares (*Peer-to-Peer*)

PME – Pequena a Média Empresa

QoS – Qualidade de Serviço (*Quality-of-Service*)

RIC – *Rate-Interval Curves*

RNP – Rede Nacional de Ensino e Pesquisa

SaaS – Software-como-Serviço (*Software-as-a-Service*)

SCN – Rede de Colaboração de Serviços (*Service Collaboration Network*)

SLA – Acordo de Nível de Serviço (*Service Level Agreement*)

SLC – Contrato de Nível de Serviço (*Service Level Contract*)

SLI – Indicadores de Nível de Serviço (*Service Level Indicators*)

SLO – Objetivo de Nível de Serviço (*Service Level Objective*)

SME – Pequena a Média Empresa (*Small to Medium Size Enterprises*)

SOA – Arquitetura Orientada a Serviço (*Service Oriented Architecture*)

TADE – Mecanismo de Detecção de Anomalias de Tráfego (*Traffic Anomaly Detection Engine*)

TCO – Custo Total de Propriedade (*Total Cost of Ownership*)

TCP – Protoloco de Controle de Transmissão (*Transmission Control Protocol*)

TI – Tecnologia da Informação

TIC – Tecnologia da Informação e Comunicação

UDP – Protocolo de Datagrama de Usuário (*User Datagram Protocol*)

VM – Máquina Virtual (*Virtual Machine*)

VoIP – Voz sobre IP (*Voice-over-IP*)

VPC – Nuvem Privada Virtual (*Virtual Private Cloud*)

VPN – Rede Privada Virtual (*Virtual Private Network*)

VRML – Camada de Mediação de Recursos Virtual (*Virtual Resources Mediation Layer*)

XaaS – Qualquer coisa-como-serviço (*Everything-as-a-Service*)

XML – Linguagem de Marcação Extensível (*eXtensible Markup Language*)

Lista de Figuras

1.1 Visão geral da fronteira entre cliente e provedor para execução do serviço contratado. 6
1.2 Processo de detecção de anomalias. 10
1.3 Visão geral da governança de rede para sistemas de computação em nuvem [Oliveira et al. 2015a]. 11
1.4 Ciclo de vida do gerenciamento de processo de negócio aplicado à governança de rede em sistemas de computação em nuvem. 12

2.1 Evolução dos paradigmas de computação em seis fases distintas [Voas e Zhang 2009]. 19
2.2 Arquitetura de computação em nuvem e exemplos de serviços oferecidos por cada camada [Weinhardt et al. 2009]. 20
2.3 Modelos de negócio de computação em nuvem. Adaptado de [Zhang et al. 2010]. 24
2.4 Diferentes visões entre cliente e provedores de serviços de SaaS [Cusumano 2010]. 26
2.5 Exemplo de composição de uma *JiT Cloud*. Adaptado e traduzido de [Costa et al. 2010]. 33
2.6 Sondas de medições ativas e pontos de medições passivas [Ziviani e Duarte 2005]. 35
2.7 Monitoramento e análise de tráfego. 36
2.8 Exemplo de violação de SLA. Adaptado de [Cisco 2005]. 38
2.9 Gerenciamento de desempenho com valor agregado. Traduzido e adaptado de [Telecom 2007]. 39

3.1 Arquitetura TADE. 53
3.2 Fluxo de dados entre os três produtos de software que implementam a arquitetura TADE. 54
3.3 Desempenho do processamento de pacotes, variando a taxa de tráfego de entrada. 61
3.4 Percentual de perda de pacotes, com entrada de tráfego a 100 Mbps e 1 Gbps. 62

4.1 Fluxo de processamento e dados da detecção de anomalias de rede baseada em entropia. 66
4.2 Série temporal da largura de banda..... 70
4.3 Séries temporais de entropia para $n = 5$ e $m = 6$ (a) e $m = 7$ (b)..... 72
4.4 Séries temporais de entropia para $n = 10$ e $m = 6$ (a) e $m = 7$ (b). 73
4.5 Intervalos de confiança para as estimativas dos valores das variáveis *F-Measure*, *precision* e *recall*. 77

5.1 EMATADE: método híbrido proposto para detecção de anomalias..... 80
5.2 Nós que fazem parte do ataque de DoS [Quoc et al. 2014]. 84
5.3 Taxa de pacotes das aplicações MapReduce versus HDFS..... 85
5.4 Séries temporais de entropia para $n = 5$ com: $m = 6$ (a) e $m = 7$ (b). 85
5.5 Séries temporais de entropia para $n = 10$ com: $m = 6$ (a) e $m = 7$ (b)..... 86
5.6 Vazão de entrada e de saída. 87
5.7 Séries temporais de entropia para $n = 5$ com: $m = 6$ (a) e $m = 7$ (b). 88
5.8 Séries temporais de entropia para $n = 10$ com: $m = 6$ (a) e $m = 7$ (b)..... 89

6.1 VM acessando uma fonte de dados armazenada no centro de dados B. Traduzido e adaptado de [Oliveira et al. 2015b] 96
6.2 Estratégia de escalonamento baseada em custo. 97
6.3 Gráfico de caixa (*boxplot*) para avaliação variação dos custos comparando as estratégias de escalonamento [Oliveira et al. 2015b]. 107

7.1 Gerenciamento proposto para execução de serviços. 109
7.2 Exemplo de contratação de serviços de computação em nuvem em diferentes níveis de abstração. 110

7.3 Interação entre o escalonador e os demais módulos do modelo. 111
7.4 *Crawling* distribuído [Quoc et al. 2015]. 114
7.5 Vazão de entrada e saída medida por hora para cada uma das 3 máquinas virtuais. 116
7.6 Estimativa das variáveis de custo calculadas por hora para cada VM. 117
7.7 Estimativa das variáveis de volume de dados calculadas por hora para cada VM. 117
7.8 Custos das 3 VMs utilizando os dois mecanismos de escalonamento. 122
7.9 Custo total para os serviços de DaaS e IaaS de cada VM. 123
7.10 Perda total para os serviços de DaaS e IaaS em cada VM. 124
7.11 Perda acumulada para os serviços de DaaS e IaaS em cada VM. 127

B.1 Diagrama de Pareto com os resultados obtidos para a Questão 1. 159
B.2 Diagrama de Pareto com os resultados obtidos para a Questão 2. 160
B.3 Diagrama de Pareto com detalhamento do percentual de respondentes por instituição. 160
B.4 Diagrama de Pareto com os resultados obtidos para a Questão 3. 161
B.5 Gráfico de pizza com os resultados obtidos para a Questão 4. 162
B.6 Diagrama de Pareto com os resultados obtidos para a Questão 5. 162
B.7 Gráfico de pizza com os resultados obtidos para a Questão 6. 163
B.8 Gráfico de pizza com os resultados obtidos para a Questão 7. 163
B.9 Diagrama de Pareto com os resultados obtidos para a Questão 8. 164
B.10 Gráfico de pizza com os resultados obtidos para a Questão 9. 165
B.11 Gráfico de pizza com os resultados obtidos para a Questão 10. 166
B.12 Resultados obtidos utilizando Escala de Likert para a Questão 11. 166
B.13 Resultados obtidos utilizando Escala de Likert para a Questão 12. 167
B.14 Variabilidade dos resultados. 168
B.15 Círculo de correlações. 170
B.16 Comparativo do grau de satisfação geral dentre os clientes com e sem contratos de serviços de computação em nuvem. 174

D.1 Diagrama de sequência do caso de uso 1 da TADE. 182

D.2 Diagrama de Sequência do caso de uso 2 da TADE. 186
D.3 Diagrama de sequência do caso de uso 3 da TADE..... 189

E.1 Topologia do ambiente de testes. 191

F.1 Vazão de entrada e saída das 3 máquinas virtuais..... 196

Lista de Tabelas

2.1 Tabela de contingência [Salfner et al. 2010] 44
2.2 Métricas obtidas a partir da tabela de contingência [Salfner et al. 2010]. 45

3.1 Tratamentos do experimento para avaliar desempenho da implementação da arquitetura TADE. 60

4.1 Tratamentos usados na análise do detector de anomalias de tráfego baseado em entropia. 71
4.2 Valores de cada medição para a Métrica F. 74
4.3 Erros experimentais obtidos para a Métrica F. 74
4.4 Percentuais de variação dos efeitos. 75
4.5 Média obtida para as métricas *recall*, *precision* e *F1* usando detecção de anomalias baseada em entropia. 76

5.1 Tratamentos avaliados. 83
5.2 Lista de Ataques de TCP SYN FLOOD [Quoc et al. 2014]. 84
5.3 Resultado do processo de detecção de anomalias no ataque de DoS à nuvem. . 86
5.4 Resultado do processo de detecção de anomalias no ataque de DoS ao provedor de Internet. 90

6.1 Tratamentos avaliados. 105
6.2 Configuração do sistema. 106

7.1 Custo por centro de dados. 112
7.2 SLAs Contratados. 113
7.3 Modelo de penalidades para anomalias detectadas. 118

7.4 Número de anomalias injetadas, número máximo aceitável de anomalias e número de anomalias excedentes. 121
7.5 Custos totais das VMs para DaaS utilizando o escalonamento integrado e o escalonamento padrão, ... 124
7.6 Perdas totais resultantes das penalidades aplicadas aos serviços de DaaS..... 125
7.7 Perdas totais resultantes das penalidades aplicadas aos serviços de IaaS..... 125
7.8 Perdas e custos relativos entre os valores obtidos pelo escalonamento integrado e o escalonamento padrão baseado em custo..... 126

B.1 Metas, questões e métricas da pesquisa de caracterização de serviços de computação em nuvem..... 158
B.2 Levantamento de metas técnicas acordadas em SLAs. 165
B.3 *Eigenvalues*, percentual de variância e percentual de variância acumulada 169
B.4 *Loadings* de cada variável..... 169
B.5 *Scores* de cada indivíduo. 171
B.6 Análise de Consistência Interna do Questionário. 175

C.1 Propriedades utilizadas para definir os requisitos..... 177
C.2 Categorização dos requisitos funcionais e não funcionais..... 178
C.3 Descrição dos requisitos e das dependências inter-requisitos. 179

D.1 Casos de Uso para implementação da arquitetura TADE..... 181

E.1 Especificação dos softwares utilizados nas máquinas do ambiente de testes. ... 192
E.2 Especificações do *hardware* das máquinas SLAVE do ambiente de testes. 193
E.3 Especificação do hardware da máquina ANALYZER (HP ProLiant DL380 G7) do ambiente de testes..... 193
E.4 Equipamentos adicionais do ambiente de testes. 194

F.1 Estatísticas sobre a vazão do tráfego (Mbps) de entrada e saída das VMs. 195
F.2 Estimativa das variáveis R, T, q e k calculadas a cada hora de experimento para a VM #1. 197
F.3 Estimativa das variáveis R, T, q e k calculadas a cada hora de experimento para a VM #2. 198

F.4 Estimativa das variáveis R, T, q e k calculadas a cada hora de experimento para a VM #3. 199
F.5 Estimativa das variáveis R, T, q e k calculadas a para o período total do experimento. 200
F.6 Decisões do escalonamento baseado em custo padrão, detalhamento de custos e perdas para a VM 1. 201
F.7 Decisões do escalonamento baseado em custo padrão, detalhamento de custos e perdas para a VM 2. 202
F.8 Decisões do escalonamento baseado em custo padrão, detalhamento de custos e perdas para a VM 3. 203
F.9 Decisões do escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, detalhamento de custos e perdas para a VM 1. 204
F.10 Decisões do escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, detalhamento de custos e perdas para a VM 2. 205
F.11 Decisões do escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, detalhamento de custos e perdas para a VM 3. 206

Lista de Algoritmos

6.1 Algoritmo de escalonamento baseado em custo para a VM u_z. 103

7.1 Algoritmo de escalonamento baseado em custo para a VM u_z integrado ao mecanismo de detecção de anomalias. 120

Capítulo 1

Introdução

"Sê como o arqueiro: faz da tua vida
um alvo."

Ortega y Gasset

A utilização de serviços via redes de computadores vem alterando com o passar dos anos o modelo em que clientes domésticos necessitam que sejam instalados em seus computadores pessoais todos os programas de que irão precisar no dia-a-dia de suas atividades, como editores de texto, clientes de correio eletrônico, agenda, ferramentas de gerência de projeto e jogos. Ao passo que clientes empresariais precisam dispor de profissionais e recursos de tecnologia da informação e comunicação (TIC) para realizar o projeto, a implantação e a manutenção de centros de dados (*datacenters*) onde possam ser implantados no âmbito da empresa todos os serviços necessários à sua operação, como servidores de aplicações, compartilhamento de arquivos, correio eletrônico, serviços de diretório, sistemas ERP (*Enterprise Resource Planning*), entre outros.

Os usuários, clientes tanto empresariais como domésticos, estão passando a utilizar um novo modelo, onde cada vez menos precisam instalar *softwares* em seus próprios computadores, ou possuírem seus próprios centros de dados, utilizando serviços que são disponibilizados por diversos provedores terceirizados que, de modo geralmente especializado, ou mais abrangente, hospedam e disponibilizam serviços de forma distribuída em centros de dados localizados em algum lugar no mundo. Essa mudança proporcionou que diversos clientes de dispositivos móveis, com recursos até mesmo modestos de *hardware*, por exemplo, pudes-

sem ser clientes de aplicações sofisticadas, com alta escala e de baixo custo. A esse novo modelo que vem se sagrando mundialmente é dado o nome de **computação em nuvem**, ou *cloud computing*.

Exemplos de serviços na nuvem são o armazenamento de dados em discos virtuais, como o Dropbox, Google Drive e SkyDrive, em que os clientes têm a impressão de estarem editando arquivos localmente, enquanto as atualizações são sincronizadas entre os servidores que compõem a nuvem e podem ser acessadas via Internet em qualquer lugar. Há ainda a possibilidade de nem ser necessária a instalação local de aplicativos como editores de texto, planilha eletrônica, agenda eletrônica, mensagens instantâneas e clientes de e-mail, como ocorre com o Google Docs, Calendar, Imo Instant Messenger e Yahoo Mail, entre tantos serviços tanto para uso profissional como entretenimento.

Weinhardt et al. [Weinhardt et al. 2009] elaboraram um quadro comparativo entre grades computacionais e computação em nuvem. Na última há o estabelecimento e o cumprimento de acordos de nível de serviço, o que não é uma característica nativa de grades computacionais, pois estas apresentam imprevisibilidade na alocação dos recursos. Os autores definiram também uma ontologia para modelos de negócios em computação em nuvem. O modelo de negócio original de computação em nuvem contempla três camadas diferentes de serviços, que são: infraestrutura, plataforma e aplicação (*software* aplicativo). Cada camada é responsável por prover serviços no nível de negócio do modelo.

Armbrust et al. [Armbrust et al. 2010] discutiram o modelo econômico de computação em nuvem, discriminando as diferenças entre computação em nuvem e centros de dados privados, ressaltando que nos sistemas de computação em nuvem a escala econômica pode ser expandida rapidamente para sistemas extremamente grandes ou reduzida rapidamente. Para manter as características de disponibilidade escalável de recursos, o monopólio sobre serviços não deve pertencer a um determinado provedor, haja vista que a existência de múltiplos provedores dá suporte a negociações e acordos para balanceamento de carga federado e que sejam assumidos acordos que possam ir além do provisionamento de pico.

1.1 Contextualização

O aumento da demanda de serviços de comunicação trouxe também a necessidade da redução dos custos necessários para implantar e manter o sistema de comunicação. A *virtualização de recursos*, como roteadores, linhas de comunicação, servidores, discos, sistemas operacionais, aplicações, entre outros, ganhou a atenção dos provedores de serviços e vem sendo aplicada como forma de redução de custos, realização de testes de *software*, gerenciamento de infraestrutura e otimização da utilização de recursos.

O compartilhamento da mesma rede ou da mesma máquina física por parte de vários clientes acarreta algumas preocupações relacionadas à confidencialidade das informações, à segurança oferecida, à detecção de problemas no provimento de serviços, à separação de responsabilidades para garantias de metas de qualidade de serviço (QoS - *Quality-of-Service*) entre a empresa prestadora de computação em nuvem e da concessionária de telecomunicações (ISP - *Internet Service Provider*), bem como os recursos disponibilizados ao cliente para acompanhar a utilização dos serviços e eventuais falhas na prestação dos mesmos.

A computação em nuvem gerou uma demanda por sistemas de contabilização eficientes, que sejam leves o suficiente para garantir bom desempenho em tempo real e que ao mesmo tempo extraiam as informações chaves, seguindo padrões e com agilidade na busca e recuperação de informações. Os sistemas de contabilização fornecem subsídios para os sistemas de tarifação que também necessitam estar alinhados ao tipo de serviço acordado entre as partes e o que é realmente oferecido.

O tráfego de rede produzido por sistemas de computação em nuvem revela o comportamento dos usuários com relação à utilização dos serviços, uma vez que todos os serviços são acessados por meio da rede. Analisar o tráfego e reconhecer os fluxos de cada aplicação do sistema permite que sejam modelados os comportamentos de uso de cada serviço e que sejam extraídos padrões para o que se poderá considerar como funcionamento normal do sistema.

Com base no estudo do comportamento padrão do sistema, pode-se identificar desvios na operação normal do sistema. Estes podem indicar que aplicações estão gerando tráfego além do esperado; por exemplo, um serviço com erro está enviando pacotes de difusão (*broadcast*) na rede, ou que há ausência de tráfego às 10 horas para um serviço que sempre está operando

em horário comercial.

Nesse contexto, **anomalia de tráfego** é um termo bastante genérico que abrange ataques, tráfego indesejado devido a aplicações em erro, perda de pacotes e de injeção de tráfego indesejado de aplicativos não permitidos, entre outros comportamentos anormais que possam estar presentes no tráfego.

A qualidade da comunicação de rede entre provedor e cliente afeta significativamente o desempenho da maioria das aplicações que executam na nuvem [Shetty 2013]. Uma técnica genérica para analisar tráfego com acurácia e desempenho conhecidos e satisfatórios, para qualquer cenário de tráfego, continua um problema em aberto. A análise de tráfego não é uma tarefa trivial, especialmente em sistemas de alta vazão. Há necessidade de mecanismos de monitoramento leves, de alto desempenho e não intrusivos [Callado et al. 2010].

1.2 Motivação

Clientes de serviços de computação em nuvem não possuem domínio sobre todos os níveis de execução dos serviços contratados. Existem camadas abstratas para que um serviço seja executado e pode haver mais de um provedor envolvido no provimento de um mesmo serviço. Um provedor de dados pode contratar um provedor de infraestrutura para executar máquinas virtuais que, por sua vez, pode contratar um provedor de serviços de rede.

Os acordos de nível de serviço (*Service Level Agreement* – SLA) entre provedor e cliente são estabelecidos quando o serviço é contratado e as metas técnicas definidas para cada serviço devem ser garantidas. O monitoramento das metas técnicas é realizado pelo provedor de serviços. Após a contratação ou execução do serviço de computação em nuvem, o sistema de tarifação do provedor gera uma fatura, que é paga pelo cliente de acordo com a política de pagamento em vigor. Tanto o provedor quanto o cliente possuem seus sistemas de contabilidade gerencial, que analisam os custos e as receitas. Caso ocorra alguma anormalidade no provimento de um serviço, mesmo que o cliente observe a falha, os meios para comprovar essa irregularidade e contestar a tarifação de forma legítima não são claros.

A computação em nuvem não traz apenas benefícios. Novas estratégias para a implantação de Governança de TI (planejamento, controle, monitoramento e gestão) são necessárias para promover a transparência na prestação de serviços. Os prestadores de serviços devem

encontrar uma forma de atender às demandas de seus clientes, garantindo os princípios de confiabilidade, confidencialidade, integridade e disponibilidade [Khatri e Brown 2010].

Nesta tese foi realizada uma pesquisa para caracterização do uso de serviços de computação em nuvem. Um dos critérios analisados foram os tipos de SLAs contratados pelos clientes. Apenas 24% dos respondentes afirmaram ter contratado algum tipo de SLA, contudo dessse percentual todos indicaram que o monitoramento desses SLAs poderia ser feito também no nível de rede. Apenas 15% dos provedores analisados forneciam informações sobre métricas de não-conformidade de SLAs. Padronizar o acesso a uma interface aberta entre os provedores de computação em nuvem e clientes para dar suporte aos processos de gerenciamento de contabilidade sob a perspectiva do cliente também é importante. Maiores detalhes sobre o método empregado na pesquisa e sobre os resultados obtidos serão apresentados no Apêndice B.

Há uma necessidade de ferramentas mais especializadas e de monitoramento regulatório, que possam operar dentro da infraestrutura do provedor, ou fora dele, deixando os clientes a par do estado atual ou do histórico do desempenho dos serviços contratados.

Apesar da análise de conformidade de métricas de SLA ser uma exigência clara, os clientes muitas vezes não têm acesso padronizado a elas. Fornecedores podem beneficiar-se por venderem recursos que deveriam estar alocados, ou deixar de pagar multas pelo descumprimento de SLAs por não haverem evidências suficientes para que os clientes possam reclamar.

Um modelo de *tarifação confiável* auxilia o cliente, o qual pode ter acesso padronizado a métricas de desempenho dos serviços contratados de modo a aferir as tarifas cobradas, mas também fornece subsídios ao provedor de serviços para melhorar na alocação de seus recursos, bem como indicadores para investimentos em infraestrutura de modo a evitar perdas financeiras pelo pagamento de multas.

A Figura 1.1 apresenta uma visão geral da fronteira que separa o acesso de clientes e provedores aos dados sobre a execução dos serviços e apresenta uma proposta de acesso compartilhado a informações sobre SLAs e monitoramento de tráfego, que será defendida ao longo deste trabalho.

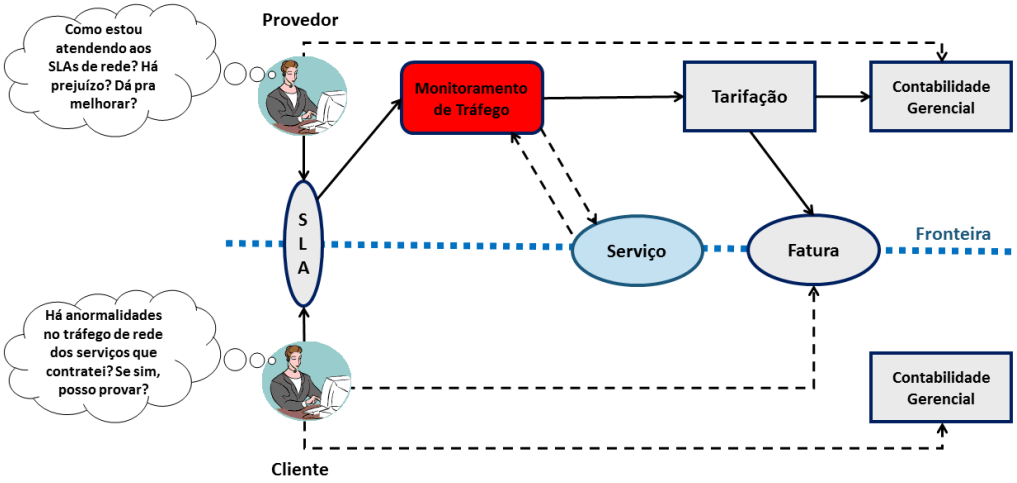


Figura 1.1: Visão geral da fronteira entre cliente e provedor para execução do serviço contratado.

1.3 Problematização

Vários desafios permanecem abertos relacionados à computação em nuvem, incluindo a falta de padronização de suas APIs para facilitar o desenvolvimento e mobilidade de aplicações em diversas nuvens, necessidade de novos protocolos de comunicação abertos garantindo sua agilidade e escalabilidade e um sistema de gerenciamento. Por exemplo, há uma grande quantidade de informações sobre conceitos, arquiteturas e modelos para implantar sistemas em nuvem. Por outro lado, há falta de informações padronizadas sobre a utilização, os contratos, e as principais métricas de desempenho exigidas pelos serviços de nuvem. Esses dados são geralmente privados, não-estruturados e até mesmo os clientes não sabem exatamente suas cláusulas e exigências contratuais.

Gerenciar e analisar tráfego em sistemas de computação em nuvem de larga escala é um desafio. As técnicas empregadas para monitorar e analisar tráfego em sistemas distribuídos convencionais apresentam diferenças com relação a sistemas de computação em nuvem. Nos métodos convencionais são feitas suposições de que existem padrões para os fluxos de rede, até aceitáveis para aplicações em redes corporativas, mas as aplicações que rodam em centros de dados na nuvem podem sofrer mudanças significativas nos padrões de tráfego [Zhang et al.

2010].

1.3.1 Problema de Negócio

Clientes de sistemas de computação em nuvem contratam serviços que nem sempre são providos com os mesmos requisitos de qualidade esperados e, acima de tudo, que foram acordados entre as partes. O descumprimento de SLA pode implicar em prejuízos financeiros tanto para o provedor de serviços, quanto para o cliente. Um cliente pode pagar na íntegra pelo provimento de um serviço, quando na prática o serviço provido e utilizado pelo cliente não esteve disponível durante todo o tempo cobrado ou não atendeu a todos os requisitos acordados no contrato de prestação de serviços. O provedor pode ter de pagar multas por quebra de contrato, ou ter de ceder recursos adicionais ao cliente, que não estavam previstos em seu plano de alocação, como crédito. As perguntas de pesquisa que se pretende responder nessa tese de doutorado são:

I. Como o cliente pode avaliar se seus serviços de rede em sistemas de computação em nuvem foram providos com os requisitos de qualidade contratados, comprovar se a tarifação dos serviços está correta e, quando necessário, ser compensado caso os serviços não forem devidamente prestados?

II. Como fornecer subsídios para que o provedor não tenha prejuízos causados por penalidades no provimento de serviços de computação em nuvem?

1.3.2 Problema Técnico

A solução para o problema de negócio apresentado envolve problemas de caráter técnico, contemplando ações para:

- Implantar um modelo de serviço de monitoramento de métricas de desempenho de rede que forneça informações relevantes aos tomadores de decisão;
- Notificar tanto o provedor quanto o cliente sobre o desempenho dos serviços contratados para fins de auditoria (contabilidade) e ajuste de contas (tarifação confiável), bem como indicador de investimentos em infraestrutura ou na contratação de novos serviços na nuvem;

- Desenvolver um modelo de tarifação de serviços de computação em nuvem que utilize o monitoramento da incidência de anomalias de tráfego de rede em tempo real e a estimativa do custo causado por tais anomalias para o modelo de negócio.

1.4 Objetivos

Este trabalho de tese de doutorado pretende desenvolver um modelo para monitoramento de tráfego e tarifação confiável de serviços de computação em nuvem que envolva a detecção e notificação de anomalias de tráfego de rede em tempo real e dê suporte à estimativa do custo causado por tais anomalias para o modelo de negócio.

1.4.1 Objetivo Geral

Desenvolver um mecanismo de contabilização e tarifação dos serviços de computação em nuvem que seja capaz de detectar anomalias de tráfego em tempo real, de estimar o custo causado por tais falhas na prestação do serviço levando em consideração questões relevantes, como a natureza da falha, o perfil do tráfego, o modelo de negócio do serviço contrado e o SLA estabelecido. De acordo com os dados obtidos pelos sistemas de contabilização e das anomalias detectadas, será estimado o custo do tráfego anômalo no contexto de cada serviço.

1.4.2 Objetivos Específicos

- **Objetivo 1:** Caracterizar a utilização de serviços de computação em nuvem;
- **Objetivo 2:** Projetar e implementar um sistema de monitoramento de tráfego para identificar anomalias de tráfego em tempo real;
- **Objetivo 3:** Caracterizar acordo de nível de serviço (SLA), seus objetivos (SLOs) e o modelo de negócio empregado para os serviços de computação em nuvem;
- **Objetivo 4:** Levantar custos incorridos por anomalias de tráfego com base no modelo de negócio do cliente;

- **Objetivo 5:** Propor um modelo de monitoramento e tarifação para os serviços de computação em nuvem, levando em consideração os prejuízos causados por tráfego anômalo.

1.5 Solução Proposta

Clientes de serviços de computação em nuvem não possuem controle sobre a infraestrutura física que suporta a execução dos serviços. É importante que informações gerenciais relevantes ao negócio do cliente estejam disponíveis a ele e que possam ser acessadas seguindo um processo bem definido. O processo de notificação de anomalias de tráfego em sistemas de computação em nuvem pode fornecer informações úteis tanto para o cliente quanto para o provedor, ou mesmo a uma entidade externa que realiza auditorias de qualidade dos serviços.

Nesta tese defende-se que essas informações estejam disponíveis de modo padronizado tanto aos clientes, quanto aos provedores. Contudo, a detecção de anomalias de tráfego também é um problema, dada a escala de sistemas de computação em nuvem e a necessidade de realizar esse processamento de modo *online*.

O processo para detecção de anomalias de tráfego envolve a captura do tráfego da rede, passando pelo agrupamento dos pacotes por serviço, a aplicação de métodos para detecção de anomalias de tráfego e a notificação. A fase de detecção de anomalias é estudada neste trabalho por meio de quatro técnicas, que são a técnica baseada em monitoramento de SLAs, a técnica baseada em entropia, a baseada em aprendizagem de máquina e uma que envolve a integração das duas últimas. A fase de notificação de anomalias de tráfego proposta nesta tese envolve mais de um interessado. Uma alternativa é utilizar um **canal de publicação** de informações para assinantes, também chamado de **Publicador-Assinante**, ou *Publish-Subscriber* (P-S). Esse processo está ilustrado na Figura 1.2.

Governança de TI abrange quatro fases principais: (i) identificação das necessidades; (ii) visão da solução; (iii) planejamento da solução e (iv) implementação da solução [ao Brasileira de Normas Técnicas 2009][Grembergen e Haes 2010]. Oliveira *et al.* [Oliveira et al. 2015a] sistematizaram um processo de negócio para **governança de rede em sistemas de computação em nuvem** com base nessas fases.

Na primeira fase são levantadas as estratégias de negócio. Depois disso, é definido o

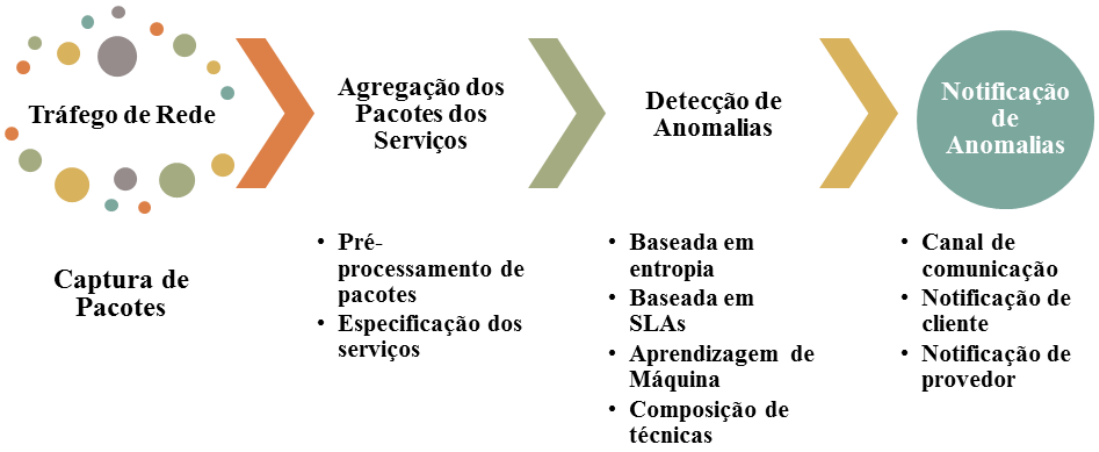


Figura 1.2: Processo de detecção de anomalias.

processo de captura de tráfego e o pré-processamento do tráfego de modo a filtrar os pacotes dos serviços que devem ser monitorados. A segunda fase consiste no monitoramento dos SLAs de cada serviço contratado e na verificação de conformidade dos mesmos. Ao ser encontrado tráfego anômalo, esses eventos são notificados. A fase 3 corresponde ao tratamento das anomalias e outros eventos de controle, bem como a tomada de ações necessárias. Clientes e provedores podem realizar seus processos de contabilidade gerencial com base nas informações obtidas, levando a um processo de tarifação confiável. Essas medidas contribuem para fomentar a confiança do cliente no provedor, bem como auxiliam o provedor a dimensionar seus recursos de modo a reduzir prejuízos. Os princípios básicos da governança de rede em computação em nuvem estão ilustrados na Figura 1.3 sob a perspectiva de COBIT [Grembergen e Haes 2010].

O ciclo de vida da governança de rede em sistemas de computação em nuvem foi definido de acordo com o ciclo de vida do gerenciamento de processos de negócios (BPM – *Business Process Management*) [Rademakers 2012]. A criação de um processo de negócio funcional envolve cinco etapas: projeto, modelagem, execução, monitoramento e otimização. Cada uma destas cinco etapas representa uma fase de desenvolvimento importante na implementação de uma solução de processo. Essas cinco fases são mostradas na Figura 1.4 e são detalhadas abaixo [Oliveira et al. 2015a]:

I. **Projeto:** a primeira fase consiste em atividades que definem o processo de negócio,

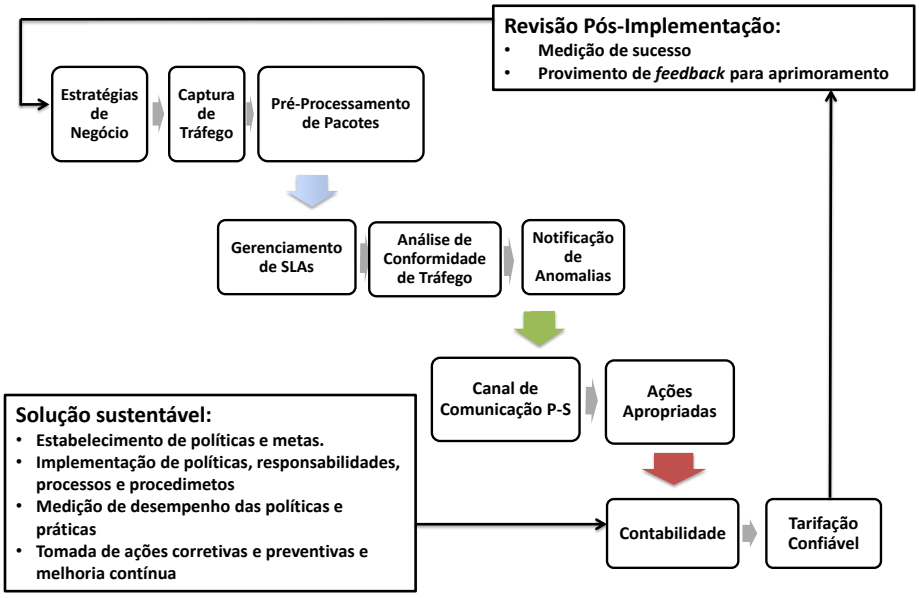


Figura 1.3: Visão geral da governança de rede para sistemas de computação em nuvem [Oliveira et al. 2015a].

como a identificação de atividades de alto nível, discussão sobre possíveis mudanças organizacionais, definição de acordos de nível de serviço e especificação de detalhes do processo, como atores, notificações e escalonamentos. Nesse passo, são identificadas as *necessidades de negócio* e a é realizado o *planejamento da solução*;

II. **Modelagem:** nesta etapa, o processo de negócio é especificado e validado. O fluxo do processo é formalizado, por exemplo, usando BPMN. As variáveis do processo são definidas e os serviços de nuvem candidatos que executarão uma atividade serão identificados e contratados;

III. **Execução:** o modelo de processo de negócio é implementado, muitas vezes usando um sistema de gerenciamento de processos de negócios (BPMS). Antes de executar o processo, é necessário adicionar detalhes técnicos para o mesmo;

IV. **Monitoramento:** os processos são monitorados de acordo com os objetivos de negócio que são assegurados por métricas definidas no SLA. Exemplos de métricas são disponibilidade e tempo de resposta e largura de banda. Nesta etapa, o provedor de serviços em nuvem realiza a verificação de conformidade do tráfego e, se detectar anomalias,

deve notificá-las via canal de comunicação compartilhado entre as partes interessadas;

V. **Otimização:** baseado em novas percepções, em mudanças nos requisitos de negócios e no monitoramento dos resultados, os processos de negócios implementados serão otimizados. Quando a fase de otimização é concluída, o processo de negócio vai para a fase de projeto novamente e o ciclo se completa.

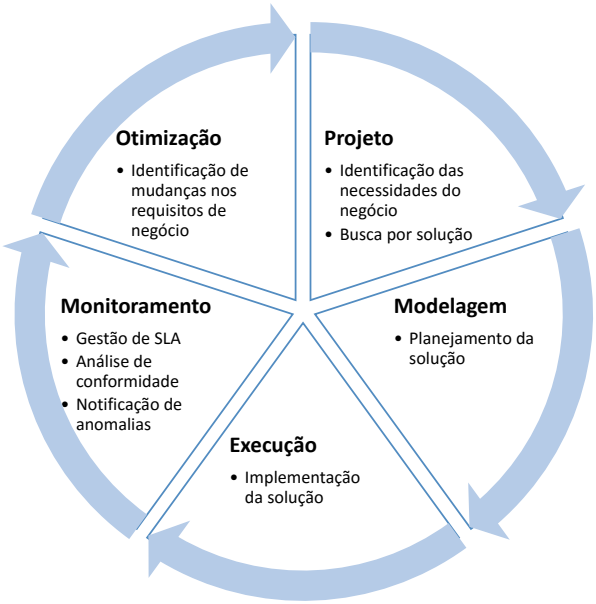


Figura 1.4: Ciclo de vida do gerenciamento de processo de negócio aplicado à governança de rede em sistemas de computação em nuvem.

1.6 Metodologia

A metodologia para solucionar o problema em estudo nesta tese envolve mais de um tipo de pesquisa. Quanto à *abordagem*, há um levantamento inicial do problema utilizando *survey* que é uma *pesquisa qualitativa* e as demais etapas possuem abordagens *quantitativas*. Quanto à *natureza*, o problema em estudo consiste em uma *pesquisa aplicada*. Quanto aos *procedimentos*, envolve *pesquisa bibliográfica* e *pesquisa experimental* [Gerhardt e Silveira 2009].

Detalhes sobre os procedimentos metodológicos empregados na tese estão apresentados nas Seções 1.6.1 e 1.6.2.

1.6.1 Pesquisa Bibliográfica

Esta tese se iniciou com uma pesquisa bibliográfica, onde foi feito um levantamento *descritivo* das referências téorico-práticas relacionadas ao tema em estudo, disponíveis por meios eletrônicos e escritos, envolvendo livros, revistas técnicas e sítios Web. A partir desse levantamento, o problema e os objetivos foram melhor delineados. As estratégias para solucionar o problema foram traçadas com base em melhoramentos nos métodos existentes.

Há citações de referências ao longo de todos os capítulos da tese, contudo elas se concentram com maior intensidade nos Capítulos 2 e 8, que versam sobre a fundamentação teórica e os trabalhos relacionados, respectivamente.

1.6.2 Pesquisa Experimental

Esta tese possui cinco capítulos e um apêndice que envolvem pesquisa experimental. O Apêndice B apresenta um estudo para caracterização da utilização de serviços de computação em nuvem com abordagem qualitativa e os demais possuem abordagem quantitativa.

O Capítulo 3 envolve o processo de definição de uma arquitetura para monitoramento de detecção de anomalias de tráfego baseado em SLA chamada TADE. Esse capítulo relata também um experimento em laboratório para avaliação do desempenho dos produtos de *software* que foram desenvolvidos para implementar a arquitetura.

Outro método para detecção de anomalias de tráfego é estudado no Capítulo 4, tendo como base o cálculo do grau de desorganização de métricas (entropia) como medida para avaliar se existe ou não uma anomalia. Neste capítulo foram encontradas limitações da técnica por meio de pesquisa experimental em laboratório do tipo simulação.

No Capítulo 5 é realizada uma pesquisa experimental para avaliar um melhoramento à técnica de detecção baseada em entropia por meio de uma técnica híbrida, que inclui aprendizagem de máquina ao processo, envolvendo simulação e análise de tráfego de um arquivo de rastro contendo anomalias provocadas por um ataque de negação de serviço real.

Para determinar prejuízos causados por anomalias de tráfego em serviços de computação em nuvem, é necessário conhecer o custo de tais serviços. No Capítulo 6 é introduzido o modelo de precificação para Dados-como-Serviço (DaaS) que é avaliado por meio de simulação. Neste estudo é feita uma análise exploratória das variáveis que compõem o modelo e

que causam impacto no custo total.

O modelo de custo de DaaS introduzido é utilizado como estudo de caso no Capítulo 7. O objetivo deste estudo é identificar prejuízos causados por anomalias de tráfego ao negócio. Neste capítulo foi realizado um experimento real para execução de uma aplicação de DaaS com o objetivo de calibrar as variáveis que compõem o modelo a partir de um estudo prático. Após isto, foram realizadas simulações para análise de custo-volume-lucro (CVP).

1.7 Contribuições e Publicações

As contribuições deste trabalho de doutorado estão parcialmente associadas a submissões de artigos a conferências técnicas nacionais e internacionais e a revistas especializadas. As contribuições, seguidas dos artigos produzidos, estão listadas abaixo:

I. **Arquitetura e sistema de detecção de anomalias:** o projeto de uma arquitetura para detecção e gerenciamento de anomalias de tráfego para serviços de computação em nuvem baseada em acordos de nível de serviço que possui uma API aberta e acessível aos clientes. Essa arquitetura foi implementada, foi validada e disponibilizada como ferramenta de código aberto. As publicações relacionadas são:

   (a) *"Efficient network service level agreement monitoring for cloud computing systems"* (ISCC 2014) - Qualis A2 [Oliveira et al. 2014a]

   (b) *"Just-in-Time Clouds: An Approach to Federate Private Clouds"* (Salão de Ferramentas - SBRC 2013) - Qualis B2 [Fraga et al. 2013]

   (c) *"Desenvolvimento de Mecanismos para Melhoria do Desempenho da Análise de Métricas de Rede em Tempo Real"* (CONNEPI 2013) - Sem Qualis [Chagas e Oliveira 2013]

II. **Técnica de detecção de anomalias híbrida empregando entropia e aprendizagem de máquina:** um mecanismo complementar de análise de tráfego para os serviços que não necessitam possuir SLAs contratados, ou onde estes são desconhecidos, que realiza a análise do comportamento do tráfego. As publicações relacionadas são:

(a) *"Improving Network Traffic Anomaly Detection for Cloud Computing Services"* (ICSNC 2014) - Qualis B3 [Oliveira et al. 2014b]

(b) *"Implementação de um Detector de Anomalias de Tráfego de Rede Baseado na Entropia de Métricas para Sistemas de Computação em Nuvem"* (CONNEPI 2012) - Sem Qualis [Oliveira e Ferreira 2012]

III. **Caracterização da utilização de serviços de computação em nuvem:** foi realizada uma pesquisa para caracterização da utilização de serviços de computação com o intuito de fornecer uma visão geral sobre modelos de negócio e métricas de desempenho aplicadas a serviços de computação em nuvem:

(a) *"From the Dark Net to the Cloudy Data: Guidelines to Cloud Network Governace"* (SCCC 2015) - Qualis B3 [Oliveira et al. 2015a]

IV. **Modelo de precificação e análise do custo de anomalias:** para realizar um processo de tarifação confiável do tráfego, é necessário, além da investigação de pontos de desvio do comportamento esperado do tráfego, um modelo de precificação que leva em consideração anomalias de tráfego e que essas informações estejam disponíveis ao cliente para sua conferência e eventual prova de utilização, caso necessário. As publicações relacionadas são:

(a) *"Optimizing Query Prices for Data-as-a-Service"* (Aprovado para o SERVICES 2015 e publicado no IEEE BigData Congress 2015) - Qualis B2 [Oliveira et al. 2015b]

(b) **(Convite para versão estendida)** *"Cost-based Virtual Machine Scheduling for Data-as-a-Service"* (IJBD – *International Journal on BigData*) - Qualis ainda não definido [Oliveira et al. 2015c].

De modo complementar, a candidata estabeleceu parcerias em decorrência de trabalhos conjuntos em disciplinas do doutorado resultando na coautoria de artigos científicos na área de Ciência da Computação, que enriqueceram a experiência proporcionada pelo curso. Destacam-se as seguintes publicações:

1. *Selecting Frameworks For Multi-Agent Systems Development For The Oil Industry - Revista Cubana de Ciencias Informáticas* - Sem Qualis [Moura et al. 2015]

2. *Selecting Frameworks for Multi-Agent Systems Development for the Oil Industry* - CIIISI 2014 - Sem Qualis [Dóra et al. 2014a]

3. *Simultaneously Improving Quality and Time-to-Market in Agile Development* - Capítulo de Livro publicado pela Springer sobre Tecnologias de *Software* (*Communications in Computer and Information Science*) - Qualis B4 [Dóra et al. 2014b]

4. *A Baseline for Quality Management in Software Projects* - CICCI 2013 - Sem Qualis [Dóra et al. 2013a]

5. *Improving Quality in Agile Development Processes* - ICSOFT-EA 2013 - Qualis B4 [Dóra et al. 2013b]

1.8 Estrutura do Documento

Esta tese de doutorado está estruturada em oito capítulos. No Capítulo 1 foram apresentadas a contextualização, a problematização, os objetivos deste trabalho e as publicações relacionadas a esse estudo. A fundamentação teórica para embasar o entendimento do documento encontra-se no Capítulo 2. As contribuições técnicas deste trabalho de tese são relatadas nos Capítulos 3, 4, 6 e 7. A revisão da literatura está presente no Capítulo 8. Por fim, as conclusões e discussão sobre trabalhos futuros estão contidas no Capítulo 9. Convenções textuais adotadas na tese estão listadas no Apêndice A. Uma pesquisa para caracterização de serviços de computação em nuvem que auxilia no entendimento da motivação deste trabalho é apresentada no Apêndice B. Os Apêndices C e D contêm material complementar sobre o processo de desenvolvimento dos produtos de software da arquitetura TADE, que representam o levantamento de requisitos e os casos de uso, respectivamente.

Capítulo 2

Fundamentação Teórica

"A inteligência é caracterizada por uma incompreensão natural da vida."

Bergson

2.1 Computação em Nuvem

A comunicação para fins militares baseada nas redes de computadores em sua concepção, deu espaço a inúmeros tipos de aplicações e serviços. Com o passar dos anos, a infraestrutura para equipamentos de redes de computadores passou a prover serviços de comunicação global em altíssima velocidade, aumentando a escala e o escopo de utilização.

Cloud computing, ou **computação em nuvem**, é um tipo de serviço que foi viabilizado graças à evolução das redes de computadores. A computação em nuvem provê uma infraestrutura virtual para prestação de serviços em rede sob demanda. Os clientes contratam serviços em que a infraestrutura primária de *hardware* e *software* encontra-se em centros de dados (*data centers*) remotos e não necessariamente geograficamente próximos e sob o domínio do cliente. Um centro de dados geralmente contém milhares de servidores que são organizados em *racks* e interconectados por meio de comutadores e roteadores [Mikkilineni e Sarathy 2009].

Uma nuvem, ou *cloud*, é definida como o *hardware* e o *software* de um *data center*. Quando os serviços de uma nuvem são oferecidos abertamente aos clientes e estes pagam por sua utilização, essa nuvem é considerada uma **nuvem pública** e os serviços são chamados

de **Computação Utilitária** ou *Utility Computing*. Uma **nuvem privada** é uma infraestrutura proprietária em que os serviços não estão disponíveis ao público geral [Armbrust et al. 2009].

A computação utilitária é uma ideia antiga foi retomada e está sendo utilizada por meio da construção de sistemas distribuídos interoperáveis e do compartilhamento de recursos entre nuvens que permitem o surgimento de sistemas de ultra-larga escala, agregando transparentemente potência computacional e armazenamento de modo que tais recursos aparentem ser ilimitados [Parkhill 1966].

Segundo Armbrust *et al.* [Armbrust et al. 2009], a computação em nuvem fornece uma visão de que há:

I. Infinitos recursos computacionais disponíveis sob demanda;

II. Eliminação do compromisso de ter super-provisionamento de recursos em um centro de dados corporativo, por exemplo, enquanto não existir demanda por parte dos usuários;

III. Pagamento por utilização com recursos disponíveis em um curto prazo. Empresas estão investindo em prover infraestrutura de computação em nuvem, cobrando os serviços virtuais de processamento, armazenamento e rede por tempo de utilização.

Segundo Voas e Zhang [Voas e Zhang 2009], a computação em nuvem pode ser entendida como uma evolução dos serviços de telecomunicações e uma retomada ao antigo modelo cliente-servidor com *thin-client* acessando um *mainframe*. Com a evolução, o *mainframe* tornou-se a própria Internet, passando a ideia de capacidade ilimitada de recursos. Conforme ilustrado na Figura 2.1, os paradigmas de computação evoluíram dos terminais para computadores pessoais, às redes locais, à Internet, à computação em grade e computação em nuvem.

Buyya [Buyya 2009] define a computação utilitária como o resultado de alguns paradigmas computacionais incluindo a computação em grade, a computação Par-a-Par (P2P - *Peer-to-Peer*) e a computação em nuvem, onde os usuários requisitam serviços computacionais sob demanda, de forma semelhante às utilidades básicas dos serviços de água, eletricidade e gás, não importando a localização de tais serviços. Buyya acrescentou ainda que empresas como a Amazon, HP, IBM e Sun Microsystems (atualmente de propriedade da Oracle) estão criando e implantando nuvens em vários lugares do mundo, favorecendo a criação de um

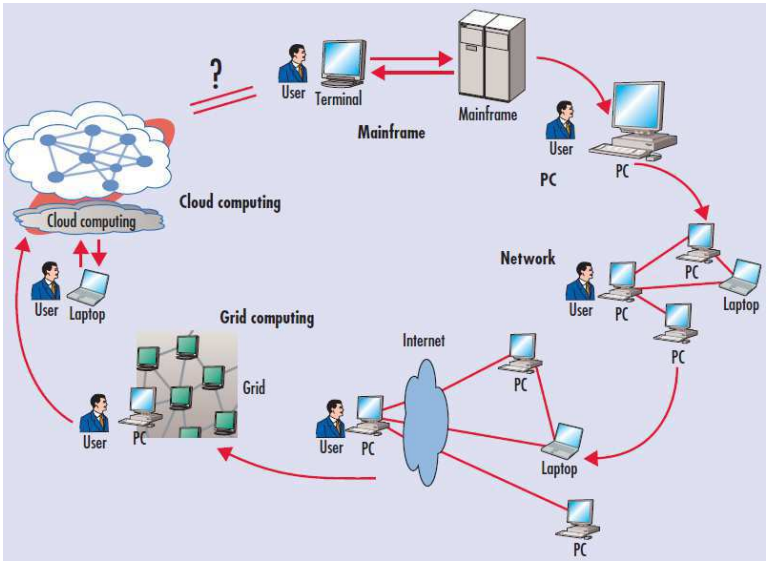


Figura 2.1: Evolução dos paradigmas de computação em seis fases distintas [Voas e Zhang 2009].

ambiente para interconexão e provisionamento de recursos dinâmicos. No entanto, ateve-se à conceitualização e não entrou em detalhes sobre as especificações de como cada serviço deve ser padronizado e oferecido.

Os custos para realizar a implantação de um centro de dados físico e do seu gerenciamento e suporte são altos, especialmente se a infraestrutura for provisionada para acomodar os momentos de utilização de pico. O tempo para iniciar as operações também pode ser muito elevado. Sistemas de computação em nuvem podem prontamente escalar para atender a demanda de recursos no momento [Grossman 2009].

A computação em nuvem também é mencionada na literatura como **computação elástica**, haja vista que o conjunto de recursos alocado pode variar dinamicamente. Os serviços S3 (*Scalable Storage Service*) e EC2 (*Elastic Computing Cloud*) da Amazon empregam esse modelo. O provimento de serviços na computação em nuvem possui duas categorias distintas [Grossman 2009]:

I. Provimento de instâncias computacionais sob demanda;

II. Provimento de capacidade computacional sob demanda.

No primeiro caso, o aumento da escala dá-se por prover unidades de processamento adicionais, enquanto que na segunda opção a capacidade computacional é projetada para

escalar de acordo com a demanda intensiva de dados e computação das aplicações.

Três fatores que contribuem para o crescimento da computação em nuvem [Grossman 2009] são:

I. Escala;

II. Simplicidade;

III. Preço.

Weinhardt et al.[Weinhardt et al. 2009] ressaltam que linhas de pesquisa em computação em nuvem focarão na padronização de uma API, em questões de segurança, novos modelos de negócios e sistemas de precificação dinâmicos para atender a serviços complexos. Exemplos de serviços de infraestrutura em computação em nuvem são armazenamento (*storage*) e processamento (*computing*), como podem ser vistos na Figura 2.2, bem como outros exemplos de serviços por camada.

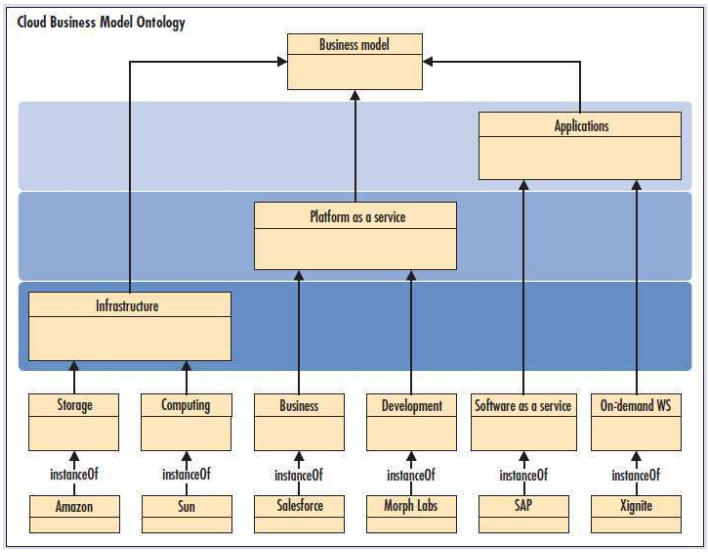


Figura 2.2: Arquitetura de computação em nuvem e exemplos de serviços oferecidos por cada camada [Weinhardt et al. 2009].

Armbrust *et al.* [Armbrust et al. 2010] apontam os dez principais obstáculos e oportunidades com foco na computação em nuvem:

I. Continuidade de negócio e disponibilidade de serviço;

II. Bloqueio de dados;

III. Confidencialidade e auditoria dos dados;

IV. Gargalos na transferência dos dados;

V. Imprevisibilidade de desempenho;

VI. Armazenamento escalável;

VII. Falhas em sistemas distribuídos de larga escala;

VIII. Rapidez para escalar o sistema;

IX. Compartilhamento de reputação;

X. Licenciamento de *software*.

A elasticidade dos sistemas de computação em nuvem é uma característica bem tratada pelos esquemas de pagamento sobre o que for utilizado. Os sistemas devem ter ciência do que está executando ou não nas máquinas virtuais. O monitoramento e a tarifação devem acompanhar a execução do sistema desde o princípio do funcionamento (*pay-for-use*). A cobrança dos serviços prestados deve considerar os custos do sistema, em termos da especificação do *hardware*, e custos proporcionais ao trabalho do sistema, como a utilização de memória ou energia elétrica [Armbrust et al. 2010].

Clientes de serviços de computação em nuvem não possuem conhecimento e nem domínio sobre a infraestrutura de rede do provedor, uma vez que eles não têm acesso à topologia física da rede, bem como aos roteadores e comutadores que encaminham os pacotes de e para suas aplicações. Por conseguinte, os clientes também não podem utilizar soluções que melhorem o desempenho de suas aplicações por meio de ações sobre o tráfego da rede como, por exemplo, utilizar difusão de mensagens entre todos os nós que lhes foram alocados ou para um grupo deles. Caso um cliente de computação em nuvem tenha domínio sobre a estrutura da rede, ou ao menos à parte da rede que é utilizada para realizar a comunicação entre as máquinas que fazem parte do conjunto de máquinas virtuais alocadas aos clientes. Um ponto positivo disso é que os clientes poderiam implementar mecanismos de engenharia de tráfego, por exemplo, para melhorar a utilização de largura de banda e, consequentemente, o

desempenho de suas aplicações. Por outro lado, estão envolvidas questões de segurança da máquina física que poderiam afetar os serviços dos demais inquilinos dessa máquina.

2.1.1 Arquitetura de Sistemas de Computação em Nuvem

Zhang e Zhou [Zhang e Zhou 2009] propuseram uma **Arquitetura Aberta para Computação em Nuvem** (CCOA - *Cloud Computing Open Architecture*) com base em sete princípios arquiteturais e em dez modelos de arquitetura. Esse trabalho reune as potencialidades da Arquitetura Orientada a Serviços (SOA - *Service Oriented Architecture*) e da virtualização para o valor de negócio e o valor prático para produtos de *software* emergentes, de *hardware* e processos de negócio para provisionamento de serviços na Internet no contexto de computação em nuvem. A arquitetura tem foco em ser extensível e configurável, para que possa prover orientação normativa na definição de infraestrutura, na especificação de *software*, aplicativos e compartilhamento de processos de negócios de uma maneira unificada.

Zhang e Zhou [Zhang e Zhou 2009] identificaram três objetivos que norteiam a estruturação de uma boa arquitetura para computação em nuvem e definiram uma arquitetura aberta, que associaram à ideia de um modelo "OSI"[1] para computação em nuvem. Para que se obtenha uma boa arquitetura para um sistema de computação em nuvem, devem ser alcançados os seguintes objetivos:

1. **Objetivo 1:** Desenvolver uma forma reutilizável para a criação de plataforma de provisionamento escalável e configurável para computação em nuvem;

2. **Objetivo 2:** Propor um conjunto de serviços comuns e compartilhados para a construção de plataformas de computação em nuvem, quer seja para prestação de serviços de negócios (a um usuário final) ou outros serviços de nuvem para consumidores empresariais empregando uma abordagem unificada;

3. **Objetivo 3:** Maximizar o valor potencial do negócio na computação em nuvem com base em uma infraestrutura de TI extensível e no sistema de gestão. A ideia principal

[1] O modelo OSI (*Open Systems Interconnection*) é um modelo em sete camadas que define as funções de todos os serviços que devem ser providos para que ocorra a comunicação entre dois processos remotos por meio da rede de comunicações, desde a parte mecânica-fisíca da transmissão dos bits até o recebimento dos dados na aplicação remota.

é monetarizar o valor agregado dos serviços da nuvem de negócios, combinando as potencialidades da SOA e da computação em nuvem.

Os sete princípios arquiteturais propostos por Zhang e Zhou [Zhang e Zhou 2009] para a CCOA são:

1. **Princípio 1:** Gerenciamento integrado do ecossistema da nuvem. Componentes arquiteturais: gerenciamento do ecossistema da nuvem, painel do vendedor da nuvem, painel do parceiro da nuvem, painel do cliente da nuvem;

2. **Princípio 2:** Virtualização para a infraestrutura da nuvem de *hardware* ou *software*. Componentes arquiteturais: gerenciamento da infraestrutura de TI da nuvem e a infraestrutura de núcleo da nuvem;

3. **Princípio 3:** Reutilização de serviços comuns. Componentes arquiteturais: serviços de negócios horizontais e serviços de negócio verticais;

4. **Princípio 4:** Provisionamento e assinatura de acordos extensíveis. Componentes arquiteturais: serviço de provisionamento da nuvem e o serviço de assinantes da nuvem;

5. **Princípio 5:** Habilitação configurável dos serviços oferecidos pela nuvem. Componente arquitetural: valor agregado dos serviços da nuvem;

6. **Princípio 6:** Unificação da representação da informação e de um arcabouço para troca de informações. Componente arquitetural: arquitetura de informação da nuvem;

7. **Princípio 7:** Qualidade e governança da nuvem, responsável por monitoramento, QoS, medição, cobrança e tratamento de exceções. Componente arquitetural: qualidade e governança da nuvem.

2.1.2 Modelos de Negócio de Computação em Nuvem

A computação em nuvem emprega um modelo de negócio orientado a serviço. Como em outros modelos arquiteturais de rede, afirma-se que a camada inferior provê um serviço à camada superior; ou ainda, que a camada superior é cliente da camada inferior. De modo geral, classificam-se os modelos de negócio em três categorias [Zhang et al. 2010], conforme listagem a seguir.

1. **Infraestrutura-como-serviço:** ou *Infrastructure-as-a-Service* (IaaS), é o modelo de negócio de nível mais baixo. O IaaS refere-se ao provisionamento de recursos de infraestrutura sob demanda, geralmente em termos de VMs. Exemplos de provedores de IaaS: Amazon EC2, GoGrid e Flexiscale;

2. **Plataforma como serviço:** ou *Platform-as-a-Service* (PaaS), provê os serviços que ficam situados no nível de suporte a sistemas operacionais e arcabouços para desenvolvimento de *software*. Exemplos de provedores de PaaS incluem o Google App Engine, a Microsoft Windows Azure e Force.com;

3. **Software-como-Serviço:** ou *Software-as-a-Service* (SaaS), onde os provedores de SaaS fornecem aplicações sob demanda por meio da Internet. Exemplos de provedores de SaaS: Salesforce.com, Rackspace e SAP Business ByDesign.

Na prática, frequentemente os provedores de IaaS e PaaS são uma mesma organização, como Google e Salesforce. Por essa razão, provedores de PaaS e IaaS são comumente generalizados como provedores de infraestrutura ou provedores de nuvem. A arquitetura de sistemas de computação em nuvem com a equivalência dos modelos de negócio abrangidos em cada camada é resumidamente ilustrada na Figura 2.3.

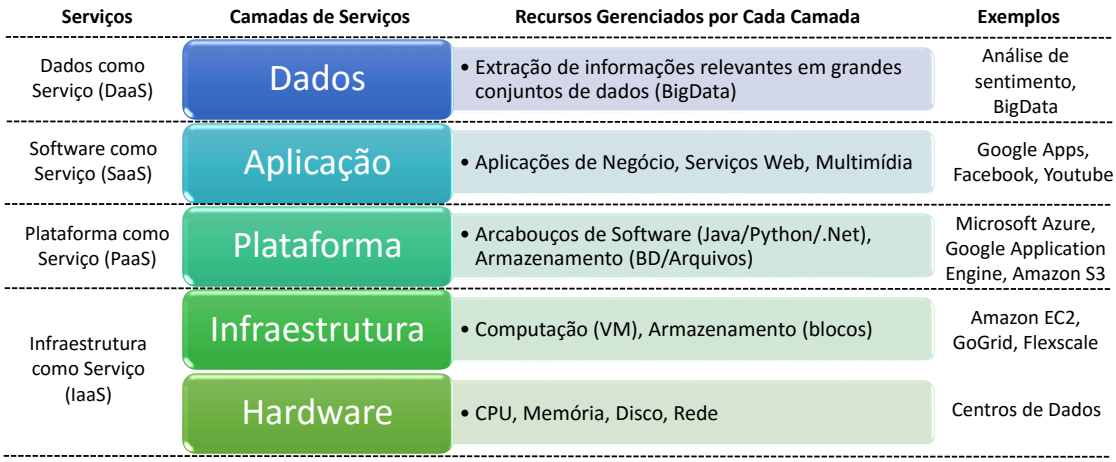


Figura 2.3: Modelos de negócio de computação em nuvem. Adaptado de [Zhang et al. 2010].

A arquitetura de um ambiente de computação em nuvem foi originalmente dividida em quatro camadas: a **camada de hardware** ou *centro de dados*, a **camada de infraestrutura**, a **camada de plataforma** e a **camada de aplicação**. Há uma separação de conceitos entre cada camada, com baixo acoplamento entre elas, como ocorre no modelo OSI, conforme as funcionalidades descritas abaixo [Zhang et al. 2010]:

1. ***Hardware*:** a camada de *hardware* é implementada em centros de dados. Esta camada é responsável pelo gerenciamento dos recursos físicos da nuvem, incluindo servidores físicos, roteadores, comutadores, energia e sistemas de refrigeração. Problemas típicos da camada de *hardware* incluem a configuração de *hardware*, tolerância a falhas, gerenciamento de tráfego, de energia e de resfriamento;

2. **Infraestrutura:** a camada de infraestrutura cria um conjuntos de recursos de armazenamento e computação, organizando e disponibilizando os recursos físicos por meio de tecnologias de virtualização como Xen, Eucalyptus, OpenStack, KVM e VMware. A camada de infraestrutura também é conhecida como a camada de virtualização e provê funcionalidades chaves, como a atribuição dinâmica de recursos;

3. **Plataforma:** o objetivo da camada de plataforma é minimizar a carga de implantação de aplicativos diretamente em recipientes VM. A camada de plataforma consiste de sistemas operacionais e estruturas de aplicativos. Por exemplo, o Google App Engine opera na camada de plataforma para fornecer API de suporte para a implementação de banco de dados, armazenamento e de lógica de negócios de aplicações web típicas.

4. **Aplicação:** a camada de aplicação consiste nas aplicações em nuvem reais. Aplicações em nuvem pode alavancar o recurso de dimensionamento automático para obter melhor desempenho, disponibilidade e menor custo operacional.

Novos modelos foram propostos para dar suporte a esses três tipos básicos de modelos de negócio, envolvendo a virtualização de redes, dados e alguns tipos de serviços bem mais específicos. Atualmente fala-se em **"Qualquer coisa-como-serviço"** (*Everything-as-a-Service* – XaaS). Para fornecer uma visão geral, alguns desses modelos serão apresentados nas próximas seções.

Everything-as-a-Service

Há diversas classificações para serviços de computação em nuvem. Duan *et al.* [Duan et al. 2015] realizaram uma pesquisa exploratória para identificar a expressão "como-serviço" nas bibliotecas digitais DBLP, ACM Digital Library e IEEE Xplore Digital Library. Eles encontraram essa expressão em diversos contextos além dos originais, como Rede-como-serviço, Banco de Dados-como-Serviço, Análise-como-Serviço, Roteamento-como-Serviço, Modelagem-como-serviço, dentre outros. Ainda falta na literatura uma sistematização explícita que trate todos esses contextos e as relações entre cada um deles. Diante desse cenário, surgiu a expressão **"Qualquer coisa-como-serviço"** (*Everything-as-a-Service* – XaaS).

Os serviços de computação em nuvem originais e o conceito de Dados e de Rede-como-Serviço serão discutidos nas próximas seções.

Software-as-a-Service

O conceito de SaaS pode ser confundido com o conceito de SOA. Laplante et al. [Laplante et al. 2008] caracterizaram e diferenciaram os termos SaaS e SOA. O primeiro termo diz respeito a um modelo para **entrega de *software***, ao passo que o segundo representa um modelo para o **desenvolvimento de *software***. Embora diferentes, são termos que podem ser usados como apoio um para o outro. O modelo de SaaS separa a propriedade sobre o software do usuário, sendo de propriedade do provedor de serviços e vendidos ao usuário sob demanda (e.g., *salesforce.com*). A arquitetura SOA, por sua vez, define os componentes de software como serviços reusáveis.

Na computação em nuvem, o termo **serviço** pode variar bastante de contexto, bem como as referências entre **provedor** e **cliente** dependendo do foco da camada arquitetural que está sendo focada. Como mostrado na Figura 2.4 [Cusumano 2010], existe o papel de um **provedor de nuvem** (ou *cloud provider*) que representa o provedor de infraestrutura, o papel de um **usuário da nuvem** (*cloud user*) que é cliente do serviço de infraestrutura para prover aplicações, atuando também um provedor de SaaS (*SaaS provider*). O terceiro ator da figura representa o cliente final (*SaaS user*) que acessa as aplicações Web providas.

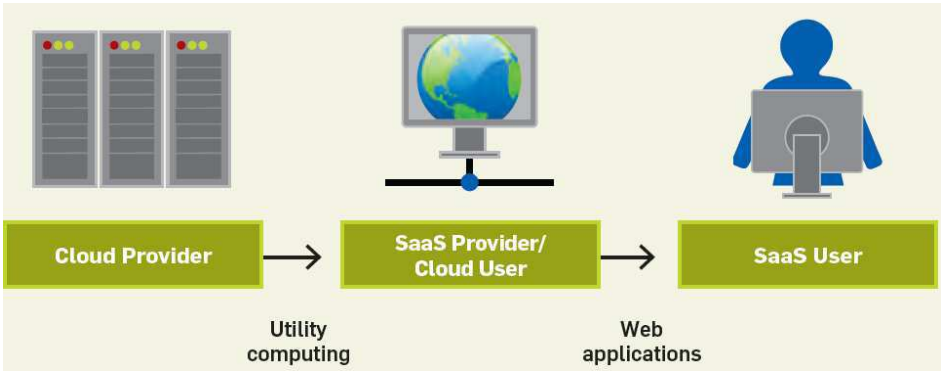


Figura 2.4: Diferentes visões entre cliente e provedores de serviços de SaaS [Cusumano 2010].

Platform-as-a-Service

Os provedores de PaaS ofertam serviços não apenas de máquinas virtuais, mas também bases de dados para armazenamento de informações do negócio, ferramentas de desenvolvimento, implantação e manutenção de novas aplicações na plataforma, além de um arcabouço para realizar testes, estabelecer a comunicação entre máquinas, realizar monitoramento dos recursos utilizados e hospedagem de aplicações, cujo acesso é independente de plataforma. Exemplos de provedores de PaaS são o Oracle Cloud Platform [Oracle 2015], o Windows Azure [Microsoft 2013] e a Plataforma Salesforce [Salesforce 2015].

Infrastructure-as-a-Service

Infraestrutura como Serviço (IaaS) é um tipo de plataforma de computação em nuvem em que o cliente terceiriza sua infraestrutura de TI, incluindo armazenamento, processamento, redes e outros recursos. Com IaaS, os clientes podem aumentar a escala do uso de recursos e suas configurações dinamicamente para atender às necessidades do negócio e são cobrados apenas pelos serviços que forem efetivamente utilizados. Em serviços tradicionais de hospedagem, por exemplo, a infraestrutura de TI é alugada para períodos específicos de tempo e o cliente paga pelo tempo e a configuração do hardware, independentemente do uso [IBM 2015].

Dentre as empresas que provêem IaaS, tem-se o modelo de Infraestrutura-como-Serviço da Amazon Web Services [Amazon 2015], o IBM *infrastructure as a service* [IBM 2013], o Rackspace [Rackspace 2015] e a GoGrid [GoGrid 2015].

Data-as-a-Service

Durante a última década, houve um crescimento contínuo e crescente de fontes de dados, como aplicativos para telefones inteligentes, sensores, *blogs* e portais. O tratamento desses dados permite que as empresas ganhem uma compreensão mais profunda sobre os seus clientes e padrões de comportamento, importante para sistemas de recomendação e detecção de fraudes, por exemplo. Grandes empresas como Google e Yahoo! têm acesso a essas fontes de dados passivas, uma vez que elas próprias realizam rastreamento e processamento contínuo da Web. Contudo, pequenas a médias empresas (PMEs) não têm acesso a esses dados. Adquirir esses novos dados não só exige o uso de uma tecnologia apropriada para rastrear, armazenar e processar os dados, mas também recursos computacionais e de armazenamento necessários para desempenhar essas tarefas. Rastrear "toda" a Web e processar esses dados ainda é uma tarefa difícil para muitas PMEs, pois geralmente não dispõem da tecnologia necessária para o rastreamento, armazenamento e processamento dos dados e manter uma cópia de "toda" a Web requer uma enorme quantidade de recursos que muitas vezes não podem ser oferecidos por uma única PME [Oliveira et al. 2015b].

Diante desse problema, emergiu um novo modelo de negócio para computação em nuvem chamado *Data-as-a-Service* (DaaS), no qual várias PMEs partilham os custos necessários de infraestrutura para rastrear e armazenar uma cópia de "toda" a Web (ou pelo menos uma grande parte dela) e executar consultas específicas sobre negócios em um determinado conjunto de dados. Para representar a infraestrutura computacional dos sistemas de DaaS, será adotado um modelo não homogêneo de **micronuvem**, em vez de um único centro de dados dedicado para realizar o rastreamento, armazenamento e tratamento de dados. Micronuvens consistem em pequenas unidades de nós como, por exemplo, 20 máquinas físicas, com diferentes capacidades de armazenamento e processamento [Oliveira et al. 2015b].

Network-as-a-Service

Rede-como-Serviço, ou *Network-as-a-Service* (NaaS), é um modelo de serviço de computação em nuvem que possibilita ao cliente conhecer a topologia e os equipamentos de interconexão que fazem a comunicação entre as máquinas virtuais em sua "rede virtual". [Costa et al. 2012]. Algumas aplicações podem ter seu desempenho melhorado por meio de mani-

pulações de pacotes na rede. Por exemplo, quando um professor está fazendo uma difusão de uma videoaula para um grupo de alunos via rede (*multicast*), o uso da largura de banda poderia ser melhorado, evitando desperdício de banda, caso apenas um fluxo de pacote fosse enviado da máquina do professor e apenas nos roteadores mais próximos aos alunos esse fluxo de pacote fosse replicado, seguindo o caminho dos enlaces que conduzem até as máquinas dos alunos. Da mesma forma, caso um mesmo fluxo de pacotes fosse, por algum motivo, duplicado por dois caminhos distintos e, posteriormente, se encontrassem em um determinado roteador, apenas um fluxo poderia ser encaminhado, reduzindo o tráfego de saída do roteador.

De acordo com Costa *et al.* [Costa et al. 2012], algumas aplicações podem ter seu desempenho melhorado por meio do uso de NaaS, fazendo-se necessário um levantamento dos requisitos de rede que cada uma delas apresenta para que soluções de NaaS possam ser efetivamente implementadas. O volume de memória requerido dentro da rede varia entre aplicações. Por exemplo, realizar ***caching*** dentro da rede para reduzir o volume de tráfego, o **processamento de eventos complexos** e o **processamento de fluxos** são exemplos de aplicações que necessitam memória na ordem de MBs ou GBs, porque precisam manter dados de aplicações (i.e., armazenamento de pacotes) por longos períodos de tempo. **Agregação de dados**, ***firewalls*** e **redes baseadas em conteúdo** precisam de um volume menor de memória, pois armazenam apenas resultados temporários ou regras de policiamento. O volume de aplicações como **escalonamento de pacotes**, ou engenharia de tráfego de pacotes, **roteamento com múltiplos caminhos** e **difusão** (*broadcast* ou *multicast*) possuem, comparativamente, requisitos de memória inferiores, porque memorizam apenas pequenas tabelas de roteamento e estatísticas por fluxos.

Para que o modelo de NaaS possa ser implementado pelos clientes no nível de aplicação, urge que três funcionalidades sejam disponibilizadas ao desenvolvedor:

I. **Visibilidade da rede:** pode-se utilizar soluções proprietárias para isto, geralmente, com uma baixa precisão para realizar o mapeamento da rede, no entanto, uma solução precisa e de baixo custo seria o provedor do centro de dados disponibilizar as informações sobre a topologia da rede aos clientes. Neste caso, apenas a topologia da fatia, ou rede virtual, na qual suas máquinas virtuais estão se comunicando;

II. **Encaminhamento de pacotes customizado:** a segunda funcionalidade que o modelo de NaaS pode prover é a habilidade de controlar o encaminhamento de pacotes em roteadores e comutadores;

III. **Processamento dentro da rede:** outra funcionalidade do modelo de NaaS é a possibilidade de realização de processamento de pacotes dentro da rede e não apenas nos sistemas finais.

Para prover as funcionalidades citadas e desenvolver um modelo de NaaS bem sucedido para um centro de dados, os seguintes requisitos devem ser atendidos:

I. **Integração com o hardware atual dos centros de dados:** para que NaaS possa ser implementado na prática, deve ser possível utilizar os hardwares atualmente disponíveis nos centros de dados e não demandar a compra de equipamentos caros;

II. **Modelo de programação de alto nível:** a implementação de aplicações cientes de NaaS deve utilizar um modelo de programação comum aos já utilizados pelos desenvolvedores, escondendo os detalhes de rede sem requerer programação de baixo nível;

III. **Escalabilidade e isolamento de múltiplos inquilinos:** o modelo de NaaS deve ser capaz de dar suporte a diversos tipos de aplicações, fazendo um isolamento forte entre os recursos de rede de diversos clientes (inquilinos) e de modo escalável.

2.1.3 Classificação quanto ao Domínio Administrativo

As nuvens podem compreender soluções públicas ou privadas, ou ainda uma combinação desses dois modelos, de acordo com restrições de custo e de requisitos de qualidade que os cliente demandem para os serviços. Nesse sentido, pode-se identificar quatro tipos de nuvens [Zhang et al. 2010]:

1. **Nuvens públicas:** os serviços prestados dentro de nuvens públicas podem ser contratados pelo público geral. Seus clientes podem aproveitar o benefício de não precisar investir em infraestrutura. Neste caso, os riscos de investimento localizam-se com os provedores de infraestrutura. A desvantagem de utilizar nuvem públicas é que os clientes não possuem controle sobre a localização do armazenamento de dados, sobre

o tráfego de rede e faz-se necessário confiar que seus dados e negócio estão seguros. Essas são razões para alguns clientes não optarem por esse modelo.

2. **Nuvens privadas:** as nuvens privadas são implantadas para serem utilizadas por uma única organização, também chamadas de nuvens internas. Esse tipo de nuvem pode ser implantada e gerenciada pela própria organização que decide aplicar virtualização de recursos ou por um provedor externo. Este tipo de nuvem oferece um alto grau de controle sobre desempenho, confiabilidade e segurança. A principal desvantagem diz respeito a custos, pois neste tipo de nuvem é necessário realizar investimentos em infraestrutura, assemelhando-se ao modelo anterior de computação e perdendo alguns benefícios disponibilizados pelas nuvens públicas.

3. **Nuvens híbridas:** uma nuvem híbrida é uma combinação entre os modelos de nuvem pública e privada. Em uma nuvem híbrida, parte da infraestrutura de serviço é situada em nuvens privadas, enquanto a parte restante é executada em nuvens públicas. Nuvens híbridas oferecem mais flexibilidade do que nuvens públicas e privadas. Quando há requisitos de confidencialidade forte, por exemplo, este tipo de nuvem provê maior controle e segurança sobre os dados de aplicativos em comparação com as nuvens públicas. Os pontos fracos deste tipo de nuvem diz respeito ao cuidado para determinar a melhor separação entre os componentes de nuvem pública e privada. Os custos com infraestrutura e os requisitos dos serviços nortearão a escolha de quais dados devem fazer parte da nuvem privada ou pública.

4. **Nuvem Privada Virtual:** as nuvens privadas virtuais (VPC – *Virtual Private Cloud*) são uma solução alternativa para resolver as limitações de nuvens públicas e privadas. A VPC é uma plataforma rodando em cima de nuvens públicas. Para realizar a comunicação dentro da VPC, emprega-se a tecnologia de redes privadas virtuais (VPN – *Virtual Private Network*) que criam um túnel seguro para transferência de dados entre as máquinas virtuais da VPC que rodam sobre a infraestrutura de uma rede pública. Dessa feita, não é necessário que o cliente possua capital para fazer caros investimentos em infraestrutura e que possam se tornar também prestadores de serviços em um outro nível. Por exemplo, clientes de IaaS podem utilizar a infraestrutura contratada para proverem aplicações (SaaS), ou até mesmo infraestrutura. Esse tipo de nuvem permite

que esses clientes-provedores possam projetar sua própria topologia e configurações de segurança graças à virtualização da camada de rede.

Quando é mais indicado adotar um modelo de computação em nuvem privada, ou quando é melhor optar por serviços em uma nuvem pública? Armbrust *et al.* fizeram um levantamento de indícios que podem ser apresentados como justificativa para que se opte pela computação utilitária [Armbrust et al. 2009]:

I. Quando a demanda por serviços for muito variável no tempo, por exemplo, superestimar um centro de dados para poder suportar apenas alguns dias de processamento de pico gera subutilização dos recursos disponíveis;

II. Quando a demanda não for conhecida;

III. Quando o processamento paralelo puder ser utilizado para agilizar a conclusão de uma aplicação, como ocorre na computação em grade.

2.1.4 Federação de Nuvens

Uma nuvem federada reune recursos computacionais por meio de uma rede sobreposta (*overlay*) às redes de diversos domínios para agregar recursos computacionais dessas diversas entidades em um único sistema de computação em nuvem. Uma nuvem federada é composta por recursos computacionais de centros de dados de vários provedores [Costa et al. 2009].

Um exemplo de projeto que está desenvolvendo uma nuvem federada é o *Just-in-Time Cloud (JiT Cloud)* [RNP 2010], ou *Nuvem em um Instante*. Esse nome dá a ideia de que em um pequeno intervalo de tempo podem ser obtidos milhares de recursos computacionais. Dezessete instituições de pesquisa estão envolvidas no projeto e na implementação do *middleware* para federação de recursos, de protocolos de comunicação, de sistemas de monitoramento de tráfego e tarifação, de módulos de computação autonômica e de segurança e aplicações.

A proposta do Projeto *JiT Cloud* é produzir uma nuvem federada que aloca recursos computacionais já amortizados sob demanda; ou seja, sem incorrer novas despesas sobre o custo total de investimento (TOC – *Total Cost of Ownership*) para implantação do provimento de serviços de computação em nuvem. Uma vez que os novos centros de dados podem

fazer parte da federação a qualquer momento, esse sistema se configura como uma solução escalável e de baixo custo que pode agregar rapidamente um grande número de recursos.

No contexto de uma nuvem *Just-in-Time*, o provedor de recursos é chamado de *JiT Provider*, o centro de dados é chamado de *JiT Datacenter (JiT DC)*, um recurso (máquina virtual) é chamado de *JiT Resource* e uma nuvem, de *JiT Cloud*. Quando ocorre a federação de nuvens, usa-se também o termo *JiT Clouds*. O conjunto de recursos (*pool*) disponibilizado por um provedor à federação faz parte de uma nuvem pública. No entanto, os provedores podem manter um conjunto de recursos privados em seus centros de dados. Com soluções desse tipo, é possível a criação de nuvens híbridas [Costa et al. 2010].

A Figura 2.5 ilustra a formação de uma *Jit Cloud* composta por três *pools* de recursos amortizados, que foram cedidos pelos provedores A, B e C. Em todos eles, pode-se observar que um conjunto de recursos ficou disponível apenas localmente, não fazendo parte da federação.

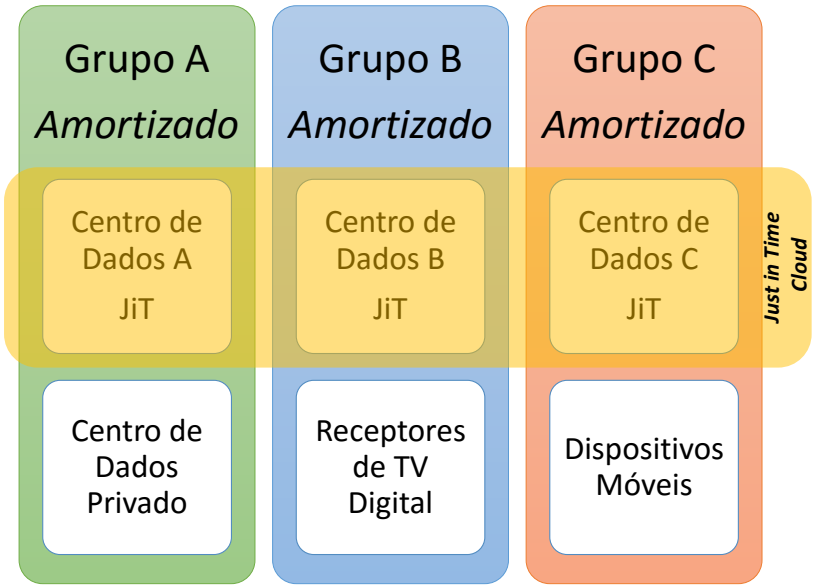


Figura 2.5: Exemplo de composição de uma *JiT Cloud*. Adaptado e traduzido de [Costa et al. 2010].

2.2 Gerência de Tráfego em Redes de Computadores

Callado et al. [Callado et al. 2009] discutem a completude e a acurácia de técnicas para identificação de tráfego da Internet. O monitoramento de sistemas de larga escala envolve diversos desafios. Um exemplo disso é como monitorar o tráfego de um sistema como a Internet. É preciso definir as políticas de monitoramento, como a opção por agregação de tráfego, fazendo análise de fluxo, ou o monitoramento da carga útil individual de cada pacote que transita na rede. Esse trabalho faz um levantamento do estado-da-arte de técnicas em coleta de tráfego usando medições ativas e passivas e técnicas para identificar tráfego, classificando as aplicações de rede.

Ao se optar por uma análise *offline* de tráfego, ou seja, armazenando o tráfego de rede para posterior análise, além do tempo de processamento de grandes volumes de dados, há o problema de armazenamento desses dados. Por exemplo, para uma rede que opera a 10 Gbps, é requerido 4,5 TB de espaço em disco para apenas 1 h de coleta de tráfego.

Uma solução alternativa é a análise *online* de tráfego; no entanto, há uma demanda alta por capacidade de processamento instantâneo, o que vem também associada a uma intensiva ocupação de memória RAM. Para o monitoramento de uma rede durante 5 s, considerando que ela opera a 8 Gpbs, é necessário que sejam armazenados 5 GB de dados em memória RAM. O processamento desses dados deve ser realizado ainda de modo que não interfira no processo de produção e transmissão dos mesmos. Portanto, mecanismos leves e não intrusivos de medição e análise devem ser utilizados para sistemas com alta vazão de dados.

2.2.1 Medições e Monitoramento de Redes

A medição das variáveis que caracterizam o desempenho da rede são divididas em duas abordagens: **medições ativas** e **medições passivas**. Na medição passiva apenas são instalados pontos de observação dos pacotes que trafegam na rede, não introduzindo ou modificando o conteúdo dos pacotes. A medição ativa do tráfego envolve o envio de pacotes de sondagem da rede (sondas – *probes*), para que se possa estimar as métricas de desempenho da rede. Exemplos de ferramentas que realizam medição ativa são o *ping* e o *traceroute*, que são usados para calcular o atraso fim-a-fim e traçar a rota entre dois sistemas finais, respectivamente. Esses comandos enviam pacotes de controle, padronizados pelo Protocolo ICMP, contendo

requisições simples no intuito de obterem prontamente uma mensagem e calcularem o atraso entre os sistemas, bem como métricas sobre o estado do enlace [Ziviani e Duarte 2005].

Chen [Chen 2001] identifica outras nomenclaturas para abordagens de medição, além de medições ativas e passivas. Elas podem ser:

I. Ligadas a um fluxo específico de pacotes ou concebidas para monitorar o comportamento da rede de forma mais genérica. No caso de estarem ligadas a um fluxo específico, as medições podem ser internas ao fluxo monitorado, onde campos adicionais no cabeçalho dos pacotes de dados podem ser usados, ou externas ao fluxo monitorado, usando sondas adicionais aos pacotes de dados;

II. Realizadas continuamente ou sob demanda;

III. Diretas ou indiretas, medindo diretamente uma determinada característica da rede ou a medindo indiretamente por meio de estimativa a partir de outra(s) característica(s);

IV. Unidirecionais ou bidirecionais;

V. Compostas de um ou múltiplos pontos de coleta de dados ou envio de sondas.

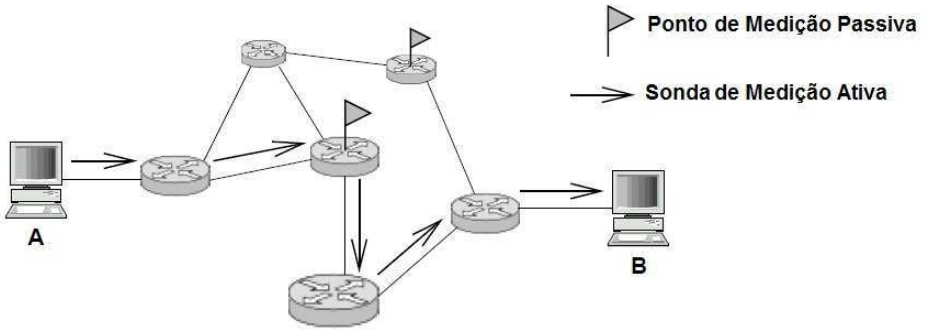


Figura 2.6: Sondas de medições ativas e pontos de medições passivas [Ziviani e Duarte 2005].

No monitoramento passivo, para se observar um volume representativo de tráfego da rede, contendo a maior parte dos pacotes, os enlaces que agregam o tráfego total da rede precisam ser monitorados. Neste caso, devem-se instalar pontos de medição na camada de

distribuição da rede ou na camada do núcleo, que componham a espinha dorsal da rede (*backbone*). Caso seja mandatório monitorar todo o tráfego da rede, os pontos de monitoramento devem ser distribuídos ao longo da topologia de rede até que se tenha a certeza de que todo o tráfego está sendo analisado.

2.2.2 Análise de Tráfego de Rede

O processo de análise de tráfego de rede consiste em três ações principais: (a) coleta; (b) monitoramento e análise; e (c) notificação. Geralmente o processo de coleta de tráfego não deve interferir no fluxo do tráfego; ou seja, coletar o tráfego não deve implicar em remover os pacotes do enlace, nem adicionar atrasos adicionais aos mesmos. Contudo, o monitoramento passivo do tráfego possui limitações para extração de algumas métricas. Por exemplo, não é possível medir o atraso entre dois sistemas finais apenas com medições passivas.

Uma aplicação específica para análise de tráfego deve ser capaz de receber os fluxos de pacotes que, agregados, devem ser alvo de monitoramento. Para tal, deve-se espelhar o tráfego que é comutado em roteadores ou *switches* em pontos estratégicos da rede e encaminhá-lo a uma porta de monitoramento, de forma transparente, em um servidor de análise de tráfego. O servidor pode monitorar e analisar o tráfego de acordo com as regras de negócio que devem ser aplicadas ao monitoramento. Toda vez que uma anomalia é identificada, ela é então notificada às partes interessadas. As anomalias podem ser armazenadas em bancos de dados para posterior auditoria ou para alimentar sistemas de contabilidade e tarifação. Essas ações estão exemplificadas através da ilustração na Figura 2.7.

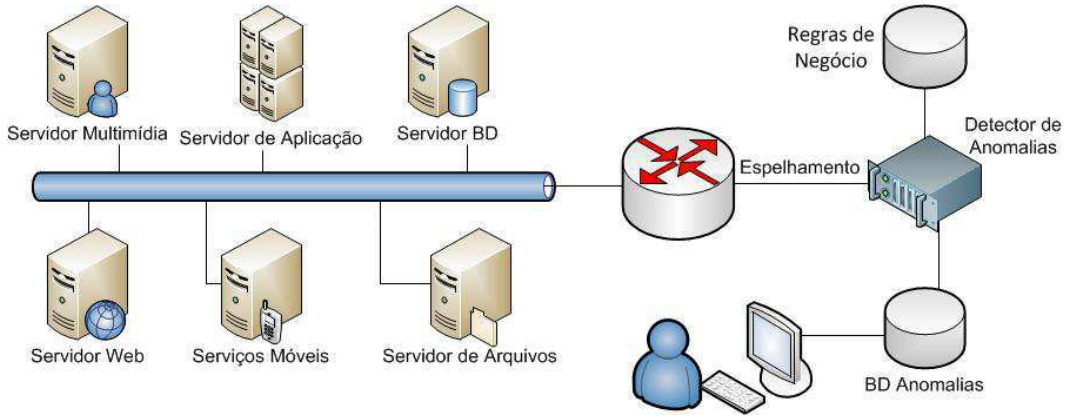


Figura 2.7: Monitoramento e análise de tráfego.

A análise de tráfego de rede pode ser realizada utilizando uma variedade de técnicas como, por exemplo, a classificação de tráfego por meio de portas conhecidas para aplicações de rede, classificação baseada em endereços de rede (IP) para servidores conhecidos, aprendizagem de máquina [Callado et al. 2010], inspeção profunda de pacotes (DPI) para casamento de padrões na carga útil dos pacotes, [Antonello et al. 2012][Lacerda et al. 2009], amostragem [Kamiyama e Mori 2006][Duffield et al. 2005], uso de processadores de rede, FPGAs (*Field Programable Gateway Array*) [Paxson et al. 2006] [Paxson et al. 2007] e processamento de eventos complexos [Dekkers 2007][Esper 2013].

O valor agregado ao tráfego de redes de computadores não é determinado simplesmente pelo volume de dados que transitam nos enlaces de rede. O grau de criticidade das aplicações e a semântica dos dados também são informações tão valorosas quanto as informações quantitativas extraídas via monitoramento de métricas. Caracterizar o tráfego das aplicações que utilizam a rede é importante para o dimensionamento e o gerenciamento de redes.

Propor modelos de custo que levam em conta não apenas métricas de volume em bytes ou a duração de fluxos de pacotes, mas que consideram também a natureza dos dados em questão, possibilita o estabelecimento de valores mais realistas para o custo dos serviços, a priorização do tráfego crítico e a penalidade por danos causados ao tráfego em maiores ou menores proporções.

2.3 Gerência de Acordo de Nível de Serviço

Os serviços de telecomunicações necessitam da rede de computadores para realizar a comunicação entre seus clientes, portanto qualquer perda de conectividade pode afetar os lucros de uma empresa. Os provedores de serviços de computação em nuvem definem níveis para o provimento dos serviços e os mesmos são acordados junto com os clientes em um documento chamado de **Contrato de Nível de Serviço**, ou (SLC – *Service Level Contract*). O SLC especifica **Acordos de Nível de Serviço**, ou ou *Service Level Agreement* (**SLA**), de conectividade e desempenho para um serviço que o cliente contrata do provedor. Um SLC contempla normalmente múltiplos SLAs, então uma violação de um SLA específico pode criar uma violação de todo o SLC [Cisco 2005].

A cada SLA geralmente são associados um *nível de serviço esperado* e um *nível de*

serviço mínimo. O nível de serviço esperado é o que é realmente contratado, portanto se o desempenho do serviço atende a esse requisito, há uma garantia de bom desempenho para o serviço; enquanto o nível de serviço mínimo desencadeia desempenho insatisfatório. Uma **violação de SLA** ocorre quando o nível do serviço esperado é violado (i.e., quando o serviço contratado fica abaixo do nível mínimo) [Cisco 2005].

Dado o caso em que um cliente deseja garantir o bom desempenho de aplicações críticas para o negócio, como voz sobre IP, redes privadas virtuais e conferências de áudio e vídeo. Supondo que o serviço pode deve ser mantido com 90% do tráfego sem perda e ela possa exceder esse limite por até três vezes em um determinado período de tempo, mas com um limite mínimo de 70% do tráfego sendo recebido ou enviado sem perda de pacotes.

A Figura 2.8 apresenta dois exemplos de violação de SLA para este caso. No primeiro exemplo, o desempenho do serviço ficou abaixo do nível esperado por quatro vezes e, no segundo exemplo, o desempenho ficou abaixo do nível mínimo [Cisco 2005], ambos constituindo violações de SLA [Cisco 2005].

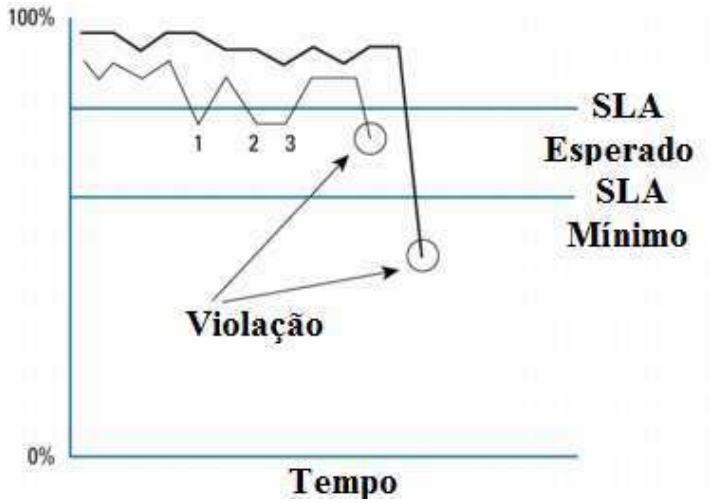


Figura 2.8: Exemplo de violação de SLA. Adaptado de [Cisco 2005].

Os níveis de desempenho adequados para a execução dos serviços contratados são chamados de **objetivos de nível de serviço**, *Service Level Objectives* (**SLO**), do SLA. A Engenharia de Nível de Serviço, *Service Level Engineering* (SLE), é a engenharia que apresenta uma metodologia para predição de indicadores de nível de serviço (SLI – *Service Level Indicators*) relevantes ao negócio e predição eficiente de SLOs [Mendes 2013].

Um dos desafios para a consolidação da computação em nuvem como uma solução corporativa confiável é cumprir com os acordos de nível de serviço firmados entre provedor e cliente. O cliente busca um serviço terceirizado para atender suas demandas de Tecnologia da Informação e Comunicação (TIC), mas necessita de garantias e monitoramento do grau de comprometimento dos níveis de serviço contratados/prestados.

O monitoramento de serviços de computação em nuvem precisa tratar os dados brutos de modo a lhes agregar valor, conferindo se as métricas críticas à execução dos serviços estão dentro dos valores-alvo requeridos e acordados. As três etapas principais do monitoramento para a obtenção de informações que possibilitem a tomada de decisão no ambiente corporativo com relação ao cumprimento dos SLAs são: (i) relatório dos dados brutos (crus); (ii) relatório de valor agregado; (iii) verificação de nível de serviço. Elas estão representadas na Figura 2.9.

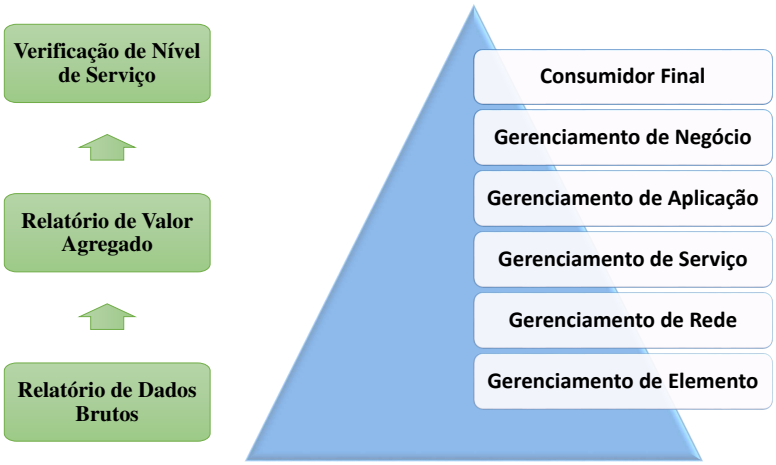


Figura 2.9: Gerenciamento de desempenho com valor agregado. Traduzido e adaptado de [Telecom 2007].

A análise de conformidade dos objetivos técnicos para os serviços possibilita o dimensionamento dos recursos sendo utilizados e do que ainda pode ser estendido para que novas aplicações sejam implantadas, bem como dar suporte a decisão para investimentos em mudança de infraestrutura de rede.

O monitoramento de SLAs pode ser feito em diferentes níveis de gerenciamento, desde o monitoramento da atividade dos elementos que compõem a rede (comutadores e roteadores),

como da gerência de rede propriamente dita (e.g., tempo de detecção de falhas, tempo de recuperação entre falhas e tempo de reconfiguração após uma falha), gerenciamento dos serviços (todos os serviços devem estar em operação, conforme a métrica de disponibilidade estabelecida), gerenciamento de negócios (como deve haver conectividade entre todos os departamentos da empresa, a rede deve ser capaz de suportar videoconferência para reuniões entre parceiros e fornecedores mesmo em horários de pico) e cliente final.

Os SLAs são comumente definidos na fase de projeto dos serviços. Mendes [Mendes 2013] propôs uma nova abordagem para a especificação de SLAs durante o período de execução dos serviços. O método de definição da qualidade de serviço é baseado em Ontologia Empresarial, *Enterprise Ontology* (EO), e em uma metodologia denominada Metodologia de Projeto e Engenharia para Organizações, ou *Design & Engineering Methodology for Organizations* (DEMO). Essa metodologia define atributos que são dinamicamente adaptados, de modo a garantir os requisitos de qualidade exigidos pelos serviços que estão executando.

Um exemplo de ferramenta que realiza monitoramento de SLAs de rede em nível de enlace é o Cisco IOS IP SLA [Cisco 2005], que possui capacidades de medição, de notificação proativa e de programação flexível.

Chhetri *et al.* [Chhetri et al. 2012] defendem que a existência de um único modelo de acordos de nível de serviço pode não ser apropriado para sistemas de computação em nuvem. Os serviços de computação em nuvem têm uma natureza de diversidade de requisitos, capacidades, restrições e preferências sobre os termos e condições de uso, além de possuírem dinamicidade com relação ao provimento e demanda de recursos, os contratos são estabelecidos de modo bilateral, ou multilateral, envolvendo no mínimo o lado do consumidor e do provedor de serviços. Diante desse cenário, a automação de acordos de nível de serviço baseada em políticas que atendam aos objetivos do negócio do cliente e na disponibilidade de recursos do provedor podem auxiliar no processo de automatização do estabelecimento de SLAs, suportando uma tomada de decisão racional.

Chhetri *et al.* [Chhetri et al. 2012] implementaram um arcabouço baseado em políticas para a automação do estabelecimento de SLAs em sistemas abertos e dinâmicos, como computação em nuvem. As principais contribuiçoes do trabalho são:

I. **Três assertivas de políticas:** assertiva de contexto, assertiva de interação de protocolo e assertiva de estratégia;

II. **Linguagem simples e flexível para especificação de políticas:** estende o arcabouço *Web Service Policy Framework* (*WS-Policy*);

III. **Implementação e validação de um protótipo usando serviços da Amazon EC2:** em que os consumidores podem delegar a compra de instâncias a um **agente de nuvem inteligente** que faz uso de políticas para a escolha do modelo de compra e de estratégias de licitações mais apropriados com base na relevância do contexto.

O processo de negociação de SLAs pode ser visto como uma busca distribuída em um espaço de potenciais acordos e a estratégia de tomada de decisão norteia a escolha do acordo mais adequado. Os quatro aspectos chaves para caracterizar o estabelecimento automático de SLAs são [Chhetri et al. 2012]:

- **Preferências dos atributos dos serviços:** cada serviço é modelado como um conjunto de atributos que podem assumir um ou mais valores;
- **Protocolos de interação:** são conjuntos de regras que regulamentam os aspectos de interações no sistema, como tipos de participantes permitidos, os diferentes estados de interação, as ações válidas nesses estados e o conteúdo das mensagens trocadas;
- **Estratégias de tomada de decisão:** as estratégias de tomada de decisão auxiliam as partes a tomarem decisões como, por exemplo, quais ofertas iniciais devem ser feitas, qual contraoferta propor e quando abandonar uma negociação, uma vez que as preferências dos atributos dos serviços variam e, geralmente, conflitam entre consumidores e provedores;
- **Contexto de interação:** representa os estados e as condições do negócio que influenciam nas interações de SLAs. Inclui informações sobre as partes, como o tamanho da empresa, taxas de crédito da empresa e histórico de interações passadas. Pode incluir ainda os objetivos do negócio com a interação, tempo e recursos disponíveis para executar as interações. O contexto de interação contribui fortemente para as estratégias de tomada de decisão.

Os protocolos para interação de SLAs são, geralmente, *documentos públicos*. Em contrapartida, as estratégias de tomada de decisão e as preferências são, normalmente, *documentos*

privados. Os protocolos especificam as regras que todos os participantes devem seguir, sendo de uso geral; enquanto as estratégias são documentos confidenciais, de uso empresarial e atendem ao protocolo de interação escolhido [Chhetri et al. 2012].

2.4 Detecção de Anomalias de Tráfego de Rede

Por definição, anomalias de tráfego de rede são os ataques, pacotes mal formados, injeção de tráfego malicioso, roubo/desvio de informações, aplicações com algum erro de configuração injetando tráfego na rede, tráfego de aplicações não permitidas e outras irregularidades que violem o padrão de tráfego esperado [Paxson et al. 2007] [Gonzalez et al. 2007][Weaver et al. 2007][Vallentin et al. 2007].

As técnicas de detecção de anomalias de tráfego em tempo real, no geral, utilizam estratégias de amostragem para diminuir o volume de pacotes a ser processado e prover informações com menor tempo de resposta possível. Pode-se perceber de antemão que as técnicas de amostragem podem mascarar anomalias [Lakhina et al. 2004][Brauckhoff et al. 2006][Ishibashi et al. 2007][Mai et al. 2006].

O **servidor analisador**, ou **detector de anomalia**, ou simplesmente **analisador**, é responsável por analisar o tráfego e notificar a ocorrência de anomalias, quando forem identificadas.

2.4.1 Técnicas Baseadas em Casamento de Padrões

Sistemas de detecção e prevenção de intrusão (IDS e IPS) são bons exemplos de aplicações que requerem processamento de pacotes *online* e que levam o sistema a tomar decisões por ações imediatas, se for detectada alguma anomalia, a fim de proteger a rede contra intrusos. Existem vários trabalhos de investigação sobre o modo de acelerar este processo, tornando-se uma combinação híbrida de software e hardware para suportar as operações [Handley et al. 2001][Katashita et al. 2007][Jacob e Brodley 2006][Paxson et al. 2006] [Vallentin et al. 2007][Paxson et al. 2007][Sommer et al. 2009][Gonzalez et al. 2007] [Weaver et al. 2007][Alharkan e Martin 2012].

A implementação de regras genéricas para protocolos específicos não é uma tarefa fácil, porque o comportamento do conjunto de regras pode variar em cada extremidade do sistema.

Por exemplo, uma extremidade do sistema pode interpretar um conjunto de pacotes de um protocolo de uma maneira diferente do que em um segundo anterior. Nem sempre, para todos os sistemas operacionais e aplicativos do receptor, a informação de uma sequência de pacotes pode levar a um comportamento inseguro ou a um ataque. Nesse sentido, o IDS pode não ser eficaz contra intrusos que sabem desses detalhes. Um exemplo de IDS de código aberto utilizado como caso de teste em vários trabalho na literatura é o SNORT [Roesch 1999][Snort 2013].

Bauer *et al.* [Bauer et al. 2001] apresentam uma técnica para detecção de anomalias que procura otimizar a varredura computacional empregada para casamento de padrões de assinaturas de anomalias de tráfego em redes de computadores. Os autores defendem que analisar toda a carga útil dos pacotes é um processo custoso e propõem uma estratégia de evolução para identificar ataques a redes de computadores combinando a análise da carga útil dos pacotes e as informações contidas nos cabeçalhos dos mesmos. Apesar de ser uma solução que apresenta aprimoramentos com relação à abordagem tradicional de detecção, ela não apresenta um bom desempenho para sistemas de análise em tempo real de tráfego, principalmente em condições de alta carga e vazão.

2.4.2 Técnicas Baseadas no Comportamento do Tráfego

Uma técnica amplamente empregada para a detecção de anomalias baseada em comportamento é a *análise de entropia*. Wang [Wang 2009] propôs um método para detecção *online* de anomalias baseado no cálculo da entropia sobre a distribuição dos valores de uma métrica do sistema, ou uma composição de métricas. O método proposto não é intrusivo, utiliza mecanismos leves de caixa-preta ou caixa-cinza, é escalável e não requer modelos previamente determinados de anomalias. O nome dado a essa técnica é EbAT (*Entropy-based Anomaly Testing*). O fundamento da mesma é estudar o comportamento das aplicações, extraindo o grau de dispersão ou concentração das distribuições de métricas. Isso é obtido a partir do estabelecimento de séries temporais de entropia, resultantes da análise de ondulação e na detecção visual de picos nas distribuições de métricas, ao invés da observância de limiares individuais para as métricas.

Smith *et al.* [Smith et al. 2010] definiram que o processo de detecção de anomalias baseadas em comportamento compreende as seguintes etapas:

1. **Transformação dos Dados:** existem várias ferramentas para monitorar a saúde de sistemas de computação em nuvem, como sensores de hardware para medir a temperatura do processador, velocidade de rotação do disco, ventilação, entre outros. Os hipervisores e o sistema operacional que provêem chamadas de sistema para rastrear as informações de uso do processador, memória, comunicação usando a rede, operações de E/S e eventos de leitura e escrita de dados. Podem ser usadas aplicações já existentes ou desenvolver alguma para que esses dados possam alimentar o arcabouço (*framework*) de detecção de anomalias;

2. **Redução de Dimensionalidade:** esta fase envolve reduzir a dimensão dos dados coletados, dada a complexidade e o volume de dados que é monitorado, transformando os dados sobre a saúde do sistema em uma métrica no espaço, mantendo apenas os atributos mais relevantes. Para reduzir os dados, é aplicado um modelo simples de Redes Bayesianas para realizar a aprendizagem supervisionada. Esse modelo representa a probabilidade da junção das variáveis coletadas. Após construído o modelo, é feita uma análise dos componentes principais (PCA – *Principal Component Analysis*) para encontrar um novo conjunto de dimensionalidades que melhor captura a variabilidade dos dados;

3. **Detecção de Discrepância:** o objetivo dessa técnica é verificar o conjunto de nós que são significativamente diferentes da maioria. Esses nós são chamados de *outliers*.

2.5 Validação de Técnicas de Detecção de Anomalias

Salfner *et al.* [Salfner et al. 2010] sistematizaram métodos de predição *online* de falhas em sistemas computacionais. Os autores destacaram as métricas mais utilizadas para avaliar a corretude e a completude de métodos de detecção de falhas que, no contexto deste trabalho, pode ser entendida por *detecção de anomalias* sem perda de generalidade. Um detector de anomalias ótimo é aquele em que o número de advertências de anomalias corresponde ao número total de anomalias presentes no sistema. Quanto menor for a incidência de falsas advertências de anomalias e maior a cobertura de verdadeiros alarmes de anomalias, melhor será o preditor.

A Tabela 2.1 apresenta os quatro estados possíveis relacionados à detecção de anomalias. Quando uma anomalia ocorre, então é feita uma suspeição de **falha**. Quando uma advertência de falha é feita e está correta, ela é chamada de **verdadeiro positivo** (TP – *true positive*). Caso haja uma advertência de anomalia e a mesma não reflita um comportamento anômalo real, então ela é contabilizada como um **falso positivo** (FP – *false positive*). Quando uma anomalia não é identificada, mas ela de fato ocorreu, então essa não-advertência é denominada **falso negativo** (FN – *false negative*). O último estado da detecção de anomalias ocorre quando o sistema está funcionando corretamente e nenhuma anomalia é notificada, então na contabilização dos eventos, esse é entendido como um **verdadeiro negativo** (TN – *true negative*).

Tabela 2.1: Tabela de contingência [Salfner et al. 2010]

Predição	**Anomalia-Verdade**	**Não-Anomalia-Verdade**	**Soma**
Anomalia (advertência)	Verdadeiro Positivo (TP) (advertência correta)	Falso Positivo (FP) (advertência falsa)	Positivos (POS)
Não-anomalia (sem advertência)	Falso Negativo (FN) (falta advertência)	Verdadeiro Negativo (TN) (falta de advertência correta)	Negativos (NEG)
Soma	**Anomalias (F)**	**Não-Anomalias (NF)**	**Total (N)**

A validação de soluções baseadas em detecção de eventos pode ser feita por meio de *métricas derivadas da tabela de contingência*. Algumas métricas possuem mais de uma notação na literatura. Na Tabela 2.2 estão apresentadas essas métricas derivadas, as quais podem ser encontradas na literatura sob mais de uma nomenclatura. Os nomes primários das métricas encontram-se na primeira coluna e os nomes alternativos estão na terceira coluna.

As métricas normalmente são analisadas em pares, como *precision/recall* e taxa de verdadeiro-positivo/taxa de falso-positivo, sensibilidade/especificidade e valor de predição positivo/valor de predição negativo.

A métrica ***precision***, ou **precisão**, representa a razão entre as anomalias corretamente detectadas e o número total de todas as anomalias detectadas, tanto verdadeiras como falsas, ou seja, caracteriza o percentual de anomalias corretamente detectadas dentre as que o sistema aceitou como anomalias. Por exemplo, caso um algoritmo de predição tenha 90% de

Tabela 2.2: Métricas obtidas a partir da tabela de contingência [Salfner et al. 2010].

Nome da Métrica	Fórmula	Nome Alternativo
Precision	$\frac{TP}{TP+FP} = \frac{TP}{POS}$	confiança, valor de predição positivo
Recall	$\frac{TP}{TP+FN} = \frac{TP}{F}$	suporte, sensibilidade, poder estatístico, taxa de verdadeiro-positivo
Taxa de falso-positivo	$\frac{FP}{FP+TN} = \frac{FP}{FN}$	*fallout*
Especificidade	$\frac{TN}{TN+FP} = \frac{TN}{NF}$	taxa de verdadeiro-negativo
Taxa de falso-negativo	$\frac{FN}{TP+FN} = \frac{FN}{F}$	1-*recall*
Valor de predição negativo	$\frac{TN}{TN+FN} = \frac{TN}{NEG}$	
Taxa de erro de falso-positivo	$\frac{FP}{FP+TP} = \frac{FP}{POS}$	1-*precision*
Acurácia	$\frac{TP+TN}{TP+TN+FP+FN} = \frac{TP+TN}{N}$	
Razão de chances	$\frac{TP \cdot TN}{FP \cdot FN}$	razão de possibilidades, *odds ratio*

precision, então 90% dos alertas estão corretos e 10% deles são falsos positivos.

A métrica ***recall***, ou **completude**, por sua vez, representa a razão de anomalias detectadas corretamente dentre o número de anomalias observadas, ou seja, dentre as anomalias que foram ou não detectadas. Por exemplo, se a métrica *recall* for 55%, isso é equivalente a dizer que 55% das anomalias foram detectadas e não houveram alertas para 45% delas.

A primeira métrica está relacionada à acurácia da detecção e a segunda é uma medida da completude da detecção. A análise dessas métricas em conjunto reflete melhor o grau de qualidade do preditor do que isoladamente.

Van Rijsbergen [Rijsbergen 1979] introduziu uma métrica chamada *métrica-F* ou *F-measure*, que representa um valor que mede a acurácia dos resultados da técnica de detecção de anomalias baseada em entropia. A métrica *F-measure* também é encontrada na literatura sob os termos **escore-**F_1 ou F_1*-score*. Na prática, essa métrica expressa a relação de custo-benefício entre as métricas *precision* e *recall*.

O cálculo da *métrica-F* é obtido pela média harmônica entre os valores obtidos para a precisão e a completude da detecção de anomalias, dado pela Equação 2.1. Quanto maior

for o valor de *F-measure*, maior será a qualidade do preditor.

$$F - measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \in [0, 1] \tag{2.1}$$

2.6 Conclusões

2.6.1 Análise de Tráfego de Serviços de Computação em Nuvem

O monitoramento de serviços de computação em nuvem demanda outro nível de abstração. Devem-se **monitorar serviços**, ao invés de pacotes individuais. Pacotes individuais que compõem um fluxo podem possuir mais de um cliente e, consequentemente, estarem associados a mais de um provedor, dependendo do tipo de serviço e do modelo de negócio de computação em nuvem que está sendo empregado. O servidor responsável por analisar o tráfego deve conhecer a lista de serviços que devem ser monitorados, seus clientes e requisitos de qualidade de serviço. À medida que os pacotes são coletados, eles vão sendo agrupados de acordo com seu(s) serviço(s) correspondente(s) e analisados de acordo com a lógica de negócio da qual fazem parte.

2.6.2 Acordos de Níveis de Serviço para Computação em Nuvem

Um mesmo conjunto de pacotes que trafegam na rede pode ser avaliado sob perspectivas de objetivos diferentes e as notificações dependerão do cliente e de suas próprias expectativas sobre os serviços. Um exemplo disso é quando um provedor A de SaaS contrata um provedor B de IaaS. Os pacotes dos clientes de SaaS do provedor A passam por um processo de análise de objetivos diferente do que é aplicado para o provedor B sob as mesmas métricas de rede. O motivo são as diferenças nos objetivos de nível de serviço (SLOs) que estão estabelecidas nos SLAs. Os clientes do provedor A contratam aplicações, que demandam diferentes níveis de serviço; portanto, entre si já existem diferenças nos SLOs dependendo da aplicação. Na visão da análise de tráfego do provedor B, o tráfego para todos os clientes do provedor A pode ser analisado apenas sob a perspectiva de um único acordo, não havendo, portanto, diferença entre o tratamento dado ao tráfego dos diversos clientes de A.

Diante desse cenário, pode-se concluir que a gestão do monitoramento de SLAs fim-

a-fim é uma tarefa difícil, especialmente no cenário de computação em nuvem. Desafios como o volume de tráfego com as redes multigigabit empregadas nos provedores de serviço, a configuração da granularidade do monitoramento, mais especificamente na definição de intervalos de medição que não comprometam a precisão do monitoramento e a qualidade geral dos serviços, as expectativas de visualização dos clientes sobre o desempenho dos serviços, a sobrecarga dos sistemas de medição e a eficiência da rede que diminui com o envio de mensagens de monitoramento.

2.6.3 Detecção de Anomalias de Tráfego de Rede de Serviços de Computação em Nuvem

Sistemas de computação em nuvem apresentam características particulares como a agregação de uma variedade de serviços, o que torna difícil a identificação das aplicações quer seja por técnicas baseadas em padrões de assinaturas de tráfego ou por meio de métodos probabilísticos de análise de comportamento, variabilidade de carga e larga escala. Em determinados pontos, especialmente quando são monitorados enlaces de provedores de computação em nuvem, há alta vazão de tráfego, o que apresenta desafios tanto para monitoramento quanto para classificação do tráfego.

O termo anomalia de tráfego de rede é genérico, pois envolve diferentes tipos de problemas. Anomalias de tráfego podem causar violações dos acordos de nível de serviço e incorrer em prejuízos financeiros para clientes ou provedores.

Diante dos diferentes tipos de anomalias e formas para detecção e tratamento, é importante delimitar o escopo relacionado à detecção e solução de anomalias de tráfego. Uma forma de simplificar o estudo de anomalias em sistemas de computação em nuvem é analisar as metas do negócio e verificar se o tráfego de rede está adequado ao que foi negociado entre as partes.

Para desenvolver modelos de contabilidade e tarifação eficientes para sistemas de computação em nuvem, é necessário tratar anomalias com relação aos SLAs. Informações de desvio do comportamento normal devem estar intrinsecamente vinculadas ao modelo de negócio do cliente. Dessa forma, o valor de uma anomalia poderá ser estimado e serem negociadas, de modo justo, penalidades ao provedor ou créditos em recursos ou em tarifas aos clientes

mediante a incidência de uma anomalia. Para tal, deve-se observar se os SLOs estão sendo garantidos e gerar alertas para possíveis desvios a esses acordos.

Capítulo 3

TADE: Arquitetura para Detecção e Gerenciamento de Anomalias de Tráfego para Serviços de Computação em Nuvem

"O segredo de aborrecer é dizer tudo."

Voltaire

Neste capítulo será apresentado o projeto de uma nova arquitetura para monitoramento, detecção e gerenciamento de anomalias de tráfego, denominada TADE (***Traffic Anomaly Detection Engine***). Essa arquitetura foi validada por meio de experimentos que envolveram a implementação prática da arquitetura e integração a uma nuvem federada em produção, a *Jit Cloud*. Os produtos de *software* desenvolvidos, que implementam e permitem integrar a arquitetura TADE a centros de dados, não são requisitos obrigatórios para um centro de dados, porém agregam importante valor quando implantados.

Neste capítulo a semântica do termo **anomalia de tráfego** será tratada no âmbito de detecção de violações de objetivos estabelecidos para métricas de rede no acordo de nível de serviço firmado entre provedor e cliente (SLA), visando verificar se o tráfego de rede satisfaz os níveis de serviço esperados que foram negociados entre as partes contratante e provedora.

O monitoramento de sistemas de computação em nuvem apresenta diferentes níveis de abstração. **Em vez de pacotes individuais, os serviços devem ser monitorados.** O mesmo conjunto de pacotes atravessando a rede pode ser visto na perspectiva de diferentes objetivos.

A notificação de uma violação de SLA vai depender do cliente, com base em seus próprios requisitos de serviço.

A **engenharia de software para computação em nuvem deve ser planejada visando a implementação de componentes que representam um serviço**. Seguindo o princípio da criação de componentes, deve-se especificar a infraestrutura de serviços e computação com foco no modelo de negócio [Goyal 2009].

A arquitetura TADE baseia-se em uma abordagem de monitoramento centrada no cliente, onde o cliente pode ter acesso a métricas de desempenho em tempo real para realizar sua gestão de contabilidade e para a tomada de decisões racionais baseadas na análise de métricas de desempenho reais. A arquitetura TADE foi implementada de modo extensível, permitindo que novas métricas sejam monitoradas.

O processo de implantação da arquitetura TADE e do serviço de detecção de anomalias de tráfego que a implementa compreendem cinco etapas:

1. Levantamento dos requisitos funcionais e não funcionais;
2. Definição da arquitetura da TADE no contexto de uma *nuvem federada*;
3. Especificação técnica de como a arquitetura TADE deve ser implementada;
4. Processo de desenvolvimento iterativo e incremental dos produtos de *software* que implementam a arquitetura TADE;
5. Empacotamento e disponibilização dos produtos de *software* da TADE.

3.1 Funcionalidades

A arquitetura de monitoramento TADE compreende cinco funcionalidades básicas que são definidas e representadas pelos seguintes módulos:

1. ***Coleta de tráfego:*** o tráfego é capturado via espelhamento de portas em um *switch* e desviado para a máquina que hospeda o software *detector de anomalias de tráfego*;
2. ***Pré-processamento de pacotes***: consiste em monitorar apenas os pacotes que requerem monitoramento. O pré-processamento dos pacotes envolve duas ações:

(a) ***Filtragem do tráfego a ser monitorado:*** por exemplo, se houverem três máquinas virtuais instanciadas com acordos de nível de serviço firmados para métricas de rede, apenas o tráfego dessas máquinas virtuais deve ser monitorado. Essa ação é dividida em duas etapas:

i. ***Início do monitoramento do tráfego de rede de uma instância de VM:*** quando uma máquina virtual que possui requisitos de SLA de rede é instanciada, é iniciado o monitoramento do tráfego dessa instância;

ii. ***Término do monitoramento do tráfego de rede de uma instância de VM:*** quando a instância de máquina virtual que está sendo monitorada é encerrada, é cancelado o monitoramento do tráfego dessa instância.

(b) ***Cálculo de métricas de rede:*** para cada instância que deve ser monitorada, deve-se avaliar o tráfego de rede que ela produz ou consome e mensurar as métricas de rede que são requisitadas no SLA associado à mesma. Por exemplo, caso o SLA vinculado a uma instância possua restrições quanto à largura de banda e atraso, apenas essas métricas serão calculadas para todo o tráfego de rede dessa instância.

3. ***Gerenciamento de acordos de nível de serviço:***o módulo de gerenciamento de SLAs é responsável por armazenar os objetivos do acordo de nível de serviço (SLO - *Service Level Objectives*), que são as métricas de desempenho (de rede) que devem ser cumpridas para cada contrato (SLA) estabelecido entre o cliente e o provedor de serviços de computação em nuvem. Esses objetivos estão vinculados à máquina virtual que foi instanciada e está sendo monitorada.

4. ***Análise de conformidade de tráfego:*** este módulo é responsável por verificar se as métricas de rede que foram mensuradas na etapa *Cálculo de Métricas de Rede* cumprem os objetivos (SLOs) individuais estabelecidos para cada uma delas. Uma anomalia de tráfego ocorre quando há um desvio nos valores das métricas do tráfego de rede mensurado com seus respectivos SLOs. Quando isso ocorre, então esse módulo publica as anomalias identificadas no **canal *Publish-Subscribe* de notificação**.

5. ***Notificação de anomalias:*** esta aplicação possui duas funções básicas na arquitetura TADE:

(a) ***Contabilidade gerencial:*** além de um *front-end* para a visualização de anomalias de tráfego em tempo real, essa arquitetura serve como ferramenta para a gerência da contabilidade de serviços de computação em nuvem. Como o canal de comunicação P-S é uma solução que proporciona flexibilidade, ele pode ser utilizado por diversos componentes existentes tanto no lado do cliente, quanto do provedor, auxiliando em processos autonômicos e fornecendo subsídios para a tomada de decisão gerencial.

(b) ***Alimentação do banco de dados de anomalias:*** a aplicação se registra no canal *Publish-Subscribe* de notificação, recebe as mensagens que descrevem as anomalias em tempo real, separa seus respectivos campos de modo a compor um registro da *tabela de anomalias* e faz a inserção do registro no *banco de dados de anomalias de tráfego (BD)*. O BD serve de histórico para as anomalias de tráfego de rede identificadas durante a execução dos serviços de computação em nuvem. Como o processo de detecção de anomalias de tráfego é completamente *online*, então o registro dessas informações pode ser útil para futuras *tomadas de decisão* tanto por parte do cliente, quanto do provedor de serviços;

Um resumo de como essas funcionalidades são agregadas e de como seus módulos cooperam para antingir os objetivos propostos encontra-se na Figura 3.1.

3.2 Produtos de *Software*

Para comprender como a arquitetura TADE foi implentada, é importante separar conceitos no tocante à função de cada produto de *software*. Para implementar a arquitetura TADE são necessários três produtos de *software*. Cada produto desenvolvido interage de acordo com suas funções na arquitetura. Esta seção descreve os detalhes de implementação de cada uma das etapas do mecanismo de detecção de anomalias de tráfego, que vão desde a submissão de requisições para a obtenção de máquinas virtuais por parte dos clientes até a visualização de anomalias no tráfego de rede nas máquinas virtuais. Esses produtos são:

1. Detector de anomalia (*Anomaly Detector*);

2. Agente de detecção de anomalias (*Anomaly Detector Agent*);

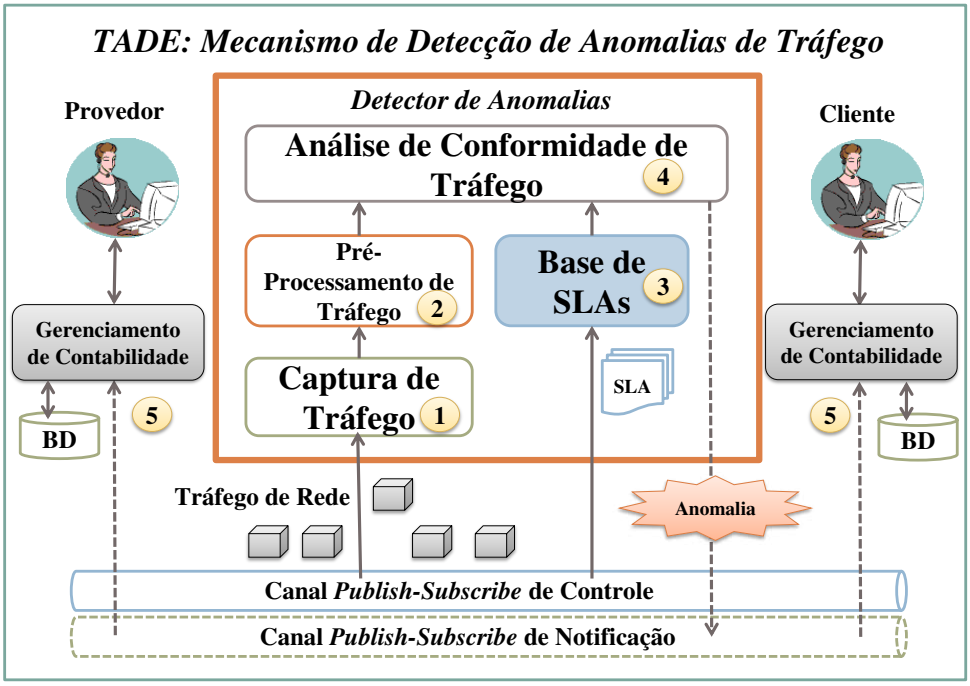


Figura 3.1: Arquitetura TADE.

3. Aplicação *Web* para notificação de anomalias.

Antes de ser feita uma descrição detalhada dos produtos de *software*, serão apresentadas as interações que ocorrem entre os produtos com base no processo de provimento de serviços de computação em nuvem e no seu monitoramento, que é realizado em paralelo.

3.2.1 Interações entre os Produtos de *Software*

As funcionalidades da arquitetura *TADE* envolvem todo o processo de tratamento do tráfego de rede, que vai desde a captura dos pacotes da rede à notificação das anomalias identificadas em tempo real. Na Figura 3.2 é ilustrado o processo de interação entre os três produtos de *software* para proporcionar uma visão geral.

Quanto à nomenclatura, nesta seção serão utilizados os termos *DC Controller* (também pode ser chamado de *DC Middleware*), ou *controlador de centro de dados*, para representar a interface de *software* entre o provedor de serviços de computação em nuvem (*DC Provider*) e os módulos da arquitetura TADE. Esse é o único componente do lado provedor que requer programação para fazer a tradução das requisições dos clientes para um padrão de dados que

seja entendido pelos componentes da arquitetura TADE.

O processo se dá da seguinte forma. O cliente requisita instâncias de máquinas virtuais ao *DC Controller* (desenvolvido em linguagem Java). A função do *DC Controller* é realizar o tratamento das requisições dos clientes e a alocação das máquinas virtuais solicitadas. Uma nuvem federada pode ser composta por diversos centros de dados (*DCs*) geograficamente distribuídos. Nem todos eles possuem a infraestrutura de *hardware* e *software* necessária para realizar a análise do tráfego de rede das instâncias que são executadas em seu contexto local.

O ponto de coleta do tráfego e processamento do tráfego foi implantado no *DC Provider* onde os serviços (VMs) estão sendo executados. Como faz parte da mesma rede local das VMs, as métricas de rede, como o atraso para outros DCs, foram generalizadas como sendo as mesmas. Este trabalho adequa-se apenas aos *DCs* que possuam a infraestrutura necessária para realizar o processo de detecção de anomalias.

Quando o *DC Controller* recebe requisições dos clientes, ele deve observar se a execução das instâncias requer o monitoramento do tráfego de rede. Nem todas as requisições pressupõem esse tipo de monitoramento.

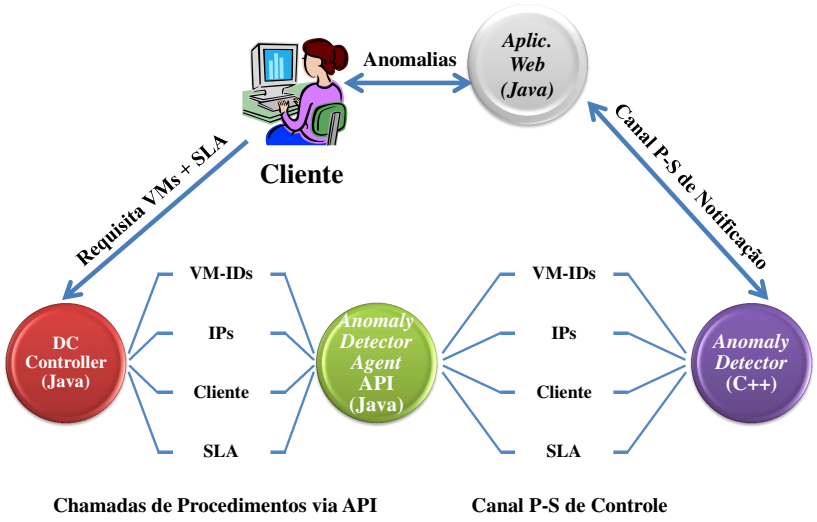


Figura 3.2: Fluxo de dados entre os três produtos de software que implementam a arquitetura TADE.

3.2.2 *Anomaly Detector*

Quando houver a demanda pela análise do tráfego, o *DC Controller* deve informar ao *Anomaly Detector* que esse monitoramento precisa ser realizado, para quais instâncias, qual o cliente e qual o SLA acordado para as métricas de rede.

Por questões de desempenho e funcionalidades de baixo nível, o *Anomaly Detector* é desenvolvido na linguagem C++. Ele não é obrigatoriamente executado na mesma máquina em que o *DC Controller* está sendo executado, principalmente porque o processamento de pacotes faz uso intensivo de CPU e memória, podendo demandar *hardware* dedicado para não haver perdas de pacotes no momento do processamento.

3.2.3 *Anomaly Detector Agent*

O produto de *software* chamado *Anomaly Detector Agent* é desenvolvido em Java e iniciado na mesma máquina em que é executado o *DC Middleware*. O *Anomaly Detector Agent* possui uma API pública, a *AnomalyDetectorAgentAPI*, cujos métodos são chamados diretamente pelo *DC Controller* no momento em que um cliente solicita a instanciação de máquinas virtuais ou o seu término.

A comunicação entre o *Anomaly Detector Agent* e o *Anomaly Detector* para informar os dados necessários para iniciar ou encerrar o monitoramento das instâncias se dá via um canal de comunicação do tipo *Publish-Subscribe (P-S)*. Esse canal será usado exclusivamente para a publicação de informações de controle, então será chamado de **Canal P-S de Controle**. Quando o *Anomaly Detector* identifica anomalias no desempenho do tráfego de rede das instâncias, ele as publica em um canal nomeado de **Canal P-S de Notificação**, pois é utilizado para publicar as notificações de anomalias.

3.2.4 Aplicação *Web*

Os clientes interessados em visualizar as anomalias de tráfego em tempo real podem se inscrever diretamente no canal P-S de notificação ou utilizar a *Aplicação Web* para visualizar essas anomalias. Essa aplicação foi desenvolvida usando a linguagem de programação Java.

Apesar de ter sido desenvolvida uma aplicação Web para a publicação de anomalias, outros clientes ou agentes, desde que possuam as credenciais necessárias, podem acessar,

sob demanda, o **canal de comunicação *Publish-Subscribe* (P-S)** e serem notificados da incidência de anomalias em tempo real.

3.3 Implantação dos Produtos de *Software*

A arquitetura TADE proposta foi integrada em uma nuvem federada em produção. O *Just-in-Time Clouds (JiT Clouds)* é um exemplo de um *middleware* de código aberto para federação de nuvens para o provimento de *Infrastructure-as-a-Service* (IaaS) [Fraga et al. 2013]. O *JiT Clouds* visa proporcionar uma nuvem federada que aloca recursos já amortizados; isto é, sem incorrer em novas despesas com custo total de propriedade (TCO). Novos centros de dados podem juntar-se à federação, em qualquer tempo. É um sistema escalável e de baixo custo, que pode lidar rapidamente com um grande número de recursos na nuvem. Alguns centros de dados em laboratórios de pesquisa e Universidades em vários estados do Brasil participam da federação usando o *JiT Clouds* [RNP 2010].

O termo *JiT Clouds* também é usado para representar uma federação de nuvens. O conjunto de recursos computacionais que um provedor disponibiliza para a federação é parte de uma nuvem pública. De outro lado, provedores podem ainda manter um conjunto privado dos recursos em seus centros de dados, viabilizando, portanto, a implementação de nuvens híbridas usando o *middleware* [Costa et al. 2010].

Como os produtos de *software* que implementam a arquitetura TADE foram implantados em uma nuvem federada, para padronizar a nomenclatura, será mantida a compatibilidade de terminologia utilizada na documentação do *middleware*. O provedor de recursos é chamado de *JiT Provider*. O centro de dados é chamado de *JiT Data Center* (ou *JiT DC*). Uma fonte de recursos computacionais, que neste contexto representa uma máquina virtual, é chamada de *JiT Resource*. O componente *JiT Cloud* desempenha o papel de gerir a infraestrutura, adicionando e removendo *JiT DCs*, de autenticação de usuário e de gerir o ciclo de vida das máquinas virtuais [Fraga et al. 2013].

Uma nuvem federada pode ser composta por centros de dados (DCs) distribuídos geograficamente e nem todos os DCs possuem a mesma infraestrutura disponível de *hardware* e *software* necessária para executar o processo de análise e monitoramento de tráfego de rede em instâncias de máquinas virtuais no seu contexto local.

A arquitetura TADE foi implantada em um *JiT DC* no laboratório do Grupo de Pesquisa em Redes Convergentes (GPRC) do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB). O experimento consistiu em instanciar os produtos de *software* da arquitetura TADE para monitorar os serviços de nuvem prestados pelo *JiT Provider* participante da federação que foi instalado no laboratório.

Os requisitos não-funcionais e funcionais ds três produtos de *software* que implementam a arquitetura TADE estão descritos no Apêndice C. O projeto dos casos de uso com diagramas de sequência e suas especificações é apresentado no Apêndice D.

O *software* do controlador executa em uma máquina separada das máquinas em que os produtos de *software* da TADE são executados. O cliente solicita instâncias VM para o controlador do DC, que pode ser desenvolvido usando qualquer linguagem de programação. Neste estudo, o *DC Controller* foi escrito em Java. O Controlador DC recebe solicitações do cliente e aloca instâncias de máquina virtual.

Sempre que o controlador recebe pedidos, deve verificar se a execução dessas requisições requer monitoramento de SLA. Clientes ou agentes específicos do provedor podem se inscrever para serem notificados sobre a ocorrência de anomalias em tempo real. Note que existem dois canais P-S diferentes. Um tem a função de fornecer dados de controle, tais como credenciais de clientes e cláusulas dos SLAs e, o outro, tem a função de realizar a notificação de violações de SLA em si.

Os SLAs são descritos e publicados usando o a linguagem de marcação XML (*eXtensible Markup Language*). Com base no SLO, o analisador de tráfego continuamente monitora os fluxos de dados originados a partir de, ou dirigido para, a VM solicitada, para verificar se existe uma violação para o SLA (ou seja, se alguma métrica monitorada ultrapassa o limite definido no SLO). Uma vez que uma violação de SLA ocorre, ele aciona uma notificação de anomalia de tráfego, e uma mensagem é enviada para todos os componentes da nuvem interessados por meio do canal P-S de notificação. Após isso, ações autonômicas podem ser disparadas, como a migração de VMs e o gerenciamento de contabilidade e de preços.

Os canais P-S utilizam o protocolo AMQP (**Advanced Message Queuing Protocol**) [AMQP 2015], por meio do arcabouço **RabbitMQ** [RabbitMQ 2015], que implementa o protocolo em quatro linguagens de programação. Cada cliente AMQP pode funcionar como um publicador ou como assinante, desde que exista ao menos um servidor que atue como

intermediador das mensagens (*broker*).

O gerenciador de máquinas virtuais de cada DC é executado sobre o arcabouço **Eucalyptus framework** [Eucalyptus 2014]. As VMs foram instanciadas e geraram tráfego sintético usando uma ferramenta chamada Ostinato [Ostinato 2015]. Para capturar o tráfego, o *software* analisador emprega um *driver* de rede chamado **PF_RING** [PF_RING 2015], para alto desempenho de captura de pacotes de rede. Utilizou-se a API do PF_RING para lidar com os dados dos pacotes dentro do analisador, bem como na instrumentação do código para coletar as métricas.

A arquitetura TADE implementada e disponível realiza o cálculo de quatro metas técnicas de SLAs (também chamadas de métricas de rede):

1. **Latência (ou atraso)**: é calculada por meio de medição ativa usando pacotes *ICMP echo request* e *ICMP echo repply*;

2. **Variação do atraso (*jitter*):** é calculada por meio da diferença entre duas medições de atraso consecutivas;

3. **Largura de banda:** é estimada por meio de medições passivas, após a medição do volume (em *bytes*) presente em cada pacote pertencente a um fluxo monitorado;

4. **Disponibilidade:** é estimada por meio de medição ativa (a mesmas utilizada para a medição do atraso). Verifica-se se o percentual mínimo exigido para a métrica de disponibilidade está sendo cumprido.

3.3.1 Verificação

Foram realizados testes de integração dos produtos de *software* que implementam a arquitetura TADE ao *middleware JiT Clouds*. Em relação à estimativa de métricas do SLA, foram realizados testes de aceitação para assegurar que as métricas foram calculadas corretamente.

3.3.2 Ambiente de Testes

O ambiente de testes foi criado com base em um modelo mestre-escravo, onde existem as máquinas para gerar tráfego de rede (também conhecidas como *SLAVES*, escravas) e uma máquina para monitorar as métricas de SLA de rede (também chamado de *MASTER*, mestre,

ou analisador). Maiores detalhes sobre os equipamentos de *hardware* e *software* do ambiente de testes estão descritos no Apêndice E.

3.4 Validação

O projeto da arquitetura TADE baseou-se nas diretrizes conceituais propostas por Zhang e Zhou [Zhang e Zhou 2009] que foram introduzidas na Seção 2.1.1, as quais são:

1. Desenvolver uma forma reutilizável para a criação de plataforma de provisionamento escalável e configurável para computação em nuvem;

2. Propor um conjunto de serviços comuns e compartilhados para a construção de plataformas de computação em nuvem, quer seja para prestação de serviços de negócios (a um usuário final) ou outros serviços de nuvem para consumidores empresariais empregando uma abordagem unificada;

3. Maximizar o valor potencial do negócio na computação em nuvem com base em uma infraestrutura de TI extensível e no sistema de gestão. A ideia principal é monetarizar o valor agregado dos serviços da nuvem de negócios, combinando as potencialidades da SOA e da computação em nuvem.

Realizando o casamento entre os objetivos para o desenvolvimento de uma arquitetura adequada propostos por Zhang e Zhou [Zhang e Zhou 2009] e o projeto da arquitetura TADE, verifica-se que há alinhamento entre eles. A arquitetura TADE atende aos objetivos propostos para resolver o problema de tornar disponível informações sobre métricas de rede para apoiar a gestão de serviços de computação em nuvem às partes interessadas.

Concluiu-se o processo de instalar e implantar os produtos *software*, configurar o *hardware* para capturar os pacotes e analisar as métricas de tráfego da rede de acordo com as restrições de SLA. Após isso, realizou-se um estudo experimental para avaliar o desempenho do processamento de métricas de SLA de rede dos produtos de *software* desenvolvidos e implantados.

Realizou-se um experimento controlado variando a taxa de geração de pacotes das VMs de 2 Mbps a 1 Gbps, conforme detalhes presentes na Tabela 3.1. Os pacotes possuem tamanho fixo de 1.518 bytes. O tráfego de rede produzido pelas VMs foi espelhado no ambiente

de testes e encaminhado para análise. Não houve perda de pacotes em sua transmissão pela rede.

Tabela 3.1: Tratamentos do experimento para avaliar desempenho da implementação da arquitetura TADE.

#Tratamento	Taxa de Geração de Pacotes
1	2 Mbps
2	4 Mbps
3	8 Mbps
4	10 Mbps
5	100 Mbps
6	1000 Mbps

Para análise do desempenho da implementação da arquitetura TADE, foram medidos:

I. Número de pacotes recebidos na aplicação analisadora (*anomaly detector*);

II. Utilização de CPU;

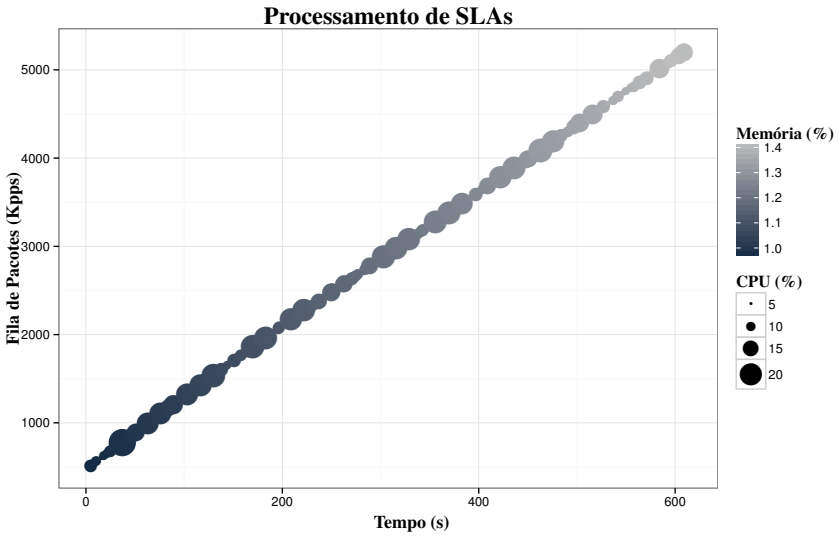
III. Utilização de memória;

IV. Perda de pacotes no sistema operacional (*kernel*).

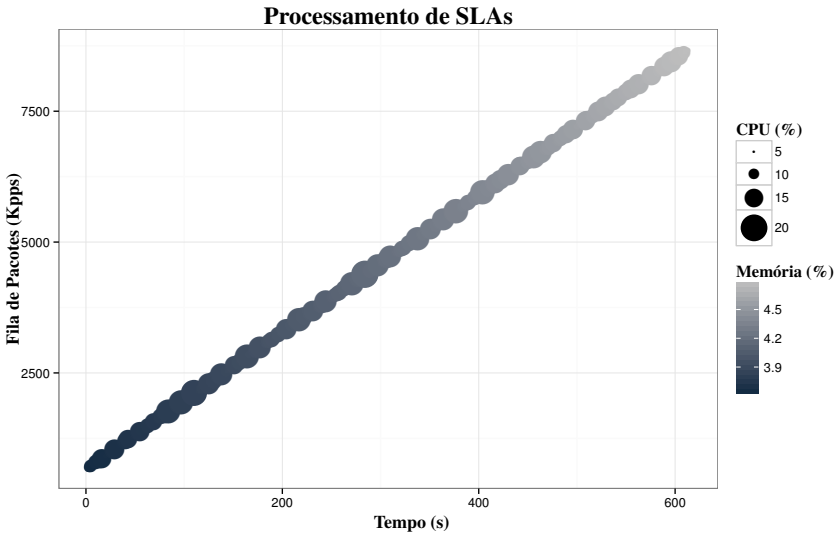
O cálculo das metas técnicas dos SLAs foi realizado para todos os pacotes recebidos (sem amostragem), o que gera uma sobrecarga considerável. Os resultados obtidos para as três primeiras métricas são apresentados na Figura 3.3. A Figura 3.3a apresenta os resultados para a taxa de chegada de pacotes pela rede de 100 Mbps, e a Figura 3.3b os resultados de quando a máquina mestre recebe a carga mais alta. Os outros resultados foram omitidos por terem melhor desempenho; portanto, serão estudados os piores casos.

Para a taxa mais alta de tráfego de entrada, a utilização de CPU, representada pelo tamanho de cada ponto no gráfico, variou de 5 a 20%; ao passo que a utilização de memória, definida por uma escala de cor cinza, permaneceu abaixo de 5% (Figura 3.3b). Os resultados mostram que o tamanho da fila de pacotes na aplicação analisadora cresce linearmente com

ponto de saturação constante; consequentemente, haverá eventualmente perda de pacotes na coleta pelo detector de anomalias (descarte realizado pelo *kernel*).



(a) 100 Mbps



(b) 1 Gbps

Figura 3.3: Desempenho do processamento de pacotes, variando a taxa de tráfego de entrada.

Em relação à perda de pacotes causada por um ponto de saturação presente na aplicação analisadora, apenas a partir da carga de 1 Gbps foram observadas perdas de pacotes pelo *kernel*. No entanto, esse percentual manteve-se abaixo de $0,01\%$, mostrado por triângulos na Figura 3.4, que configura um impacto desprezível no desempenho.

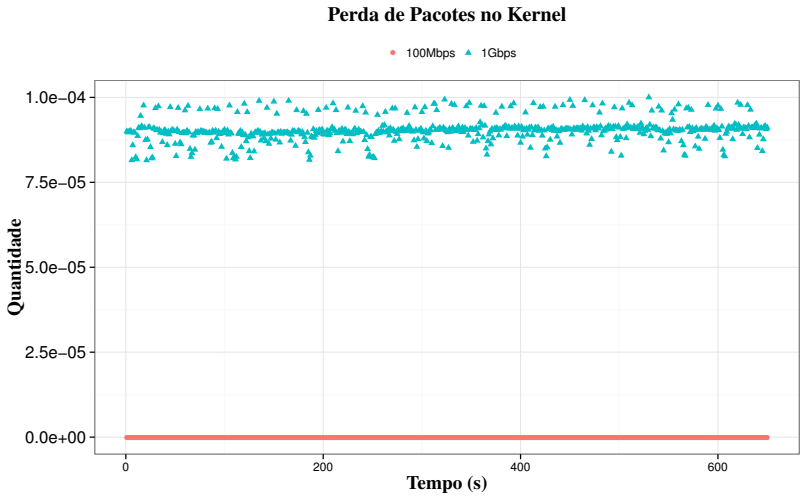


Figura 3.4: Percentual de perda de pacotes, com entrada de tráfego a 100 Mbps e 1 Gbps.

Neste estudo foi analisada apenas uma instância do *software* analisador. Como há independência para a análise dos fluxos de rede, há possibilidade de dividir o tráfego e realizar balanceamento de carga para análise distribuída do tráfego. O emprego de amostragem no processamento de métricas é uma prática realizada por quase todas as ferramentas de análise de tráfego. Caso seja empregada amostragem de pacotes, o desempenho da ferramenta implementada pode alcançar desempenho superior ao apresentado. Na prática, é aceitável realizar a amostragem, contanto que não omita violações de SLAs.

Os resultados demonstraram que os produtos de *software* que implementam arquitetura TADE constituem uma ferramenta promissora, que pode ser escalada para analisar cargas de trabalho mais pesadas. Essa ferramenta está em um estágio inicial, podendo ser aprimorada inclusive cooperativamente, pois foi disponibilizada como uma ferramenta de código aberto.

3.5 Conclusões

Os clientes precisam confiar na fiabilidade da nuvem. Tornar informações relevantes ao negócio disponível para os clientes pode ajudar a agregar valor a soluções de computação em nuvem, promovendo orientação para a negociação de SLAs e contribuindo para consolidar os serviços de nuvem em setores econômicos criticamente dependentes de informação.

Há necessidade de tornar as métricas de desempenho de rede disponível para os clientes, porque tal informação é um diferencial para o negócio, bem como um importante requisito

não-funcional que é por vezes ignorado por vários prestadores de serviços em nuvem. O conhecimento de métricas de desempenho em tempo real sobre a execução dos serviços pode contribuir para promover a venda de novos serviços e a consolidação de serviços em nuvem.

Em contrapartida, os provedores podem estar cientes de demandas críticas sobre os investimentos em monitoramento e gerenciamento de ativos para evitar perdas financeiras.

O trabalho descrito neste capítulo contribuiu para o estado-da-prática do monitoramento de rede de sistemas de computação em nuvem por meio de uma nova arquitetura para a detecção de anomalias que foi validada e integrada a uma infraestrutura de nuvem federada. Um ponto crítico para o desenvolvimento de modelos de contabilidade eficazes é como lidar com tráfego anômalo, especialmente no que tange o monitoramento de SLAs.

Outra contribuição deste trabalho é a disponibilização do código-fonte dos produtos de *software* desenvolvidos que podem apoiar o gerenciamento de contabilidade tanto de provedores quanto de clientes. Os produtos de *software* da arquitetura TADE, bem como o *middleware JiT Clouds* podem ser obtidos a partir do seguinte endereço *Web*: *http://jitclouds.lsd.ufcg.edu.br*.

Os resultados demonstram que o processamento de pacotes não ultrapassou 20% do percentual de CPU disponível e 5% da capacidade de memória para processar métricas de SLA para uma vazão de 1 Gbps. Como trabalho futuro, pretende-se propor soluções para acelerar o cômputo de métricas de rede sem comprometer a acurácia da estimativa.

O trabalho apresentado neste capítulo é fruto de um esforço colaborativo, que envolveu alunos de graduação do IFPB e da UFCG como desenvolvedores e em experimentos, a doutoranda atuou na análise e projeto dos produtos de *software*, no desenvolvimento de testes de aceitação e integração, automação de experimentos, análise de resultados, como gerente de projetos e na escrita de relatórios técnicos. Os orientadores desta tese trabalharam na discussão das ideias e em testes de aceitação. Houve ainda a colaboração de um professor do IFPB Campus Maceió que atuou na análise de resultados e melhoria do trabalho.

Capítulo 4

Mecanismo para Detecção de Anomalias de Tráfego baseado em Entropia

"Muito estudo não ensina compreensão."

Heráclito

Detecção eficiente de anomalia de tráfego de rede é um problema amplamente estudado para combater ataques e uso indesejado de infraestruturas de comunicação. As técnicas existentes para detectar, prevenir ou controlar esses ataques são geralmente baseadas em limiares conhecidos, na construção de perfis de padrões de tráfego normais, ou em padrões de assinatura de comportamento anômalo (como ocorre em antivírus). Por outro lado, existem técnicas dinâmicas que com o objetivo de prever o grau de desorganização do sistema; ou seja, a entropia do sistema.

Embora soluções baseadas em entropia não suponham o conhecimento prévio do comportamento normal do sistema, os resultados apontam para a necessidade de ajuste mais preciso de parâmetros levando em consideração a natureza dos dados, a frequência de eventos e a variação de valores de métricas.

Neste capítulo será introduzido um mecanismo para detecção de anomalias de rede com base em uma técnica para avaliar o grau de desorganização de métricas chamada EbAT (*Entropy-based Anomaly Testing*) proposto por Wang *et al.* [Wang et al. 2009], que será implementado e avaliado. As conclusões obtidas apontam dificuldades para configurar o

mecanismo e identificar anomalias.

Para melhorar os resultados, foi avaliado um novo método para detecção de anomalias baseado em entropia unido à técnica de aprendizagem de máquina não supervisionada, o EMATADE [Oliveira et al. 2014b], que será apresentado e avaliado no Capítulo 5.

4.1 Implementação da Técnica de Deteção de Anomalias Baseada em Entropia

O processo de detecção de anomalias do EbAT proposto por Wang *et al.* [Wang et al. 2010][Wang et al. 2009] é dividido em três etapas: (a) coleta de métricas; (b) construção de séries temporais de entropia e (c) o tratamento de séries temporais de entropia. A técnica EbAT pode ser usada para monitorar quaisquer tipos de métricas, como tráfego de rede, ou níveis de desempenho de aplicações, como CPU e memória, utilização de disco rígido, entre outras.

A análise de uma série temporal de entropia consiste na aplicação de um ou mais dos métodos para descobrir padrões anormais no sistema. Estes métodos podem ser uma combinação de detecção de picos, processamento de sinais, análise de subespaço ou outras técnicas pertinentes.

Neste trabalho foi desenvolvido um detector de anomalias de tráfego de rede baseado na técnica EbAT para fins de avaliação prática. O sistema de detecção *online* de rede baseado em entropia realiza a amostragem de métricas e, em seguida, calcula o grau de dispersão dos valores amostrados (entropia). A metodologia para realizar a detecção de anomalias de tráfego é resumida na Figura 4.1. O funcionamento do detector implementado é explicado em pormenores nas seções subsequentes.

4.2 Construção de Séries Temporais de Entropia

Vale ressaltar que quaisquer tipos de métricas podem ser monitorados e analisados em conjunto. Antes da construção da série temporal de entropia, os dados são pré-processados. Esse processo consiste em normalizar e categorizar (*binning*) os dados antes da entropia ser calculada.

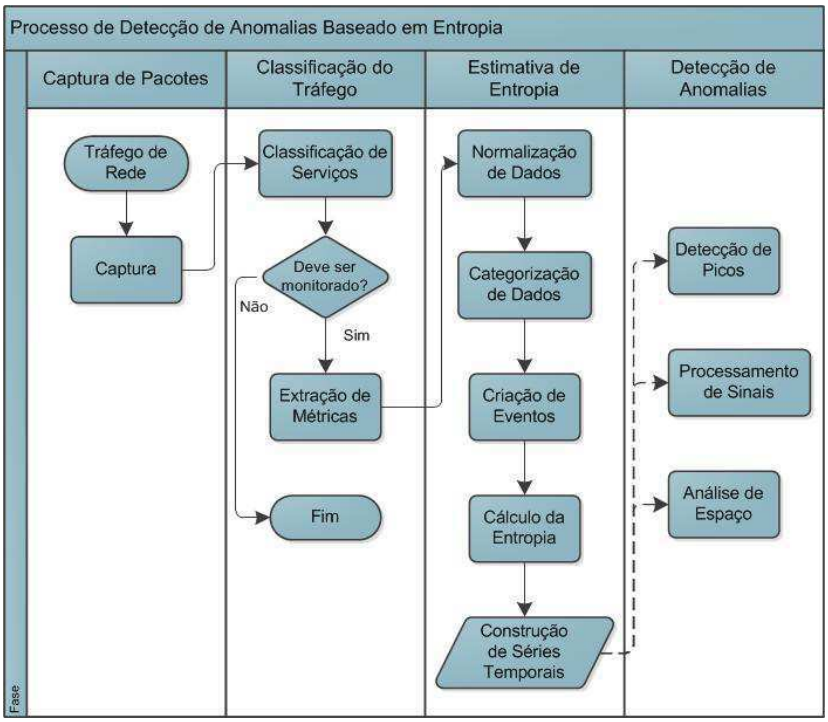


Figura 4.1: Fluxo de processamento e dados da detecção de anomalias de rede baseada em entropia.

Os valores amostrados são colocados em uma janela deslizante de tamanho *n*, onde *n* significa o número de amostras da métrica alvo da análise em um determinado momento de tempo. Essa janela desliza à medida que uma nova métrica é obtida. O primeiro valor inserido na janela sai para que o último possa fazer parte da janela, que possui um tamanho fixo. As fases para a construção da série temporal da entropia de uma dada métrica são:

I. **Fase de Normalização dos Dados:** consiste em dividir os valores amostrados contidos na janela pela média de todos os valores do mesmo tipo que pertencem à janela corrente.

$$s'_{i,j} = \frac{s_{i,j}}{\frac{1}{n}\sum_{i=1}^{n} s_{i,j}} \tag{4.1}$$

Onde:

- $i = \{1, .., n\}$ representa o índice que o valor amostrado ocupa no vetor da janela deslizante corrente;

- $j = \{1, .., k\}$ representa o índice da métrica, sendo *k* é o número de métricas que estão sendo monitoradas;
- $s_{i,j}$ é o *i*-ésimo valor amostrado da janela deslizante para a *j*-ésima métrica;
- $s'_{i,j}$ é o *i*-ésimo valor de amostra normalizado da janela da *j*-ésima métrica.

II. **Fase de Categorização (*Binning*):** nessa fase todos os valores normalizados da janela corrente são inseridos em uma classe (*bin*) de dados, que representa um intervalo para os dados. A Equação 4.2 permite que seja calculado o índice que representa a classe do valor amostrado.

$$b_{i,j} = \begin{cases} m, & s'_{i,j} > r \\ \left\lfloor \frac{s'_{i,j}}{r/m} \right\rfloor, & s'_{i,j} \leq r \end{cases} \tag{4.2}$$

Onde:

- $b_{i,j}$ é o índice da classe correspondente para $s'_{i,j}$;
- *r* é o intervalo de dados normalizados. É definido um intervalo $[0, r]$ para representar os dados mais significativos;
- *m* é o maior índice que pode ser atribuído a uma classe. Uma vez que o primeiro índice é 0, então há *m+1* classes.

O índice da classe $b_{i,j}$ do valor amostrado normalizado $s'_{i,j}$ será o índice da última classe, m, se o valor amostrado for maior do que a faixa esperada, r. Na classe m são postos os maiores valores amostrados da janela em análise. O resto dos valores são colocados dentro das classes no intervalo $[0, r]$. Para escolher a classe adequada para os valores remanescentes, divide-se o valor amostrado normalizado pela razão r/m, o que nos dá uma ideia de uma colocação justa dos valores em m intervalos de igual tamanho. Quando a divisão não é inteira, o índice da classe é considerado como o valor do piso dessa divisão, ou seja, é feito um arredondamento para baixo, como mostrado na Equação 4.2.

Seja C o conjunto de métricas sendo monitoradas, $C = \{c_1, \ldots, c_k\}$, $k = 1..|C|$. Pode-se monitorar qualquer número de métricas. Para cada métrica a ser monitorada,

existe um valor amostrado e um compartimento (classe) correspondente. Exemplos de métricas são utilização de CPU e memória de uma determinada aplicação e em nível de rede, existem métricas como atraso, largura de banda, e *jitter*, por exemplo.

III. **Fase de Criação de Evento:** após a normalização e categorização dos dados, o próximo passo é representar as métricas como eventos. Estes eventos são chamados de *eventos de medição*, ou *m-eventos*. Um evento e_i é um vetor que contém as classes de todas as métricas j analisadas. Um *evento-m* é definido como:

$$e_i = \langle b_{i,1}, b_{i,2}, \ldots, b_{i,k} \rangle \tag{4.3}$$

IV. **Fase de Cálculo da Entropia e de Agregação:** por definição, a entropia de uma variável aleatória discreta $X = \{x_1, x_2, \ldots, x_n\}$ é representada por $H(X)$ que é calculada seguindo a Equação 4.4:

$$H\left(X\right) = -\sum_{i=1}^{n} P(x_i) \log P(x_i) \tag{4.4}$$

Onde:

- $P(x_i)$ é a função de massa de probabilidade do resultado x_i;
- $\log P(x_i)$ é o *surprisal* ou autoinformação de x_i.

Seja *E* o conjunto que contém todos os eventos da janela corrente, dado por:

$$E = \{e_1, e_2, \ldots, e_v\} \tag{4.5}$$

Onde:

- v é o número de eventos distintos na janela corrente; portanto, $v \neq n$ se houver mais do que um evento igual a outro na mesma janela;
- evento e_a é igual ao evento e_b, se $b_{a,j} = b_{b,j}; \forall j \in [1, k], \forall b_{a,j} \in e_a, \forall b_{b,j} \in e_b, k = |e_a| = |e_b|$.

Seja o número de ocorrências do evento e_i representado por $n_i, \forall i = [1, v]$. Nesse contexto serão calculadas as entropias dos eventos; no caso, a entropia de E, ***H(E)***. O valor da entropia local é calculado como indicado na Equação 4.6, onde n_i/n é a probabilidade de ocorrer o evento e_i.

$$H\left(E\right) = -\sum_{i=1}^{v} \frac{n_i}{n} \log \frac{n_i}{n} \tag{4.6}$$

4.3 Processamento das Séries Temporais de Entropia

O processamento da série temporal de entropia consiste na aplicação de um ou mais dos métodos que buscam descobrir padrões anormais de métricas. Estes métodos podem ser uma combinação de técnicas, como detecção de pico, processamento de sinal e análise de subespaço, por exemplo. Nesse trabalho, como estudado por Wang et al. [Wang et al. 2010], o tráfego anômalo será analisado de utilizando detecção visual de picos.

4.4 Avaliação de Desempenho

Nesta seção serão apresentadas a metodologia empregada para avaliar o desempenho da técnica de detecção de anomalias baseada em entropia, os resultados obtidos e uma análise estatística inferencial.

4.4.1 Metodologia

Este experimento consiste em analisar a entropia da métrica de largura de banda para o tráfego de rede de uma aplicação que requer taxa de bits constante (CBR), como uma teleconferência. Para efeitos de observação do sistema de detecção de anomalias baseado em entropia, admite-se que o comportamento normal para a largura de banda é em torno de 500 Kbps. Esse nível é baseado no acordo de nível de serviço (SLA) acordado entre cliente e fornecedor.

Foram considerados dois tipos de anomalias no contexto deste experimento: (i) injeção de tráfego elevado, tal como um intruso tentando simular que é parte da rede; (ii) supressão

de tráfego, destruindo boa parte dos pacotes da aplicação, como um intruso que tenta mitigar a execução normal do serviço.

A cada rodada do experimento foi gerada uma carga sintética aleatória de tráfego da aplicação CBR com duração de 15 minutos. Essa carga chega à máquina analisadora que executa o *software* para detecção de anomalias baseado em entropia por meio de espelhamento de portas do *switch* que encaminha os pacotes entre as fontes de tráfego. Foi realizado um monitoramento passivo da largura de banda. Em cada uma dessas rodadas foram injetadas 6 anomalias, sendo 3 do tipo *injeção de tráfego*, e 3 do tipo *supressão de tráfego*. Cada anomalia durou cerca de 1 minuto.

Esses valores foram escolhidos, porque eles podem fornecer indícios sobre a capacidade que o sistema de detecção de anomalias baseado em entropia tem de se adaptar às mudanças e aceitar novos padrões de tráfego como um fluxo **normal**, se o comportamento da métrica é constantemente repetido. O objetivo deste experimento é determinar os pontos fortes e fracos do processo de detecção baseado em entropia, propondo melhorias para o mecanismo.

Com relação aos pontos de falha, há 12 pontos em que a largura de banda sofre mudanças, cada um com duração de 1 minuto e que refletem as 3 anomalias por injeção e os restabelecimentos ao comportamento normal do tráfego da aplicação CBR, constituindo 6 mudanças, e às 3 anomalias por supressão de tráfego e as voltas à normalidade, também 6 mudanças. A série temporal da largura de banda utilizada como referência para esta avaliação pode ser observada na Figura 4.2.

Figura 4.2: Série temporal da largura de banda

As alterações do volume de tráfego coletado estão indicadas na figura por meio de setas. As setas orientadas à esquerda indicam uma redução no volume de tráfego com relação ao estado anterior e as orientadas à direita, um aumento de largura de banda. A largura de banda de 500 Kbps será considerada neste estudo como a largura de banda correta (normal). No entanto, o sistema de detecção de anomalias baseado em entropia não terá como fazer essa distinção, pois entende como uma mudança atípica. Portanto, alarmes gerados para esses pontos serão entendidos como corretos nesta análise. A primeira anomalia ocorre em torno do tempo 01:00.

Para analisar o comportamento do sistema, foi adotado um modelo de projeto de experimentos 2^3-fatorial, que compõe um total de 8 tratamentos diferentes. Para cada tratamento, foram realizadas 6 replicações de experimentos, que permitiram um nível de confiança de 95%, erro padrão de 5%. O modelo de condução dos experimentos foi o de Monte Carlo, pois é feita uma varredura de parâmetros para avaliar o sistema. Na Tabela 4.1 encontram-se os 3 fatores que foram estudados, cada um possuindo dois níveis. A largura de banda foi calculada em intervalos de 1 segundo.

Tabela 4.1: Tratamentos usados na análise do detector de anomalias de tráfego baseado em entropia.

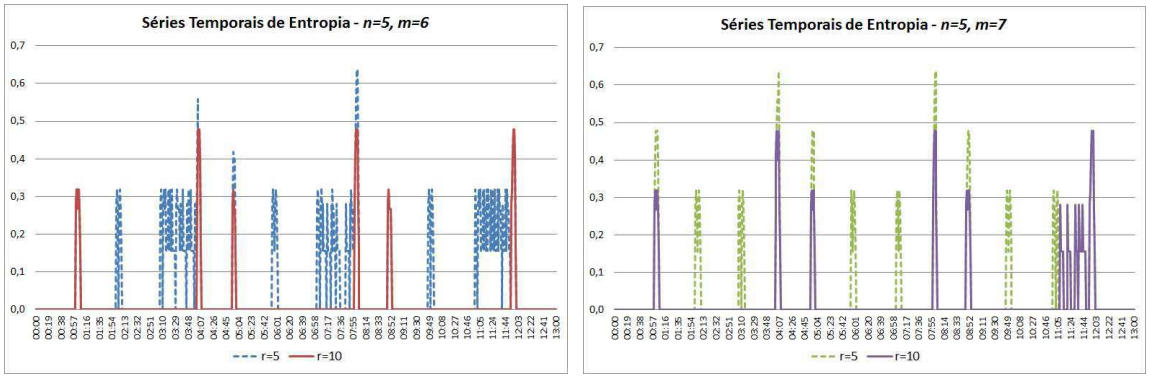
Parâmetro	**#Tratamento**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
n	5	5	5	5	10	10	10	10
m	6	6	7	7	6	6	7	7
r	5	10	5	10	5	10	5	10

4.4.2 Resultados Obtidos

Seguindo a metodologia apresentada para a detecção de anomalias baseada em entropia, após coletar pacotes e extrair as métricas de largura de banda, realiza-se o cálculo da entropia para as amostras em uma janela de valores e se compõe a série temporal de entropias para identificar padrões anômalos de tráfego.

Para fins de compreensão da ideia geral da técnica de simplificação, serão apresentadas as séries temporais obtidas apenas para a primeira réplica do experimento. Uma análise mais criteriosa da precisão dos resultados para todas as réplicas será apresentada na próxima seção.

Na Figura 4.3a estão ilustradas as séries temporais da entropia para os tratamentos 1 e 2 (vide Tabela 4.1). Na Figura 4.3b estão as séries temporais dos tratamentos 3 e 4, na Figura 4.4a os tratamentos 5 e 6 e na Figura 4.4b os tratamentos 7 e 8. As demais réplicas dos experimentos seguiram o mesmo padrão de geração de tráfego; portanto irão possuir uma série temporal de entropia que segue o mesmo comportamento.



(a) $n = 5, m = 6$ (b) $n = 5, m = 7$

Figura 4.3: Séries temporais de entropia para $n = 5$ e $m = 6$ (a) e $m = 7$ (b).

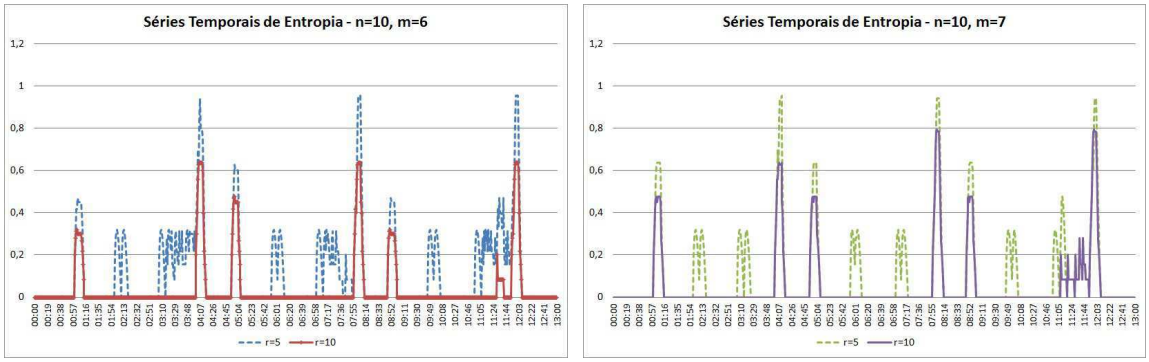
Alguns cenários são mais sensíveis a alterações de largura de banda e têm uma elevada taxa de falsos positivos que ocorrem quando $r = 5$.

Nos tratamentos 2, 4, 6 e 8, em que $m = 7$, e cujos resultados estão apresentados na Figura 4.3b não foram gerados alarmes na transição da largura de banda normal para a largura de banda com supressão de pacotes, as quais ocorreram por volta dos tempos 03:00, 07:00 e 11:00, o que significa a presença de falsos negativos. Esses FNs correspondem a 50% das anomalias injetadas.

Esse comportamento é uma consequência da razão r/m, que determina o número de compartimentos e, consequentemente, o grau de dispersão dos dados. Quando há valores muito altos na janela e apenas um muito pequeno, esse valor passa a ser diluído pelos demais e acaba passando “despercebido"(falso negativo). Com a continuação, essa transformação de dados torna-se cada vez mais suavizada e o sistema de detecção acaba se acostumando

com o novo comportamento do tráfego.

Os resultados apontam para uma preocupação com relação à configuração dos parâmetros do detector. Podemos observar que quando há uma anomalia de injeção de tráfego, o detector consegue verificar rapidamente a presença de mudanças na dispersão da métrica. Até que consiga se adaptar novamente ao novo comportamento, realiza diversos alarmes falsos. Uma solução proposta por [Wang et al. 2010] é eliminar os alarmes que são gerados sucessivamente.



(a) $n = 10, m = 6$ (b) $n = 10, m = 7$

Figura 4.4: Séries temporais de entropia para $n = 10$ e $m = 6$ (a) e $m = 7$ (b).

Sobre a escolha dos fatores, foram testadas exaustivamente várias combinações de parâmetros e, mesmo trabalhando bastante para obter uma configuração ótima (ou quase-ótima) para os parâmetros, não foi possível encontrar uma solução genérica que pudesse se adaptar a todo tipo de tráfego.

O ajuste de parâmetros do detector de anomalias baseado em entropia (n, r, m) depende do conhecimento do tráfego, tais como informações sobre padrões de tráfego a fim de compreender o comportamento irregular das séries temporais das métricas, bem como a duração de anomalias para ajustar corretamente o tamanho da janela, o número de classes e suas faixas. Sem conhecimento algum sobre o tráfego, fica bastante difícil interpretar uma série temporal de entropia. Esses pressupostos, todavia, são muito fortes para um sistema de previsão e constituem barreiras para a aplicação prática de tal técnica.

4.4.3 Estatística Inferencial

A métrica *F-Measure* (vide Capítulo 2) e o efeito de cada um todos três fatores do projeto experimental para explicação da acurácia da técnica de detecção de anomalias baseada em entropia são analisados nesta seção.

Os resultados obtidos para a *métrica F* nas 6 replicações do experimento estão apresentados na Tabela 4.2. Para estudos iniciais sobre os resultados obtidos, foi adotado um modelo de regressão linear para predizer o valor da variável de resposta *F-Measure*, apresentada na Equação 2.1. Esse modelo mostrou-se aceitável para a *métrica F*, mas não foi adequado para os modelos das métricas de precisão e completude. Estudos posteriores podem ser feitos na tentativa de obter um modelo matemático adequado para o comportamento dessas métricas. Os erros experimentais levando em consideração um modelo de regressão linear simples [Jain 1991] para a *métrica F* estão apresentados na Tabela 4.3.

Tabela 4.2: Valores de cada medição para a Métrica F.

F-Measure						
#Tratamento (i)	y_{i1}	y_{i2}	y_{i3}	y_{i4}	y_{i5}	y_{i6}
1	0,400000000	0,385964912	0,357142857	0,428571429	0,379310345	0,392857143
2	0,666666667	0,631578947	0,600000000	0,600000000	0,666666667	0,631578947
3	0,685714286	0,705882353	0,685714286	0,685714286	0,685714286	0,727272727
4	0,521739130	0,545454545	0,500000000	0,500000000	0,52173913	0,545454545
5	0,536585366	0,511627907	0,43902439	0,571428571	0,585365854	0,550000000
6	0,631578947	0,666666667	0,600000000	0,600000000	0,631578947	0,631578947
7	0,705882353	0,705882353	0,727272727	0,705882353	0,750000000	0,750000000
8	0,521739130	0,545454545	0,52173913	0,52173913	0,545454545	0,571428571

Cálculo dos Efeitos

Foi realizado o **cálculo do percentual de variação dos efeitos**, que corresponde ao percentual que cada efeito contribui para o resultado final. A Tabela 4.4 contém o percentual de variação de cada efeito.

Os efeitos significativos no resultado são aqueles que correspondem a mais do que 5% do percentual de variação. Portanto, os efeitos significativos são o número de classes (B) e a interação entre o número de classes e a faixa (interação entre B e C). O efeito dos erros na variação total corresponde a cerca de 6%.

Tabela 4.3: Erros experimentais obtidos para a Métrica F.

#Tratamento (i)	e_{i1}	e_{i2}	e_{i3}	e_{i4}	e_{i5}	e_{i6}
1	0,009359	-0,004676202	-0,0335	0,03793	-0,011330	0,002216
2	0,033918	-0,001169591	-0,03275	-0,03275	0,033918	-0,001170
3	-0,010290	0,009880316	-0,01029	-0,01029	-0,010290	0,031271
4	-0,000660	0,023056653	-0,02240	-0,02240	-0,000660	0,023057
5	0,004247	-0,020710774	-0,09331	0,03909	0,053027	0,017661
6	0,004678	0,039766082	-0,02690	-0,02690	0,004678	0,004678
7	-0,018270	-0,018270945	0,003119	-0,01827	0,025847	0,025847
8	-0,016190	0,007528703	-0,01619	-0,01619	0,007529	0,033503

Tabela 4.4: Percentuais de variação dos efeitos.

Efeito	**Variável**	**Variação do Efeito**
A	Tamanho da janela (n)	0,045930003
B	**Número de classes** (m)	**0,126422317**
C	Tamanho da faixa (r)	0,000764523
AB	Interação entre A e B	0,01210629
AC	Interação entre A e C	0,036558141
BC	**Interação entre B e C**	**0,69130786**
ABC	Interação entre A, B e C	0,025941494
E	Erro experimental	0,060969373
Total		1

Desvio Padrão para os Erros

O desvio padrão para os erros, $s_e = 0,028323209$.

Cálculo dos Intervalos de Confiança para as Variáveis de Resposta

As médias para as métricas do processo de detecção de anomalias baseado em entropia estão apresentadas na Tabela 4.5. Essas variáveis de resposta foram calculadas com um nível de confiança de 95%. A Figura 4.5a apresenta os resultados obtidos para a métrica F. Como complemento, os resultados para as métricas *precision* e *recall* são apresentados nas Figuras 4.5b e 4.5c, respectivamente.

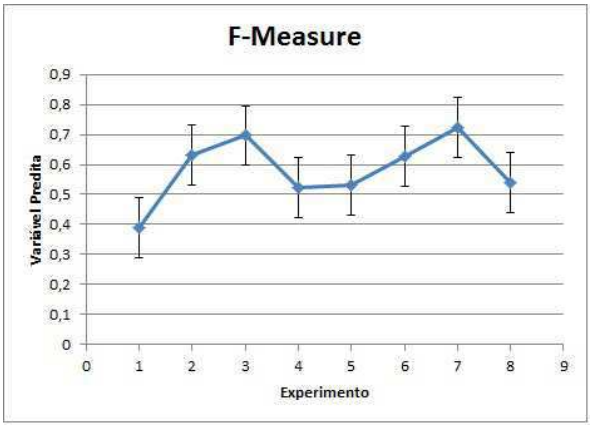
Tabela 4.5: Média obtida para as métricas *recall*, *precision* e *F1* usando detecção de anomalias baseada em entropia.

Métrica	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
#Alarmes	45,2	7,00	22,5	11,0	29,3	7,17	21,2	10,3
#Detecções com sucesso	11,2	6,00	12,0	6,00	11,0	6,00	12,0	6,00
Recall	0,93	0,50	1,00	0,50	0,92	0,50	1,00	0,50
Precision	0,25	0,87	0,53	0,55	0,38	0,85	0,57	0,58
F1 Score	0,39	0,63	0,70	0,52	0,53	0,63	0,72	0,54
FAR	0,75	0,13	0,47	0,45	0,62	0,15	0,43	0,42

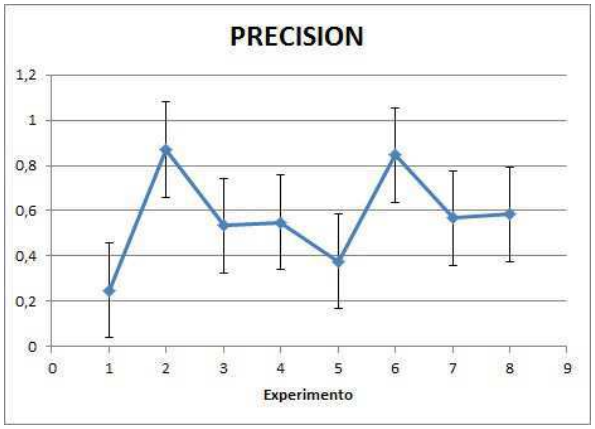
Modelo de Regressão Linear

Com base nos resultados obtidos, identificou-se que a métrica F poderia ser modelada por meio de regressão linear simples [Jain 1991], mesmo sendo uma composição de duas outras métricas não lineares. O modelo de regressão linear obtido para a métrica F é apresentado na Equação 4.7. Foram realizados arredondamentos nos coeficientes da equação.

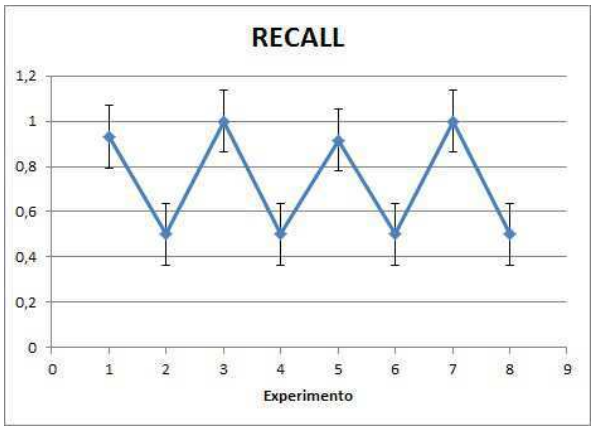
$$F_1 = 0,583 + 0,022n + 0,037m - 0,003r - 0,012nm - 0,02nr - 0,087mr + 0,017nmr \tag{4.7}$$



(a) *F-Measure*



(b) *Precision*



(c) *Recall*

Figura 4.5: Intervalos de confiança para as estimativas dos valores das variáveis *F-Measure*, *precision* e *recall*.

Este modelo apresenta muitos produtos envolvendo as variáveis, portanto não é um modelo parsimônico, que pode não ser generalizado para vários casos e constituir uma ameaça à validade das conclusões obtidas.

4.5 Conclusões

No trabalho de Wang et al. [Wang et al. 2010], as configurações da faixa para acomodação das métricas normalizadas, r, e do número de classes, m, são feitas usando valores constantes. Eles afirmam que há uma maneira para determiná-los estatisticamente; no entanto, tal passo ficou em aberto no seu trabalho.

A métrica *F-measure* não leva em consideração o número de verdadeiros-negativos. Portanto, para descrever melhor a qualidade dos algoritmos de predição, devem ser analisadas também outras métricas, como a taxa de falso-positivo, a especificidade, o valor de predição negativo e a acurácia. Esta última estabelece a razão entre o número total de predições corretas (tanto positivas quanto negativas) e o número total de predições.

Os resultados mostram que há a necessidade de melhor analisar e compreender os padrões de tráfego, descobrir o comportamento normal dos sistemas monitorados, com base em previsões usando dados históricos ou *feedback* de especialistas sobre o negócio que está vinculado ao tráfego de rede, ou fazendo novas suposições sobre o tráfego.

Como trabalho futuro, pretende-se analisar a genericidade do modelo de regressão linear para a métrica F e, caso aplicável, identificar os modelos que melhores se adequam às variáveis em estudo.

Capítulo 5

EMATADE: Arquitetura de Detecção de Anomalias baseada em Entropia e Aprendizagem de Máquina

> "O objetivo da ciência não é o de abrir portas para a sabedoria infinita, mas o de estabelecer limites para o erro infinito."
>
> Bertolt Brecht

Argumenta-se que a detecção de anomalias com base no modelo de Gauss pode ajudar a definir os parâmetros de entrada do EbAT. Pode-se usar o EbAT para rotular o conjunto de validação cruzado definido de forma adaptativa. Por outro lado, à medida que o tempo passa, ambas as técnicas trabalharão em sinergia. É difícil configurar os parâmetros de entrada da técnica EbAT e validar os alarmes identificados. Por outro lado, a partir da perspectiva de aprendizado de máquina, é difícil rotular as amostras, sem conhecimento prévio.

Neste capítulo é proposta uma técnica chamada EMATADE (*Entropy and Machine Learning-based Anomaly Detection Engine*) de detecção de anomalias híbrida, que se baseia na técnica de detecção de anomalias baseada em entropia e em aprendizagem de máquina. Essa técnica e os resultados obtidos até o momento foi publicada por Oliveira *et al.* [Oliveira et al. 2014b].

5.1 EMATADE: Detecção Híbrida de Anomalias

A técnica EMATADE é composta por cinco fases:

I. Captura de tráfego;

II. Estimativa de limiar utilizando aprendizagem de máquina;

III. Monitoramento do tráfego dos serviços da nuvem;

IV. Estimativa de entropia;

V. Detecção de anomalias.

A ideia é que os alarmes gerados pelo EbAT alimentarão o subprocesso de rotular os pacotes de tráfego anômalo e que o limiar obtido pela técnica baseada em aprendizagem de máquina também se tornará uma prova ao testar se as métricas de serviço apresentam anomalias de tráfego. A Figura 5.1 apresenta a técnica EMATADE, o processo de detecção de anomalias híbrido que utiliza EbAT (Seção 4.1) e aprendizagem de máquina (Seção 5.2).

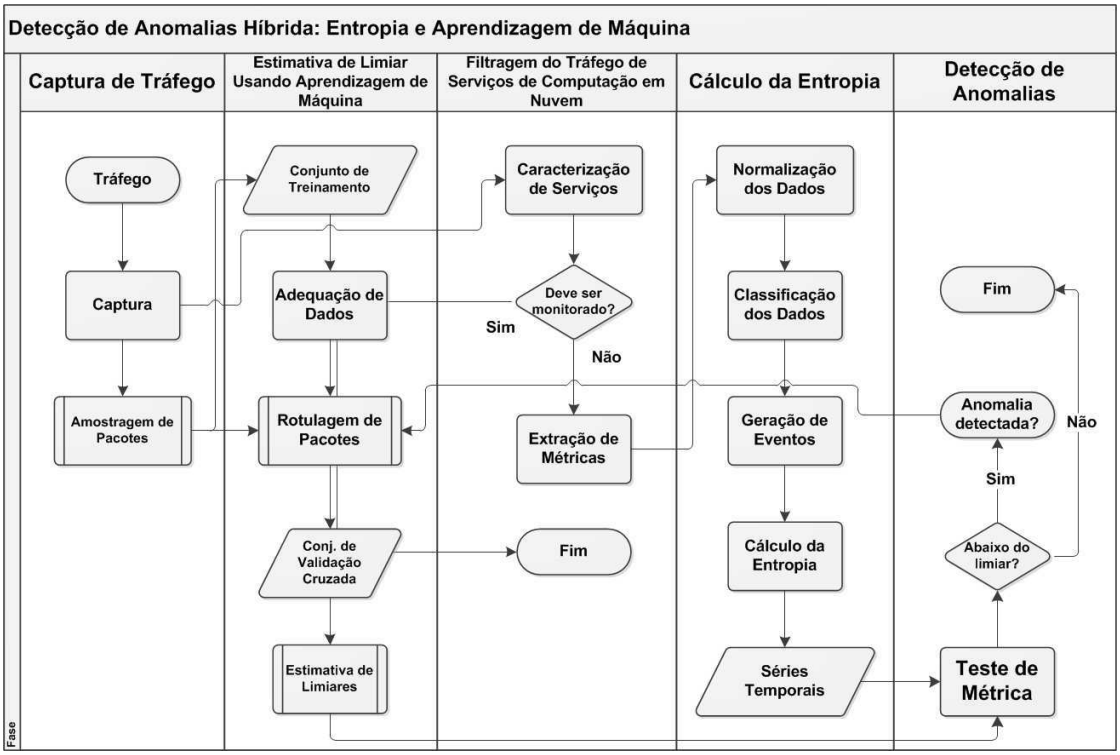


Figura 5.1: EMATADE: método híbrido proposto para detecção de anomalias.

5.2 Detecção de Anomalias por Aprendizagem de Máquina

A técnica de aprendizagem de máquina não supervisionada para a detecção de anomalias é baseda no ajuste dos dados a uma distribuição normal (Gaussiana). Os valores que possuem probabilidade muito baixa são considerados anomalias. O objetivo desta análise de probabilidade é encontrar um limiar de probabilidade, que maximiza a acurácia da detecção. Nesta seção será descrito como se pode implementar esta técnica.

Serão coletadas as *métricas* de rede (ou *características de desempenho*) que comporão o *conjunto de treinamento (TS)*. O próximo passo é calcular a probabilidade de cada valor, fazendo o ajuste dos mesmos à distribuição Gaussiana. Essa distribuição será utilizada por explicar números fenômenos e eventos do mundo real, sendo amplamente utilizada no contexto de experimentação e é capaz de prover indícios sobre o comportamento normal ou não do tráfego.

Outras distribuições poderiam ser empregadas após um estudo analítico que modele o tráfego de serviços de computação em nuvem. Há indícios de que o tráfego normal de uma rede comporta-se como uma distribuição de cauda pesada chamada Weilbull [Arshadi e Jahangir 2014b] [Arshadi e Jahangir 2014a][Arfeen et al. 2013], mas essa análise de conformidade está fora do escopo deste estudo. No entanto, a proposta de detecção de anomalias utilizando aprendizagem de máquina deste capítulo é genérica. Pode-se substituir as distribuições de acordo com o comportamento real do tráfego.

Dado o *conjunto de treinamento* $x^{(1)}, ..., x^{(m)}$ (onde $x^{(i)} \in R_n$), será calculada a distribuição Gaussiana para cada uma das características de desempenho x_i. Para cada métrica $i = 1, ..., n$, serão calculados os parâmetros μ_i e σ_i^2 para cada i-ésima dimensão do conjunto de tratamento $x_i^1, ..., x_i^m$ (ou seja, as amostras coletadas para a métrica i). A função de distribuição Gaussiana é dada por (5.1), onde μ é a média e σ^2 é a variância. Os parâmetros μ_i e σ_i^2 da i-ésima métrica são estimados segundo as Equações (5.2) e (5.3), respectivamente [Xi et al. 2009][Ng 2014].

$$p(x; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{5.1}$$

$$\mu_i = \frac{1}{m} \sum_{j=1}^{m} x_i^{(j)} \tag{5.2}$$

$$\sigma_i^2 = \frac{1}{m}\sum_{j=1}^{m}(x_i^{(j)} - \mu_i)^2 \tag{5.3}$$

Depois de calcular a média e a variância, ajustam-se os dados ao modelo de Gauss. Em seguida, observam-se quais os valores têm uma probabilidade muito elevada de acordo com a Distribuição de Gauss, e quais tem uma probabilidade muito baixa. As amostras de baixa probabilidade são anomalias de acordo com um *limiar*. O limiar,ϵ, maximiza a acurácia em um *conjunto de validação cruzada* [Xi et al. 2009].

Seja o conjunto de validação cruzada $CV = \{(x_{cv}^{(1)}, y_c v^{(1)}), \ldots, (x_{cv}^{(m_{CV})}, y_c v^{(m_{cv})})\}$, onde o rótulo $y = 1$ corresponde a uma amostra de métrica anômala, e $y = 0$ corresponde a uma amostra normal. Para cada elemento de validação cruzada, computou-se $p(x_{cv}^i)$, que é a massa de probabilidade daquele elemento de acordo com a distribuição de Gauss. Seja o vetor de todas essas probabilidades $P = \left\langle p(x_{cv}^{(1)}), \ldots, p(x_{cv}^{(m_{cv})}) \right\rangle$.

Definiu-se o limiar de probabilidade, ϵ, selecionando-o a uma distância a partir dos valores mínimos e máximos para $p(x_{cv}^i) \in P$. O valor ϵ que maximiza a acurácia será escolhido como um indicador de anomalia para o processo de detecção [Xi et al. 2009].

5.3 Metodologia

Nesta seção será descrita a metodologia para a condução dos experimentos para validar a técnica proposta. A acurácia do EMATADE será avaliada pela métrica *F*. Ela representa uma média harmônica [Salfner et al. 2010] e foi introduzida na Seção 2.5. Quanto maior o valor da métrica F, maior será a qualidade do detector.

Para analisar o comportamento do sistema, adotamos o modelo de projeto experimental 2^3-fatorial que avalia 8 tratamentos diferentes. Três fatores foram avaliados (n, m e r), cada um variando em dois níveis, conforme Tabela 5.1. Esses valores foram escolhidos para compatibilizar com os estudos realizados no Capítulo 4.

A técnica proposta neste capítulo foi avaliada utilizando dois tipos de experimentos que envolveram simulações e configuração de um ambiente real para produzir dados de entrada que refletissem o comportamento real de serviços. A primeira empregou um gerador de tráfego para simular um ataque de negação de serviço (DoS) à um sistema de computação em nuvem que provia Dados-como-Serviço. A segunda e por meio de um arquivo de rastro

Tabela 5.1: Tratamentos avaliados.

Parâmetro	**#Tratamento**							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
n	5	5	5	5	10	10	10	10
m	6	6	7	7	6	6	7	7
r	5	10	5	10	5	10	5	10

proveniente de um ataque de negação de serviço real a um provedor de telecomunicações. Esses experimentos estão descritos nas Seções 5.3.1 e 5.3.2.

5.3.1 Simulação de Ataque de Negação de Serviço à Nuvem

Para avaliar a acurácia da técnica proposta foi realizado um ataque sintético VM-a-VM usando a ferramenta de código aberto Hping3 [Hping 2014] dentro de um sistema em nuvem para DaaS executando um aplicativo chamado Nutch, que rastreia a Web à procura de páginas. A ferramenta Hping3 permite gerar pacotes arbitrários para fazer um ataque de DoS às máquinas vítimas. A experimentação reproduziu o cenário avaliado por Quoc *et al.* [Quoc et al. 2014].

Configurou-se o Hping3 em três VMs numa nuvem privada que executavam aplicações Hadoop para processamento das páginas Web para gerar pacotes SYN TCP para atacar duas VMs rodando o servidor Hadoop em cada uma delas. As VMs fazem parte da mesma nuvem, incluindo um nó mestre e um nó escravo, como mostrado na Figura 5.2. Os serviços de processamento das páginas Web foram configurados de modo real usando Nutch e Hadoop, contudo, os pacotes de ataque entre as VMs eram sintéticos. Os horários dos ataques estão detalhados na Tabela 5.2 [Quoc et al. 2014].

Em seguida, foram medidas as seguintes variáveis:

1. Número de pacotes do Hadoop Distributed File System (HDFS);

2. Número de pacotes da Aplicação MapReduce.

A taxa de pacotes dessas duas aplicações é apresentada na Figura 5.3a. Quando o volume de tráfego de uma aplicação cresce, o da outra também cresce, portanto essas taxas de

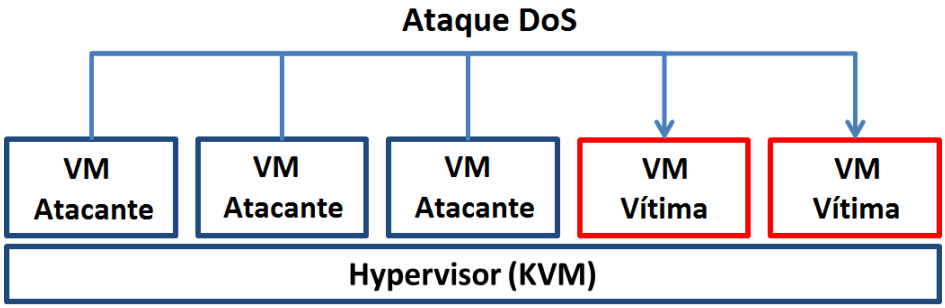


Figura 5.2: Nós que fazem parte do ataque de DoS [Quoc et al. 2014].

Tabela 5.2: Lista de Ataques de TCP SYN FLOOD [Quoc et al. 2014].

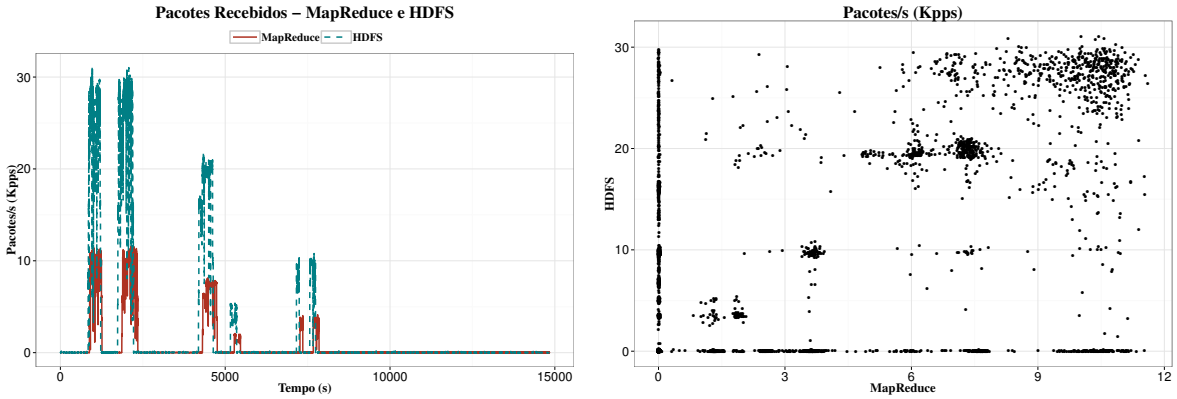
	Tempo Inicial	Tempo Final	Tempo Inicial Relativo (s)	Tempo Final Relativo (s)	Atacantes	Vítimas
1	18:23:32	18:29:10	0	338	VM3, VM4, VM5	VM1, VM2
2	18:40:14	18:47:58	1.002	1.466	VM3, VM4, VM5	VM1, VM2
3	19:20:35	19:28:15	3.423	3.883	VM4, VM5	VM1
4	19:36:29	19:39:35	4.377	4.563	VM3	VM1, VM2
5	20:09:37	20:11:06	6.365	6.454	VM4	VM1
6	20:16:36	20:19:08	6.784	6.936	VM4	VM1

pacotes estão correlacionadas, no entanto, elas não estão sincronizadas. É possível observar os momentos de ataque nos 6 picos de tráfego do gráfico. O gráfico de dispersão contendo os pacotes do MapReduce versus os pacotes da aplicação HDFS é mostrado na Figura 5.3b. Há um agrupamento ortogonal entre as taxas de pacotes e uma área em que estão distantes das origens dos eixos X e Y. Essa área indica que esses pacotes configuram um possível ataque.

Cálculo de Entropia

As séries temporais de entropia para o par de métricas medidas quando $n = 5$ foram calculados e estão apresentados na Figura 5.4 e quando $n = 10$ estão apresentados na Figura 5.5. Os resultados são difíceis de interpretar visualmente e de estabelecer os parâmetros adequados para propor os valores anômalos.

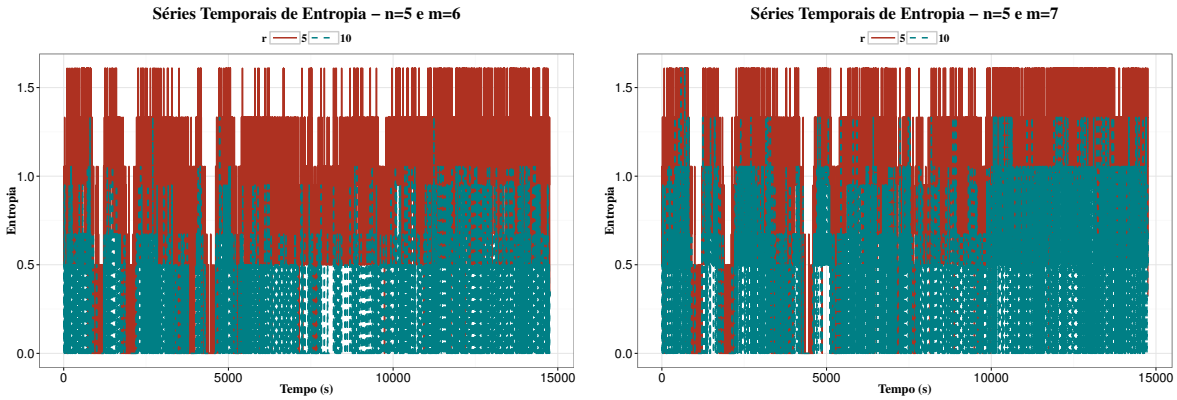
Por isso, é importante aplicar uma técnica para identificar automaticamente quando, em termos de limiares, considerar que a série temporal de entropia passa a indicar um compor-



(a) Séries temporais de entropia

(b) Gráfico de dispersão de pacotes por segundo: MapReduce versus HDFS

Figura 5.3: Taxa de pacotes das aplicações MapReduce versus HDFS.



(a) $n = 5; m = 6$

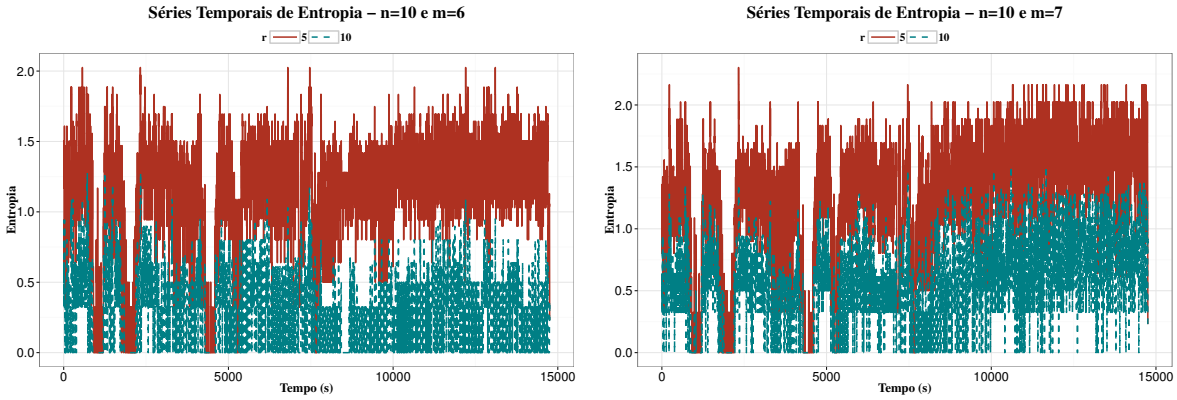
(b) $n = 5; m = 7$

Figura 5.4: Séries temporais de entropia para $n = 5$ com: $m = 6$ (a) e $m = 7$ (b).

tamento anômalo ou não.

Resultados Obtidos

Utilizando a abordagem de aprendizagem de máquina sobre as séries temporais de entropia, todos os TPs foram identificados independemente dos parâmetros. A Tabela 5.3 são apresentados os resultados obtidos para os oito tratamentos diferentes, quando utilizada a técnica EbAT pura. O cenário 3 (Tratamento 3) apresenta os melhores resultados, isto é, a precisão é de 100%. As métricas avaliadas foram explicadas na Seção 2.5 No entanto, há outros que fornecem boa precisão nos resultados, como, por exemplo, mais de 90%.



(a) $n = 10$; $m = 6$ (b) $n = 10$; $m = 7$

Figura 5.5: Séries temporais de entropia para $n = 10$ com: $m = 6$ (a) e $m = 7$ (b).

Tabela 5.3: Resultado do processo de detecção de anomalias no ataque de DoS à nuvem.

#Tratamento	VP	FP	FN	*Recall*	*Precision*	F1
1	9967	711	1077	0.902481	0.933414	0.917687
2	543	0	10501	0.0491670	1	0.093726
3	**11044**	**0**	**0**	**1**	**1**	**1**
4	2181	0	8863	0.1974828	1	0.329830
5	10397	1749	647	0.941416	0.856002	0.896680
6	153	9	10891	0.013854	0.944444	0.027308
7	10638	1773	406	0.963238	0.857143	0.907099
8	2402	155	8642	0.217494	0.939382	0.353209

5.3.2 Ataque Real de Negação de Serviço Distribuído a um Provedor de Telecomunicações

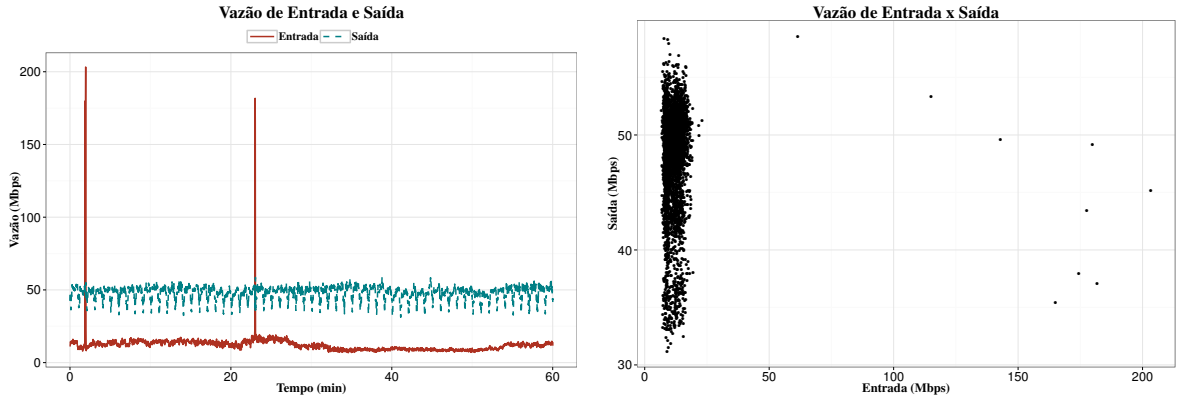
A técnica de detecção de anomalias EbAT foi avaliada ainda por meio de um ataque de negação de serviço real. A carga de tráfego analisada é proveniente de uma captura de tráfego contendo ataque de DDoS realizado à *Pohang University of Science and Technology* (POSTECH). Esse ataque ficou famoso, pois foi realizado contra sítios governamentais e comerciais na Coreia do Sul em 7 de julho 2009. Esses ataques provavelmente foram lançados por uma unidade especial de guerra cibernética pertencente ao Exército norte-coreano. Du-

rante o ataque, muitos computadores na rede do campus da POSTECHs tornaram-se zumbis. Analisou-se uma hora de captura de tráfego rede contendo os pacotes do ataque [Quoc et al. 2011].

Mediram-se duas características do tráfego da rede:

1. Vazão de entrada;
2. Vazão de saída.

Para compreender o volume de tráfego normal e como o ataque de DDoS alterou-o, a vazão de entrada e de saída de tráfego da rede do *campus* estão apresentadas na Figura 5.6a. Há dois picos com duração de poucos segundos. Juntos, o ataque total durou cerca de 13 segundos.



(a) Séries temporais de entropia (b) Gráfico de dispersão: vazão de entrada versus saída

Figura 5.6: Vazão de entrada e de saída.

No gráfico de dispersão entre a vazão de entrada e de saída, que se encontra na Figura 5.6b, há uma área que apresenta 9 pontos que está distante de um padrão normal entre o tráfego de entrada versus saída. Esses pontos são indícios de uma anormalidade no tráfego. Contudo, como identificar esse tipo de comportamento em uma análise de tráfego *online*?

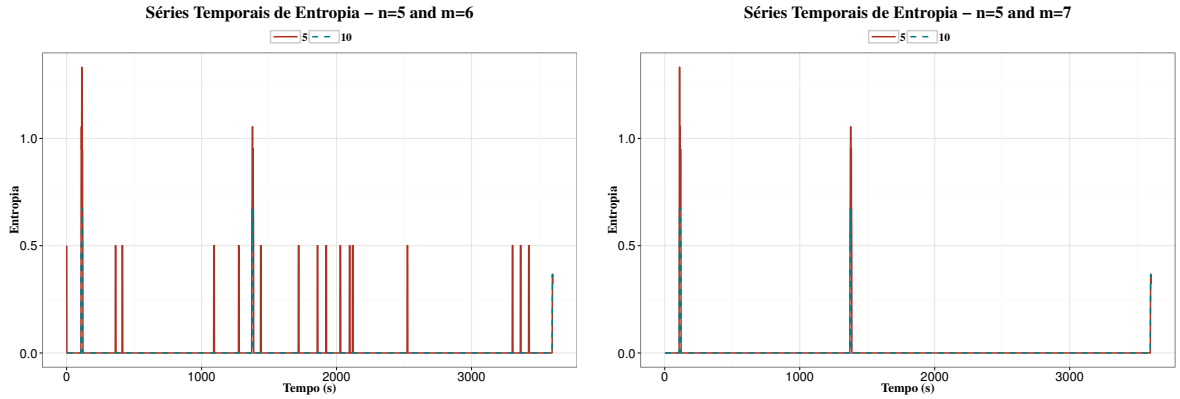
Aprendizagem de Máquina

A técnica de aprendizagem de máquina não-supervisionada aplica as métricas de tráfego a uma distribuição normal e verifica quais desses valores estão distantes do comportamento esperado.

Aplicou-se a distribuição Gaussiana em um conjunto de validação cruzada para identificar o limiar de probabilidade, quando os valores amostrados seguem uma distribuição normal. Em seguida, utilizamos este valor para prever quais pacotes participaram do ataque DoS.

Cálculo de Entropia

As Figuras 5.7 e 5.8 apresentam os resultados do cálculo da entropia para as séries temporais para $n = 5$ e $n = 10$, respectivamente.



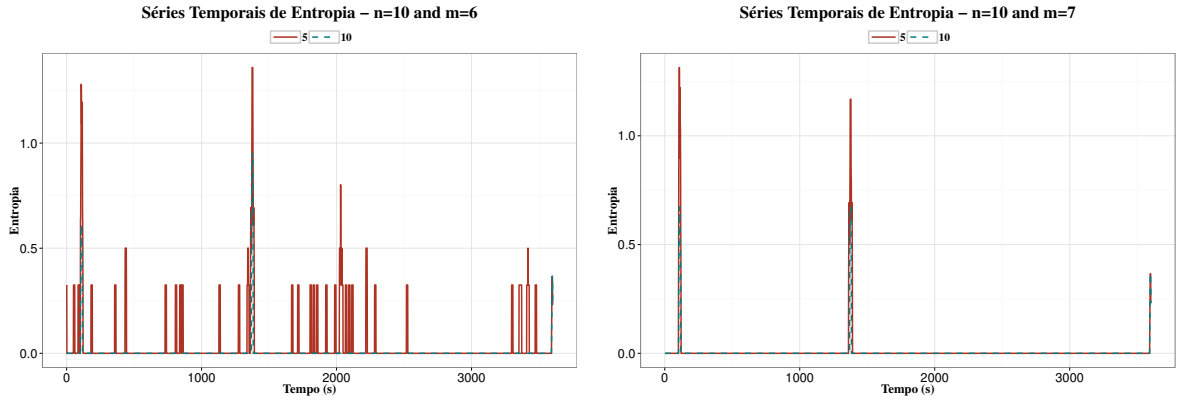
(a) $n = 5$; $m = 6$ (b) $n = 5$; $m = 7$

Figura 5.7: Séries temporais de entropia para $n = 5$ com: $m = 6$ (a) e $m = 7$ (b).

Detectar de anomalias com o mecanismo baseado em entropia puro é complicado, pois é difícil se estabelecer limites claros para o que é ou não uma anomalia dadas apenas as séries temporais de entropia. Isso se torna ainda mais difícil quando os dados são densos e sensíveis a mudanças, constituindo barreiras para a implementação prática de tal técnica.

Resultados Obtidos

Os 9 pontos que podiam ser observados no gráfico da Figura 5.6b são na prática 13 pontos (há sobreposição de alguns). Eles representam os verdadeiros positivos (TPs) na análise. Utilizando a abordagem de aprendizagem de máquina sobre as séries temporais de entropia, todos os TPs foram identificados independemente dos parâmetros. Na Tabela 5.4 são apresentados os resultados médios obtidos para os oito tratamentos diferentes, quando utilizada a técnica EbAT pura.



(a) $n = 10$; $m = 6$ (b) $n = 10$; $m = 7$

Figura 5.8: Séries temporais de entropia para $n = 10$ com: $m = 6$ (a) e $m = 7$ (b).

É possível utilizar a técnica EbAT pura após uma compreensão das melhores configurações de parâmetros para o sistema de detecção. Neste caso, existem dois tratamentos que contribuem para os melhores resultados de precisão, que são o primeiro e terceiro cenário (em negrito). Dessa forma, limitou-se o escopo e se pode configurar com propriedade os parâmetros que levam o sistema a fornecer as melhores previsões sobre a presença de anomalias no tráfego. Neste caso, foi selecionado o terceiro cenário que possui os parâmetros $n = 5$, $m = 7$ e $r = 5$. Nessas configurações, a métrica F atingiu 100%.

5.4 Conclusões

Sistemas de computação em nuvem tem características peculiares, como a agregação de muitos serviços diferentes, o que torna difícil classificar aplicações por meio de técnicas baseadas na carga útil da assinatura do pacote correspondente, ou métodos probabilísticos para a identificação de padrões de tráfego e perfil de comportamentos normais de tráfego. Essas características trazem desafios em termos de monitoramento e análise *online* de tráfego, que deve ser realizada na velocidade de redes gigabits, possuindo um grande volume de dados a ser processado.

Escolher a melhor técnica de detecção de anomalia de tráfego é uma tarefa complexa, porque para se obter bons resultados, devem-se conhecer várias características do tráfego, que na prática não são conhecidas. Um outro desafio é realizar amostragem de pacotes de

Tabela 5.4: Resultado do processo de detecção de anomalias no ataque de DoS ao provedor de Internet.

#Tratamento	TP	FP	FN	Recall	Precision	F1
1	**13**	**0**	**0**	**1**	**1**	**1**
2	0	0	13	0	NA	NA
3	**13**	**0**	**0**	**1**	**1**	**1**
4	0	0	13	0	NA	NA
5	12	9	1	0.923077	0.571429	0.705882
6	5	2	8	0.384615	0.714286	0.5
7	10	8	3	0.769231	0.555556	0.645161
8	0	0	13	0	NA	NA

rede de alto desempenho e aprimorar o processamento dos dados, uma vez que o volume de tráfego é da ordem de gigabits por segundo.

Os resultados obtidos mostram que, ao aplicar a técnica de EbAT pura, ainda é preciso analisar e compreender melhor os padrões de tráfego, descobrir o comportamento normal dos sistemas monitorados, com base em previsões utilizando histórico dados, ou *feedback* de especialistas sobre o tráfego de rede e sobre o negócio, ou fazer novas hipóteses sobre o tráfego. Neste sentido, argumenta-se que a acurácia dos resultados pode ser melhorada com o auxílio de modelos de probabilidade, tais como detecção de anomalias utilizando o modelo gaussiano.

Uma ameaça à validade do trabalho é o impacto que pode ocorrer caso o tráfego não siga a distribuição Gaussiana. Como trabalho futuro, a técnica EMATADE será implementada supondo que o tráfego segue a distribuição Weilbull, que possui cauda pesada, e comparados aos resultados obtidos aplicando a distribuição Gaussiana.

Capítulo 6

Escalonamento baseado em Custo

"All truth passes through three stages. First, it is ridiculed. Second, it is violently opposed. Third, it is accepted as being self-evident."

Arthur Schopenhauer

Nesta tese foi realizado um levantamento do custo cobrado para o modelo de negócio de Dados-como-Serviço (*Data-as-a-Service* – DaaS). O modelo de custo foi usado como um estudo de caso para avaliar prejuízos causados por anomalias de tráfego considerando esse tipo de modelo de negócio de computação em nuvem. Neste capítulo será apresentado o modelo de custo para Dados-como-Serviço e um mecanismo de escalonamento baseado nos custos para alocar máquinas virtuais. O mecanismo de escalonamento considera os custos para atender aos requisitos de processamento, armazenamento e comunicação de cada VM, tendo-se como objetivo minimizar os custos. O modelo de custo e o escalonamento baseado nesse modelo foram contribuições deste trabalho e publicadas por Oliveira *et al.* [Oliveira et al. 2015b].

6.1 Modelo de Custo para Dados-como-Serviço

Nesta seção serão apresentados o modelo de custo para DaaS e o algoritmo de escalonamento baseado no mesmo, onde as *tarefas* (tratadas por *máquinas virtuais*) como, por exemplo, rastreamento e processamento, são escalonadas em um conjunto de micronuvens. Os resultados

obtidos apresentaram economia de custos em comparação com a abordagem tradicional de alternância circular (*round robin*) [Oliveira et al. 2015b].

6.1.1 Modelo do Sistema

Uma **nuvem** pode ser representada como um conjunto de **centros de dados**, cada um possuindo máquinas físicas para hospedar as máquinas virtuais e para armazenamento de dados. O termo ***micronuvem*** representa um pequeno conjunto de elementos de processamento localizados na mesma rede local (LAN), sem necessariamente significar a infraestrutura de um centro de dados complexo. Contudo, sem perda de generalidade, para compatibilizar conceitos, será utilizado o termo ***centro de dados*** ao longo deste trabalho com o mesmo conceito de *micronuvem*. Desse modo, pode-se representar genericamente diferentes tipos de nuvens; incluindo nuvens privadas, públicas ou federadas.

Um trabalho (*job*) em sistemas de DaaS é genericamente representado por uma consulta (*query*) a grandes quantidades de dados que é realizada de forma distribuída (*Big Data*). As consultas são divididas em tarefas, e cada tarefa é executada por meio de uma **máquina virtual** (VM). O processamento de uma consulta segue o estilo de programação *MapReduce* [Dean e Ghemawat 2008]. Para realizar uma consulta, uma ou mais VMs lêem e processam os dados e, em seguida, enviam os resultados para o destino por meio da rede de comunicação. O processo de executar uma computação sobre um conjunto de dados é a fase de *map* (mapear) e o processo de redução dos dados mapeados para o cálculo do resultado final é chamado de *reduce* (reduzir). O modelo *MapReduce* está sendo amplamente aplicado para processar *Big Data* [Hurwitz et al. 2013].

Para executar uma tarefa, uma VM necessita acessar uma fonte de dados localizada na área de armazenamento. As **fontes de dados** (*data sources*) utilizadas para o processamento das tarefas são replicadas e encontram-se distribuídas pelos centros de dados. Neste modelo, uma tarefa acessa uma cópia dos dados replicados. As fontes de dados são persistidas usando um modelo de armazenamento baseado em **chave/valor** ou *Key/Value (K/V) store*. Exemplos de ferramentas que implementam o modelo de *K/V store* são a *Google BigTable* [Chang et al. 2006] e o arcabouço *Chubby* [Burrows 2006].

6.1.2 Modelo de Custo

Na visão do cliente de DaaS, os custos do serviço são entendidos como *custos*. Esses custos se baseiam em dois pressupostos:

I. **VM e fonte de dados estão localizados no mesmo centro de dados**: se as VMs e os dados estiverem localizados no mesmo centro de dados, então não há custos de comunicação;

II. **VM e fonte de dados estão localizados em centros de dados diferentes**: os custos de comunicação são aplicáveis somente quando a VM enviar dados fora do centro de dados em que está localizada (em outra rede local).

Para realizar uma consulta, a VM lê e processa os dados e, em seguida, envia os resultados da consulta para o destino por meio da rede de comunicação. O processo de executar uma computação sobre um conjunto de dados é a fase de *map* (mapear) e o processo de redução dos dados mapeados para o cálculo do resultado final é chamado de *reduce* (reduzir). O modelo MapReduce está sendo amplamente aplicado para processar grandes quantidades de dados de forma distribuída, como no processamento de *Big Data* [Hurwitz et al. 2013].

Os componentes do modelo de precificação para dados como serviço são apresentados a seguir:

- Seja D o conjunto de centros de dados:
  - $D = \{d_1, d_2, \ldots, d_{|D|}\}$;
- Seja U o conjunto de VMs utilizadas para consultas:
  - $U = \{u_1, u_2, \ldots, u_{|U|}\}$;
- Seja S o conjunto de fontes de dados:
  - $S = \{s_1, s_2, \ldots, s_{|S|}\}$.

As **variáveis independentes** do modelo são as seguintes:

- R_z: número, em *bytes*, de dados recebidos (lidos) pela VM u_z;

- T_z: número, em *bytes*, de dados transmitidos (escritos) pela VM u_z;
- r_z: fator de redução de dados da VM u_z; ou seja, a razão entre T_z e R_z;
- k_z: a taxa de tempo para processar 1 byte lido pela VM u_z; ou seja, a VM u_z requer $k \cdot R$ horas para ser finalizada;
- q_z: taxa de saída de bytes por hora da VM u_z, onde $q_z = \frac{r_z}{k_z}$;
- v_{d_i} ou v_i: custo de execução de uma VM durante 1 hora no centro de dados d_i;
- t_{d_i} ou t_i: custo do tráfego de rede no centro de dados d_i; ou seja, o valor a ser pago para transmitir 1 *byte* tendo como origem o centro de dados d_i.

Diante das variáveis apresentadas, são definidos os seguintes vetores de custos:

- Seja $\vec{v}$ o vetor de custos de execução de cada centro de dados em D:
  - $\vec{v} = \langle v_1, v_2, \ldots v_{|D|} \rangle$;
- Seja $\vec{t}$ o vetor de custos de transferência de tráfego cada centro de dados em D:
  - $\vec{t} = \langle t_1, t_2, \ldots t_{|D|} \rangle$.

A **variável dependente** é o custo. Seja $C(u_z, d_i, d_j)$ o custo para executar a VM da consulta u_z no centro de dados d_i e acessar os dados do centro de dados d_j; onde $1 \leq i, j \leq |D|$ e $1 \leq z \leq |U|$. O *custo para executar uma consulta é o custo total relativo à execução da VM (Ce_z) mais o custo de comunicação para transferência dos dados (Cd_z)*, tendo em conta a localização da VM e dos dados. Neste sentido, os custos são definidos como se segue:

I. **Custo para executar uma VM de consulta em um centro de dados que contém os dados de entrada:** o custo relativo à execução de uma VM u_z no centro de dados d_i que contém a fonte de dados necessária ao processamento é dado pela Equação 6.1.

$$\begin{aligned} C(u_z, d_i, d_i) &= C_i^z \\ &= (k_z \cdot R_z) \cdot v_i + T_z \cdot t_i \\ &= (k_z \cdot R_z) \cdot v_i + (R_z \cdot r_z) \cdot t_i \\ &= (k_z \cdot R_z) \cdot v_i + (R_z \cdot q_z \cdot k_z) \cdot t_i \\ &= k_z \cdot R_z \cdot (v_i + q_z \cdot t_i) \end{aligned} \tag{6.1}$$

II. **Custo para executar uma VM de consulta em um centro de dados e acessar dados armazenados em outro centro de dados:** o custo relativo à execução de uma VM u_z no centro de dados d_i e ler os dados do centro de dados d_j é dado pela Equação 6.2. Esse custo corresponde ao custo para transmissão dos dados entre os centros de dados ($R_z \cdot t_j$) acrescido do custo apresentado na Equação 6.1.

$$\begin{aligned} C(u_z, d_i, d_j) &= C_{i,j}^z = Ce_z + Cd_z \\ &= k_z \cdot R_z \cdot (v_i + q_z \cdot t_i) + R_z \cdot t_j \\ &= R_z \cdot [t_j + k_z \cdot (v_i + q_z \cdot t_i)] \end{aligned} \tag{6.2}$$

O resumo da função das variáveis que compõem o modelo e do esquema de atribuição de custos está ilustrado na Figura 6.1.

Para um grande volume de dados a ser processado por uma VM e que se encontra em centros de dados remotos, o tempo de transferência da fonte de dados poderia inviabilizar o serviço de DaaS. Mover dados pode implicar em custo de armazenamento, caso o processamento não seja em fluxo contínuo (*streaming: entra-processa-sai*). Se for necessário armazenar todo o conjunto de dados antes de processá-lo, isso irá aumentar o tempo de processamento, o que aumentaria também o custo da VM e poderia tornar inviável o processamento da consulta.

O escopo deste trabalho são as aplicações de fluxo contínuo, que não requerem armazenamento de grandes volumes de dados para iniciar o processamento dos mesmos.

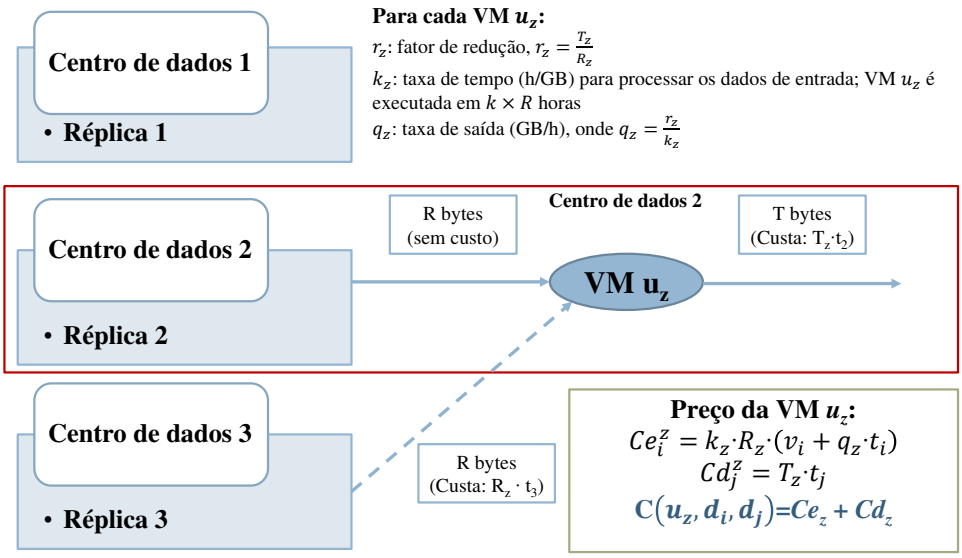


Figura 6.1: VM acessando uma fonte de dados armazenada no centro de dados B. Traduzido e adaptado de [Oliveira et al. 2015b]

6.2 Escalonamento baseado em Custo

Uma solução para redução de custos na execução das consultas de DaaS é empregar uma **estratégia de escalonamento de máquinas virtuais baseada em custo**. Nem sempre o custo para executar uma VM que realiza o processamento parcial ou integral de uma consulta no centro de dados que armazena a fonte de dados é menor do que se esses dados fossem recuperados via rede de comunicação.

Na estratégia de escalonamento baseada em custo, o centro de dados que apresenta os menores custos é o primeiro a receber máquinas virtuais. Quando sua capacidade de hospedar VMs satura, então o próximo centro de dados de menor custo passa a receber as novas VMs a serem escalonadas.

Se o custo de execução da VM somado ao custo de comunicação for inferior ao custo de execução da VM em um centro de dado que contenha uma réplica da fonte de dados necessária ao processamento da consulta, então essa consulta será alocada no centro de dados de menor custo de execução e os dados serão recebidos via rede.

A resposta do processo de escalonamento é um par odernado, que compreende o centro de dados que irá hospedar a máquina virtual e o centro de dados de onde a fonte de dados será recebida para o processamento da consulta. Os principais objetos que compõem a estratégia

de escalonamento baseada em custo são mostrados na Figura 6.2.

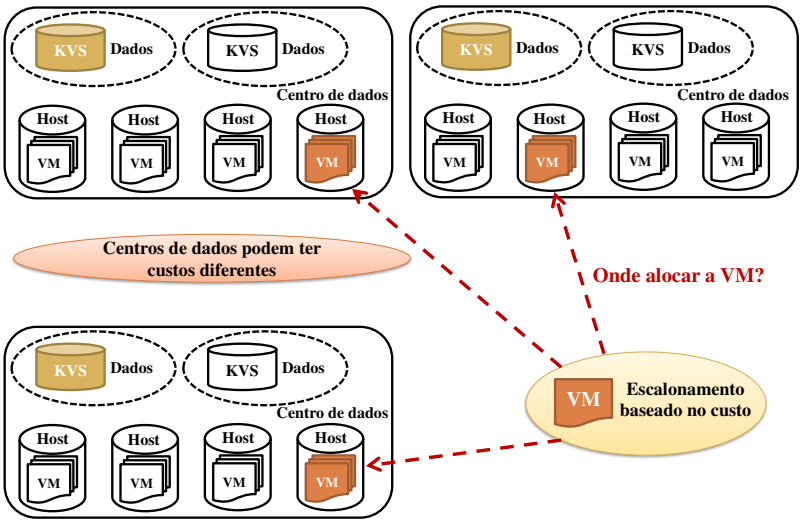


Figura 6.2: Estratégia de escalonamento baseada em custo.

6.2.1 Formalização do Problema

O problema da estratégia de escalonamento baseada em custo é minimizar o custo de alocar uma VM individualmente (escalonamento guloso), seguindo a ordem em que as VMs se encontram na fila de escalonamento.

Pode-se representar o custo mínimo para executar a VM u_z por:

$$\min C_z = \min C^z_{i,j} = \min C(u_z, s_z, d_i, d_j); \forall d_i, d_j \in D$$

onde C^z é um conjunto que contém todos os custos possíveis para o problema de escalonamento e pode ser decomposto em dois custos que, se minimizados de forma separada, a soma dos mesmos também representa o custo mínimo, como definido na Equação 6.3.

$$C^z = Ce^z + Cd^z \tag{6.3}$$

onde:

- Ce_z é o conjunto de custos de execução para as VMs, onde Ce^z_i representa o custo de executar a VM no centro de dados d_i:

  - $Ce_z = \{Ce_1^z, Ce_2^z, \ldots Ce_{|D|}^z\}$

- Cd_z é o conjunto de custos para recuperar a fonte de dados s_z tendo origem no centro de dados d_j:

  - $Cd_z = \{Cd_1^z, Cd_2^z, \ldots Cd_{|D|}^z\}$

Como os centros de dados de execução e para recuperação das fontes de dados podem ser diferentes, então não se pode realizar operações sobre vetores para o cálculo do custo final em uma única etapa. Portanto, torna-se necessário identificar cada centro de dados de modo separado.

Os custos para executar a VM u_z em cada centro de dados pode ser dado por uma multiplicação de vetores (álgebra linear), pois há correspondência entre os índices dos vetores em cada operação. Transformando o problema em operações sobre vetores, tem-se que:

$$\vec{Ce_z} = k_z \cdot (\vec{v} + q_z \cdot \vec{t}) \tag{6.4}$$

onde:

- k_z e q_z são constantes para a VM u_z.

O custo para transferir a fonte de dados s_z do centro de dados d_j para a VM u_z processar a consulta é dado por:

$$\vec{Cd_z} = R_z \cdot \vec{t} \tag{6.5}$$

onde:

- R_z é constante para a VM u_z.

A decisão do escalonador é baseada em encontrar o menor custo dentre as possíveis decisões de escalonamento. Portanto, essa decisão pode ser representada utilizando a função $argmin$ que está presente em todos os arcabouços para Programação Linear, conforme definido na Equação 6.6.

$$\{i, j\} = \underset{1 \leq i,j \leq |D|}{\operatorname{argmin}} \mathrm{C^z} \tag{6.6}$$

6.2.2 Formalização da Solução

O escalonamento baseado em custo é composto por dois custos distintos; portanto, o escalonamento será realizado em duas etapas. Na primeira etapa, faz-se um levantamento de qual centro de dados minimiza o custo de execução da VM. Na segunda etapa do escalonamento procura-se pelo centro de dados que contém uma cópia da fonte de dados necessária ao processamento da VM e que minimiza os custos de transferência desses dados.

Custos de Execução

Para calcular o vetor de custos de execução para uma VM, pode-se utilizar operações usando vetores diretamente, mas também é possível calcular o custo de execução de várias VMs utilizando operações sobre matrizes. Ferramentas para Programação Linear permitem realizar otimizações nesses cálculos, como as seguintes ferramentas livres e de código aberto: Numpy [van der Walt et al. 2011] e SciPy [Jones et al. 2015] que são desenvolvidas em Python, R [R Core Team 2015] e Octave [Eaton et al. 2008].

Primeiramente, define-se uma **matriz de coeficientes** θ^T. Este modelo em particular tem dois coeficientes relativos às VMs que serão multiplicados pelas variáveis de custo dos centros de dados; ou seja, o número de produtos da função de custo de execução. A matriz de coeficientes θ e a sua matriz transposta θ^T são definidas pela Equação 6.7.

$$\theta = \begin{bmatrix} k_1 & k_2 & \dots & k_{|}U| \\ k_1 \cdot q_1 & k_2 \cdot q_2 & \dots & k_{|U|} \cdot q_{|U|} \end{bmatrix}_{2 \cdot |U|} \iff \theta^T = \begin{bmatrix} k_1 & k_1 \cdot q_1 \\ k_2 & k_2 \cdot q_2 \\ \dots & \dots \\ k_{|U|} & k_{|}U| \cdot q_{|U|} \end{bmatrix}_{|U| \cdot 2} \tag{6.7}$$

Pode-se manter o mesmo sistema linear multiplicando uma linha por um escalar. No caso da matrix θ^T, pode-se simplificar a matriz multiplicando cada linha por $\frac{1}{k_z}; 1 \leq z \leq |U|$, mantendo o sistema linear equivalente. Como k é uma constante para cada VM, variando apenas os valores v e t, pode-se executar o algortimo de minimização do escalonamento sem considerar esse valor.

Em segundo lugar, define-se a **matriz de custos dos centros de dados** X. A matriz é composta pelos vetores de custos de execução e comuninicação dos centros de dados, que

são requeridos para calcular o custo total de execução da VM. A matriz X é definida na Equação 6.8.

$$X = \begin{bmatrix} \vec{v} \\ \vec{t} \end{bmatrix} = \begin{bmatrix} v_1 & v_2 & \dots & v_{|D|} \\ t_1 & t_2 & \dots & t_{|D|} \end{bmatrix}_{2 \cdot |D|} \tag{6.8}$$

A matriz de custos, C, contém os custos de execução de cada VM $u_z \in U$ em cada um dos centros de dados $d_i \in D$ e está representada na Equação 6.9.

$$C = \theta^T \cdot X \tag{6.9}$$

$$C = \begin{bmatrix} 1 & q_1 \\ 1 & q_2 \\ \dots & \dots \\ 1 & q_{|U|} \end{bmatrix}_{|U| \cdot 2} \cdot \begin{bmatrix} v_1 & v_2 & \dots & v_{|D|} \\ t_1 & t_2 & \dots & t_{|D|} \end{bmatrix}_{2 \cdot |D|} =$$

$$\begin{bmatrix} v_1 + q_1 \cdot t_1 & v_2 + q_1 \cdot t_2 & \dots & v_{|D|} + q_1 \cdot t_{|D|} \\ v_1 + q_2 \cdot t_1 & v_2 + q_2 \cdot t_2 & \dots & v_{|D|} + q_2 \cdot t_{|D|} \\ \dots & \dots & \dots & \dots \\ v_1 + q_{|U|} \cdot t_1 & v_2 + q_1 \cdot t_2 & \dots & v_{|D|} + q_1 \cdot t_{|D|} \end{bmatrix}_{|U| \cdot |D|}$$

Custos de Acesso aos Dados

Seja M a **matriz de fontes de dados** necessárias ao processamento das consultas nas VMs. A matriz M possui $|U|$ linhas e $|D|$ colunas. As linhas representam as VMs e as colunas representam os centros de dados. Se o centro de dados d_i armazena uma cópia dos dados necessários à VM u_z, então o valor m_{zi} da matriz M, que faz a interseção entre a linha z e a coluna i é definido como 1; em caso contrário, é 0. O vetor $\vec{M^z}$ corresponde à z-ésima linha de M e descreve se os centros de dados contêm uma cópia dos dados necessários à consulta realizada pela VM u_z.

$$M_{|U|\cdot|D|} = \begin{bmatrix} 1 & 0 & \dots & 1 \\ 1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{6.10}$$

O custo t_j^z representa o custo de comunicação para recuperar uma cópia da fonte de dados s_z a partir do centro de dados d_j. Para fins de escalonamento, o objetivo consiste em minimizar esse custo dentre as possíveis opções de transferência de dados.

Para minimizar o custo de acesso aos dados, $\forall u_z \in U$, primeiro prepara-se o vetor de custos de transferência de dados $\vec{t^z}$ dividindo elemento a elemento o vetor de custos de comunicação $\vec{t}$ pelo vetor $\vec{M^z}$ correspondente da matriz de fontes de dados. Quando os dados não estão armazenados em d_j, então ocorre uma divisão por zero. O resultado da divisão por zero é um valor que tende a infinito, logo o custo assume o valor infinito (∞) e não há a possibilidade do escalonador escolher esse centro de dados.

Seja d_i o centro de dados que minimiza o custo de execução da VM u_z. Antes de escolher o custo mínimo dentre os valores do vetor $\vec{t^z}$, deve-se zerar o custo de acesso t_i^z, que é o custo de tranferir a fonte de dados de d_i para d_i. Por definição, deve haver ao menos uma cópia da fonte de dados disponível dentre os centros de dados.

Neste ponto, o vetor de custos de acesso à fonte de dados para a VM u_z está pronto e pode ser buscado o custo mínimo dentro desse vetor.

6.2.3 Algoritmo de Escalonamento

A decisão proposta pelo algoritmo de escalonamento baseado em custo é o par ordenado {d_i, d_j} contendo os centros de dados que minimizam a função de custo sobre todos os centros de dados, onde d_i é o centro de dados que minimiza o custo de execução da VM e d_j é o centro de dados de onde serão acessados os dados de entrada para o processamento da VM. Essa decisão está formalizada nas duas últimas seções e sumarizada na Equação 6.11. Note que o mecanismo de escalonamento pode propor um conjunto de resultados vazio. Este caso denota que a fonte de dados necessária ao processamento da VM não está disponível; portanto, a consulta não pode ser executada.

Uma instância de VM só pode baixar uma fonte de dados tendo como origem os centros

$$schedule(u_z, s_z, D) = \begin{cases} \{d_i, d_j\} = \underset{d_i, d_j \in D}{\operatorname{argmin}} \mathrm{C}(u_z, s_z, d_i, d_j) & \text{if } C(u_z, s_z, d_i, d_j) \neq \infty \\ \emptyset & \text{caso contrário} \end{cases} \tag{6.11}$$

de dados que as armazenam. Para representar essa premissa matematicamente, foi definida uma função chamada *contains* que verifica se é possível recuperar uma fonte de dados a partir de um determinado centro de dados. Se o centro de dados contiver a fonte de dados, a função *contains* retornará o valor 1; caso contrário, ela retorna 0. A função é definida na Equação 6.12.

$$contains(d_j, s_z) = \begin{cases} 1 & \text{se } s_z \text{ está armazenada em } d_j \\ 0 & \text{se } s_z \text{ não está armazenada em } d_j \end{cases} \tag{6.12}$$

O custo para escalonar a VM u_z no centro de dados e ler uma fonte de dados localizada no mesmo centro de dados é dada pela Equação 6.1, considerando que não há custo de comunicação para transferência de dados entre os diferentes centros de dados; e pela Equação 6.2; caso contrário. Se o centro de dados não é uma alternativa possível, então o custo de escalonamento é atribuído como infinito, não sendo, portanto, uma decisão passível de ser tomada pelo escalonador. A Equação 6.13 resume os valores atribuídos aos custos em cada uma dessas condições.

$$C^z_{i,j} = \begin{cases} k_z \cdot R_z \cdot (v_i + q_z \cdot t_i) & \text{if } d_i = d_j \wedge contains(d_j, s_z) = 1 \\ R_z \cdot [t_j + k_z \cdot (v_i + q_z \cdot t_i)] & \text{if } d_i \neq d_j \wedge contains(d_j, s_z) = 1 \\ \infty & \text{caso contrário} \end{cases} \tag{6.13}$$

Em resumo, o Algoritmo 6.1 apresenta de forma genérica o escalonamento baseado em custo da máquina virtual u_z, que utiliza a fonte de dados s_z como entrada para o processamento da consulta. O algoritmo recebe como entrada os vetores contendo todos os custos de execução da VM ($\vec{Ce_z}$) e outro vetor contendo os custos de recebimento dos dados ($\vec{Cd_z}$).

O centro de dados escolhido para executar a VM é o que minimiza o custo de execução. Neste caso, o centro de dados d_i (Linha 2). Se houver a fonte de dados requerida para o

processamento da VM (s_z) estiver replicada no centro de dados escolhido para a execução da VM (d_i), então o custo de recebimento dos dados de entrada não se faz necessário. Portanto, esse custo é alterado para zero (Linha 4). Quando não há uma fonte de dados, o custo de transferência é infinito. A cópia da fonte de dados será recebida do centro de dados que minimiza o custo de transfência. Neste caso, o centro de dados é o d_j (Linha 6). Essa letra foi utilizada para facilitar o entendimento, mas i pode ser igual a j, pois há a possibilidade de um mesmo centro de dados minimizar ambos os custos.

Algoritmo 6.1: Algoritmo de escalonamento baseado em custo para a VM u_z.

Entrada: u_z, s_z, $\vec{Ce}_z$, $\vec{Cd}_z$, D

Saída: $\{d_i, d_j\}$

```
1 início
2     d_i ⟵ argmin_{d_i∈D} Ce_z;
3     se (contains(d_i, s_z) == 1) então
4         t_i ⟵ 0;
5     d_j ⟵ argmin_{d_j∈D} Cd_z;
```

Resultado: $\{d_i, d_j\}$

6.3 Estudos Preliminares

A estratégia de escalonamento baseada em custo foi comparada à estratégia de escalonamento por alternância circular em um trabalho anterior [Oliveira et al. 2015b]. Os custos finais obtidos para as consultas que foram realizadas utilizando a estratégia de escalonamento baseada em custo foram, em média, pelo menos duas vezes menores do que os da abordagem tradicional utilizando alternância circular.

6.3.1 Escalonamento por Alternância Circular

A estratégia de escalonamento por **alternância circular**, também chamada de *round robin*, seleciona o primeiro par de centros de dados que satisfaça as condições para executar a VM, realizando a busca dos centros de dados em uma lista. Ao atingir o final da lista de centros de dados, a busca retorna para o primeiro da fila novamente. Não há garantias sobre a ordem dos centros de dados na lista, podendo estar aleatoriamente ordenados. Um exemplo de

escalonamento de VMs que utiliza essa estratégia é o utilizado pelo Openstack [Openstack 2015].

6.3.2 Simulação

O escalonamento baseado em custo foi avaliado por meio de simulação utilizando as ferramentas CloudSim (desenvolvida em linguagem Java) [Buyya et al. 2009] e dois arcabouços para programação linear em Python, o Numpy [NumPy 2015] e o SciPy [SciPy 2015].

Custo Ótimo

Os centros de dados com os recursos disponíveis para executar uma VM variam de acordo com o número de máquinas virtuais que já estão alocadas a eles. Para avaliar o impacto que a alocação traz ao conjunto de centros de dados no início e ao longo do provimento dos serviços, os **custos ótimos** são estimados com base na disponibilidade do conjunto inicial de centro de dados; ou seja, os **custos ótimos ideais** são os custos que deveriam ser cobrados se todos os centros de dados tivessem capacidade disponível para hospedar a VM a ser escalonada. Os **custos ótimos reais** são os custos obtidos pela estratégia de escalonamento baseada em custo, que sempre encontra os centros de dados de menor custo que estão disponíveis.

Seja $C^*(u_z, s_z, D)$, ou simplesmente C^{*z}_D, o custo ótimo para executar a VM u_z acessando a fonte de dados s_z, dado o conjunto de centros de dados D. O **custo ótimo** para uma máquina virtual é o custo obtido quando o par de centros de dados que minimiza o custo é escolhido sobre o conjunto inicial de centros de dados $|D|$ (considerando todos os centros de dados como disponíveis para executar a VM), conforme definido na Equação 6.14.

Isso possibilita uma análise de como os custos influenciam a utilização dos centros de dados e auxilia uma adaptação dos custos de modo a realizar balanceamento de carga. Vale ressaltar que a estratégia de escalonamento baseada em custo sempre decide pela alocação ótima (possível) dentre os centros de dados disponíveis.

$$\begin{aligned} C^*(u_z, s_z, D) &= \min C_z \\ &= \min C^z_{i,j} \\ &= \min C(u_z, s_z, d_i, d_j); \\ & \quad u_z \in U, s_z \in S, \forall d_i, d_j \in D \end{aligned} \tag{6.14}$$

Métrica: Custo Normalizado

A métrica analisada foi o **custo normalizado**, que é a razão entre o custo obtido com a abordagem de escalonamento e o *custo ótimo ideal*. O custo normalizado $NC(z)$ é formalmente definido na Equação 6.15.

$$\begin{aligned} NC(z) &= \frac{C^z_{i,j}}{C^{*z}_{D}}; \\ & \quad d_i \wedge d_j \text{ foram as escolhas do escalonador} \end{aligned} \tag{6.15}$$

onde:

- $C^{*z}_{D} = \min C^z_{i,j}; \forall d_i, d_j \in D$.

6.3.3 Resultados Preliminares

Os valores discrepantes e a variação presentes nos resultados de duas estratégias de escalonamento foram representados por meio de gráficos caixas (*boxplots*). Dois cenários foram analisados: (i) quando todas VMs poderiam ser alocadas em um único centro de dados; (ii) quando no máximo 20% das VMs poderiam ser hospedadas em um mesmo centro de dados, conforme Tabela 6.1.

Tabela 6.1: Tratamentos avaliados.

#Tratamento	Parâmetro	Nível do Fator
1	Número máximo de VMs por DC	8
2	Número máximo de VMs por DC	40

Assumiu-se que:

1. Uma instância de VM executa uma consulta;
2. Cada consulta lê dados a partir de uma fonte de dados;
3. Todas as VMs requerem a mesma capacidade computacional;
4. Todos os centros de dados têm capacidade de hospedar o mesmo número de VMs.

Os custos v e t de cada centro de dados foram atribuídos aleatoriamente de acordo com a distribuição uniforme dentro de uma faixa de valores. Para cada consulta, as variáveis q e k também foram atribuídas aleatoriamente. Cada fonte de dados foi replicada utilizando o fator de replicação igual a 3 e distribuídas aleatoriamente dentre os centros de dados. O número de centros de dados, de consultas e de fontes de dados são constantes. Essas variáveis dependentes estão detalhadas na Tabela 6.2. Os custos foram baseados nos custos utilizados pela Amazon Elastic Cloud Computing EC2.

Tabela 6.2: Configuração do sistema.

#		**Configuração**
1	$\vec{v}$	Valores aleatórios na faixa $[0.065, 3.41[$
2	$\vec{t}$	Valores aleatórios na faixa $[0.015, 0.51[$
3	$\vec{q}$	Valores aleatórios na faixa $[0.3, 2.1[$
4	$\vec{k}$	Valores aleatórios na faixa $[0.28, 2.5[$
5	$\|D\| = \|U\| = \|S\|$	40
6	Fator de replicação	3 (aleatoriamente posicionado entre os DCs)

Valores extremos para os atributos das VMs contribuem para aumentar a variação nos valores de custos. Pela Figura 6.3, pode-se identificar uma linha dentro da caixa indicando o valor da mediana. No experimento anterior, a média e a mediana possuem praticamente o mesmo valor. As linhas que delimitam a parte superior e a inferior da caixa são os 25%-quantis. O quantil inferior indica que 25% dos resultados estão abaixo da linha inferior e o quantil superior indica que 75% dos resultados estão acima dessa marca. A mediana para o custo normalizado da estratégia de escalonamento baseada em custo é próxima a 1 (valor ótimo) e cerca de 3 para o escalonamento por alternância circular.

Resultados adicionais podem ser encontrados em Oliveira *et al.* [Oliveira et al. 2015b].

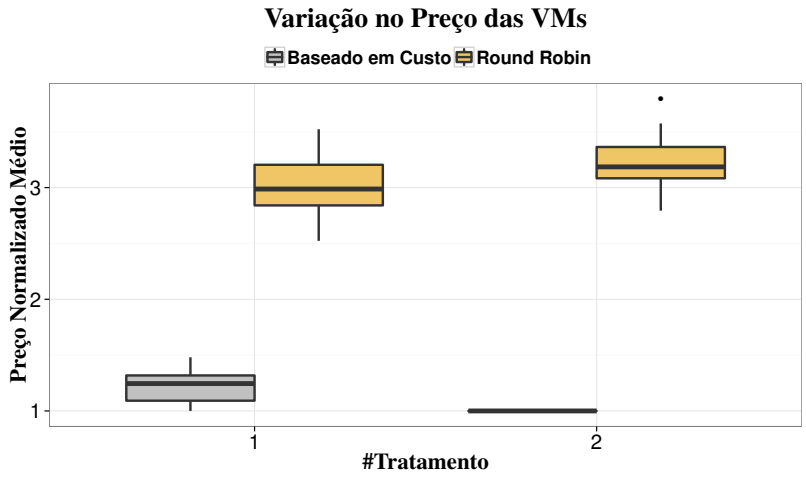


Figura 6.3: Gráfico de caixa (*boxplot*) para avaliação variação dos custos comparando as estratégias de escalonamento [Oliveira et al. 2015b].

6.4 Conclusões

Dados-como-Serviço (*Data-as-a-Service* – DaaS) é um modelo de negócio para serviços de computação em nuvem que fornece suporte para "consultar a Web". A escala do volume de dados disponível na Web é alta. Portanto, é importante estabelecer políticas para os custos de recursos que priorizem o cumprimento dos requisitos de SLA, minimizando a incidência de violações de SLA, bem como otimizar a utilização de recursos e o custo-benefício dos serviços.

Neste capítulo foi proposto e analisado um modelo de tarifação para DaaS. Em seguida, este modelo foi implementado em um mecanismo de escalonamento de máquinas virtuais baseado em custo.

O modelo de escalonamento baseado em custo proposto pode também apoiar o balanceamento de carga e a escalabilidade dos serviços, quando combinado com uma estratégia de custos adaptativa que prioriza a alocação de VMs em centros de dados subutilizados, de baixo custo e/ou que irão minimizar o número de violações de SLAs e suas devidas penalidades.

O trabalho apresentado neste capítulo está tecnicamente alinhado com o projeto LEADS financiado pela Comissão Europeia sob o Programa *Seventh Framework* (concessão no. 318.809). Este trabalho também foi apoiado pelo projeto TruEGrid financiado pela Co-

ordenação Brasileira para Aprimoramento de Pessoal de Nível Superior (CAPES), o Serviço Alemão de Intercâmbio Acadêmico (DAAD) e a Sociedade Alemã de Cooperação Internacional (GIZ) sob o Programa Novas Parcerias (NOPA) e financiado pelo Programa Ciências sem Fronteiras (CsF).

Capítulo 7

Modelo para Tarifação Confiável em Computação em Nuvem

> "É melhor perder-se o bom por querer o melhor."
>
> Shakespeare

Este capítulo reúne todos os trabalhos apresentados nos capítulos anteriores. Eles atuam em conjunto para dar suporte a um modelo de tarifação confiável, que é baseado em um processo de transparência no desempenho dos serviços de computação em nuvem e em estimativas de perdas financeiras causadas por tráfego anômalo que, quando identificadas, são acardas pelo provedor desse serviço.

O modelo para tarifação confiável em sistemas de computação em nuvem é uma solução que envolve diversos níveis organizacionais, podendo ser avaliado sobre diversas perspectivas. Este trabalho focará na contribuição que as técnicas de detecção de anomalias trazem para o negócio.

A Figura 7.1 retoma o conceito de monitoramento compartilhado entre as partes provedor e cliente, abstraindo detalhes técnicos, mas destacando o ponto da fatura poder ser inclusive estimada durante a execução dos serviços ou contestada, caso necessário. Por outro lado, a parte provedora pode planejar investimentos e alocação de recursos, também com base nos resultados obtidos do processo de monitoramento.

Para avaliar o impacto que o custo de anomalias de tráfego acarreta para sistemas de

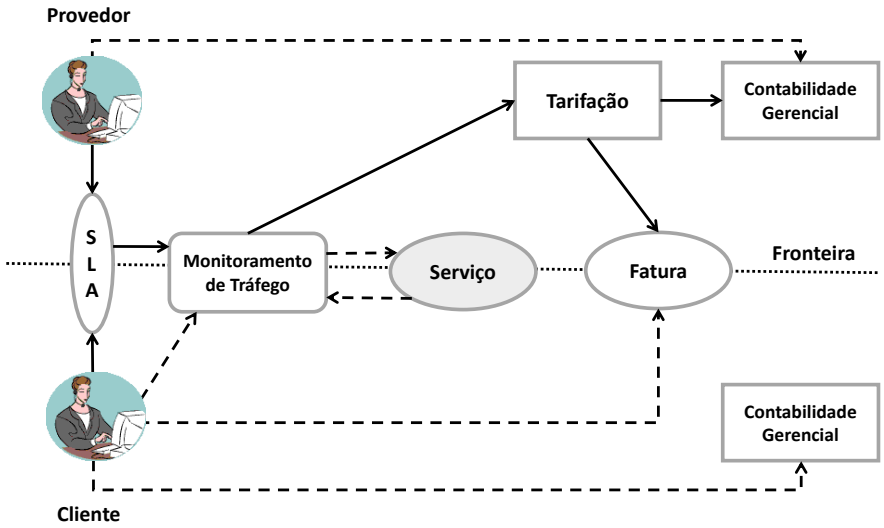


Figura 7.1: Gerenciamento proposto para execução de serviços.

computação em nuvem, o modelo de negócio de DaaS será utilizado como estudo de caso. Neste caso, o provedor de DaaS contrata um provedor para fornecer infraestrutura e vende o serviço de DaaS a outros clientes, como mostrado na Figura 7.2.

Há o conceito de custo e de preço. O preço é o valor que é cobrado ao cliente por um serviço e o custo é o valor que o provimento do serviço acarreta ao provedor [Werner e Jones 2003]. Em sistemas de computação em nuvem, como esses papéis podem alternar, esses valores podem ser vistos de maneiras diferentes para a mesma entidade (provedor ou cliente).

Para simplificar, neste capítulo será utilizado apenas o termo **custo**, não tratando de margens de lucro. Para o caso do provimento de DaaS, há duas possibilidades. Ou cliente pode pagar de modo variável, se conseguir a oportunidade de executar seus serviços nos centros de dados de menor custo, ou pode pagar um valor fixo referente aos centros de dados de maior custo e a diferença entre esses custos é revertida ao provedor. Essas são decisões de negócio. Este trabalho flexibiliza esse tipo de decisão, concentrando-se em reduzir os custos e penalidades em sua concepção primária.

Um implementação prática do modelo de tarifação confiável em serviços de computação em nuvem será apresentada com foco no mecanismo de escalonamento do provedor de DaaS,

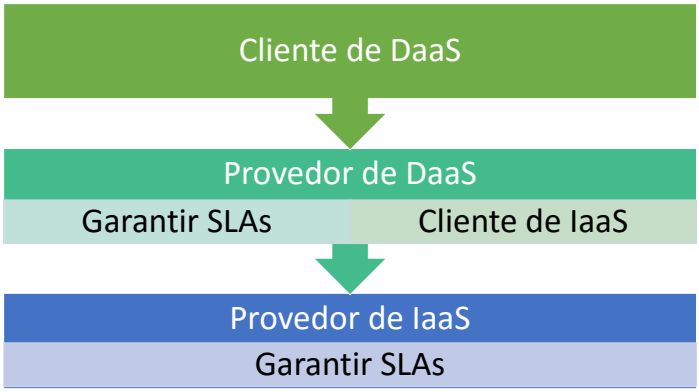


Figura 7.2: Exemplo de contratação de serviços de computação em nuvem em diferentes níveis de abstração.

que também é cliente de IaaS. Portanto, é possível dimensionar custos com a compra de serviços, lucro com a venda de serviços, perda financeira devido à incidência de penalidades em decorrência de anomalias de tráfego, mas também ganho advindos do provedor de IaaS, quando essas anomalias ocorrem.

O provedor de DaaS acompanhará os resultados do processo de monitoramento de anomalias e fará o escalonamento dos recursos de modo a reduzir prejuízos financeiros. Para propor quais os centros de dados receberão VMs, ou de qual DC os dados serão recebidos, escalonador recebe informações de quais VMs devem ser escalonadas e quais seus respectivos SLAs. O detector de anomalias provê dados sobre o estado de cada centro de dados e caso sejam encontrados desvios, há um módulo que realiza o cálculo das penalidades devidas. Considerando que será aplicado o mecanismo de escalonamento baseado em custo apresentado no Capítulo 6, dois outros módulos ainda são empregados no escalonamento: o calculador de custos e o atualizador de custos. O calculador de custos verifica o custo para executar cada VM, de modo que o escalonador priorizará os centros de dados de menor custo. Quando ocorrem anomalias, o centro de dados em falha tem seu custo atualizado de modo a envolver os custos das anomalias. Esse processo está apresentado na Figura 7.3.

Antes de realizar a validação do modelo de tarifação confiável proposto, é necessário definir as variáveis que compõem o escalonamento baseado em custo para DaaS definido na Seção 6.1. Os custos de processamento (v) e de transporte dos dados (t) nos DCs, os tipos de SLAs, as propriedades das VMs (k, q), os centros de dados (D), os perfis de tráfego das

Figura 7.3: Interação entre o escalonador e os demais módulos do modelo.

VMs (R, T) e a forma de injeção de anomalias (Equação 7.1). Para tal, foram realizadas as seguintes fases:

I. Definição inicial de custos: variáveis v, t e custo do serviço de IaaS;

II. Definição de SLAs para cada VM;

III. Experimento com uma aplicação de DaaS real para calibragem de variáveis do modelo (variáveis k, q, R, T, $tempo$ e o perfil do tráfego);

IV. Injeção de anomalias de tráfego;

V. Estudo de caso para avaliar o prejuizo causado por anomalias de tráfego em cada tipo de serviço.

7.1 Definição Inicial de Custos para os Serviços

O modelo de tarifação confiável utilizará os custos para IaaS aplicados pela empresa Cloud & Heat [Cloud&Heat 2015] para configuração do estudo de caso. Outros custos podem ser empregados sem perda de generalidade.

As instâncias que serão analisadas são as de porte médio (M), que possuem 4 CPUs virtuais, 4 GB de memória RAM, 120 GB de disco rígido virtual, 4 TB de tráfego, sendo cobrado €$0,08$ por hora de cada instância de VM e €$0,02$ a cada GB que excede essa quota de rede, contudo, esse excedente de tráfego não será cobrado nesta avaliação para simplicá-la.

Três centros de dados comporão a avaliação do modelo para tarifação confiável. Os custos v e t para DaaS nesses centros de dados e os custos de IaaS pagos para executar uma VM nesses centros de dados a durante $1h$ estão contidos na Tabela 7.1.

Tabela 7.1: Custo por centro de dados.

ID DC	**DaaS v** (€)	**DaaS t** (€)	**IaaS** ($\frac{€}{h}$)
1	0,09	0,02	0,08
2	0,1	0,03	0,08
3	0,07	0,05	0,08

7.2 Definição dos SLAs

Três tipos de SLAs foram definidos para a métrica disponibilidade, que são chamados de *Golden*, *Silver*, *Bronze*. Essa nomenclatura baseou-se na terminologia utilizada por Li *et al.* [Li et al. 2012]. Os níveis atribuídos para cada tipo de SLA foram estabelecidos após entrevista com um gerente responsável pelos serviços de Internet de um provedor de telecomunicações. Os três tipos são definidos abaixo:

I. **Ouro (*Golden*):** equivale a $99,9\%$ de disponibilidade. O custo adicional para esse tipo de SLA é de 20% do valor original. O tempo máximo que o serviço pode estar indisponível é de $3,6$ segundos em 1 hora.

II. **Prata (*Silver*):** equivale a $99,8059\%$ de disponibilidade. O custo adicional para esse tipo de SLA é de 10% do valor original. O tempo máximo que o serviço pode estar indisponível é de $6,98$ segundos em 1 hora.

III. **Bronze**[1]**:** equivale a $99,589\%$ de disponibilidade. O custo adicional para esse tipo de SLA é de 5% do valor original. O tempo máximo que o serviço pode estar indisponível é de $14,796$ segundos em 1 hora.

Para a VM 1, o tipo de acordo estabelecido foi o *Golden*. O tipo de SLA da VM 2 é *Silver* e o da VM 3 é *Bronze*. Na Tabela 7.2 estão reunidas as informações sobre os tipos de SLAs e as VMs correspondentes.

Tabela 7.2: SLAs Contratados.

ID VM	**Disponibilidade Mínima (%)**	**Tipo**	**Custo Adicional (%)**	**Máx. Tempo (s/h) Indisponível(s/h)**
1	$99,9\%$	Golden	20%	3,6
2	$99,8059\%$	Silver	10%	6,98
3	$99,589\%$	Bronze	5%	14,796

7.3 Experimento para Definição de Variáveis

As variáveis que compõem o modelo de custo de DaaS são difíceis de ser obtidas. Para avaliar o modelo, realizou-se um experimento com uma aplicação que permitiu que essas variáveis pudessem ser estimadas. O experimento consistiu em uma aplicação de *crawling* (rastreamento) da Web utilizando o modelo de programação MapReduce para o processamento dos dados. O arcabouço para desenvolvimento de *software* foi o Hadoop [Hadoop].

Os *softwares* mapeadores e redutores foram distribuídos em em três micronuvens localizadas em 3 cidades alemãs, são elas Dresden, Münster e Hamburg (vide Figura 7.4. O tempo de execução desse experimento foi de 24 horas. As VMs permaneceram ativas durante todo o período do experimento.

[1]O SLA do tipo bronze também pode ser encontrado na literatura como o nível que não oferece garantias para o serviço (*best effort*, ou melhor esforço) [Li et al. 2012]. Contudo, este trabalho está considerando níveis diferentes reais prestados por um ISP que oferece serviços de voz e dados em todo Brasil e em vários outros países

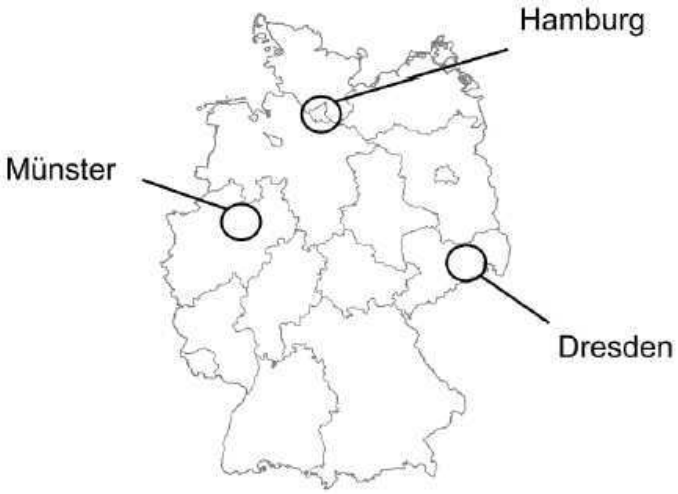


Figura 7.4: *Crawling* distribuído [Quoc et al. 2015].

7.3.1 Caracterização do Tráfego de Rede

Durante a execução do experimento foram coletadas métricas de utilização de memória, CPU e de rede. O tráfego de rede foi coletado usando a ferramenta *tcpdump*. O tráfego que foi recebido e enviado de cada VM foi processado e representado graficamente por meio de séries temporais da vazão de cada VM. Estatísticas sobre o volume de tráfego das máquinas virtuais em Mbps estão descritas no Apêndice F na Tabela F.1 e gráficos complementares contendo a vazão em Megabits por segundo (Mbps) estão mostrados na Figura F.1.

Na Figura 7.5 o volume de tráfego de entrada e de saída foi agrupado por hora para fornecer informações mais compreensíveis sobre a execução das máquinas virtuais. A VM #1 apresenta um volume de tráfego de saída superior ao de entrada inicialmente. Após a 10ª hora eles se tornam equivales. Esse também é o momento em que ambas as direções do tráfego tem seu horário de pico, com cerca de 20 GB para o volume de entrada e 30 GB para o volume de saída. A VM # 2 apresenta um volume de entrada maior do que o de saída durante praticamente toda a sua execução. Com volume máximo de entrada de quase 20 GB na 10ª hora e 8,65 GB na hora 17. A VM # 3 possui volumes equiparáveis para o tráfego de entrada e saída. O volume máximo de entrada foi de 17,41 GB na 8ª hora de 17,82 GB na 10ª.

Todos os valores de volume de tráfego de entrada e saída amostrados por hora encontram-se no Apêndice F. Nas Tabelas F.2, F.3 e F.4 estão apresentadas as variáveis calculadas por

hora para as VMs 1, 2 e 3, respectivamente.

Essas informações serão utilizadas para estimar as variáveis q e k de cada VM por hora na Seção 7.3.2. Uma análise complementar envolvendo essas variáveis será realizada na Seção 7.7. Para facilitar o entendimento com relação ao período do experimento, o eixo que identifica o tempo está representado pela unidade *hora* (h), contudo, os dados apresentados foram amostrados em segundos.

7.3.2 Estimativa de Variáveis do Modelo

As variáveis q, k, R e T das máquinas virtuais foram estimadas utilizando os dados obtidos pelo processamento do tráfego. Essas variáveis foram contabilizadas para o tempo de execução total do experimento (de aproximadamente $24,79$ horas) e também a cada hora de execução. O volume foi medido em GiB[2], mas ao longo deste trabalho essa medida será tratada simplesmente por GB. Os valores-base dessas estimativas estão disponíveis no Apêndice F. Na Tabela F.5 estão apresentadas as variáveis calculadas pelo tempo total do experimento e nas Tabelas F.2, F.3 e F.4 estão apresentadas as variáveis calculadas por hora para as VMs 1, 2 e 3, respectivamente.

A Figuras 7.6a, 7.6b, 7.7a e 7.7b apresentam as estimativas calculadas por hora para as variáveis q, k, R e T, respectivamente.

7.4 Definição de Penalidades

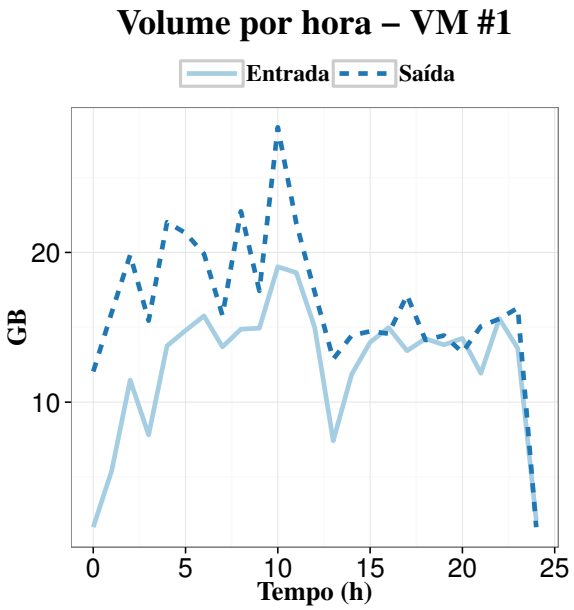
Esta seção aborda uma forma de estimar penalidades para a incidência de anomalias de tráfego de rede no modelo de DaaS e no modelo de IaaS. Essas estimativas podem variar, conforme o tipo de negócio e contratos envolvidos. Todavia, independemente das estimativas, é possível obter uma visão dos ganhos obtidos por meio do modelo de tarifação confiável proposto. As anomalias detectadas serão penalizadas de acordo com o prejuízo financeiro que acarretam para o modelo de negócio.

[2]Para compatibilizar a leitura, nesta tese:

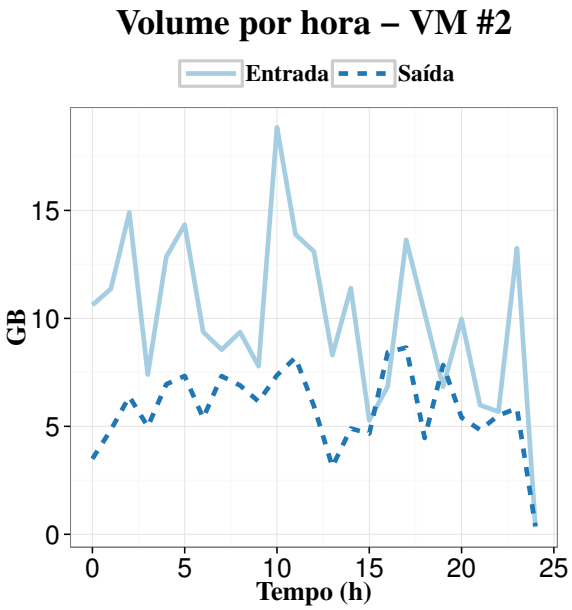
1 GB (gigabyte) = 1.000 MB (megabytes) = 10^9 bytes

1 GiB (gibibyte) = 1.024 MiB (mebibytes) = 2^{30} bytes

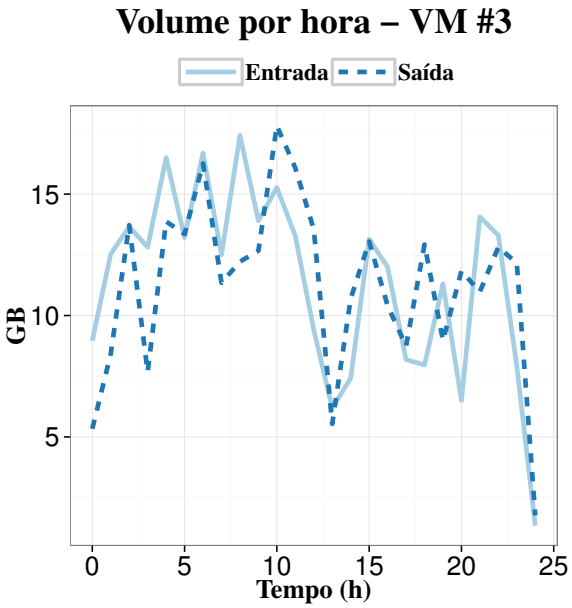
Ambas as notações são encontradas na literatura.



(a) VM #1

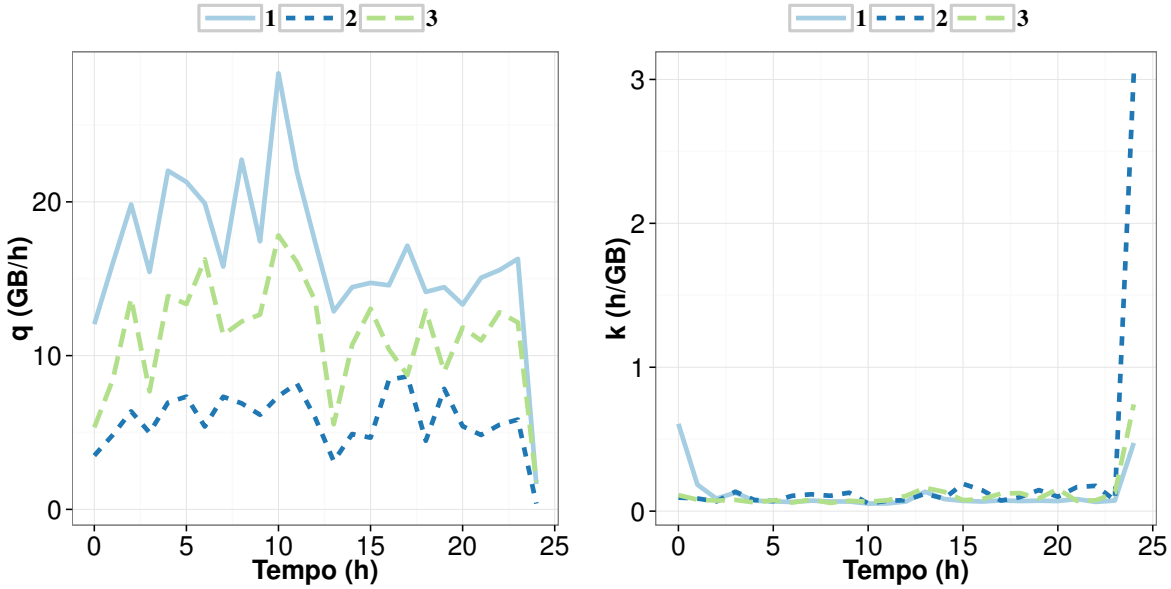


(b) VM #2



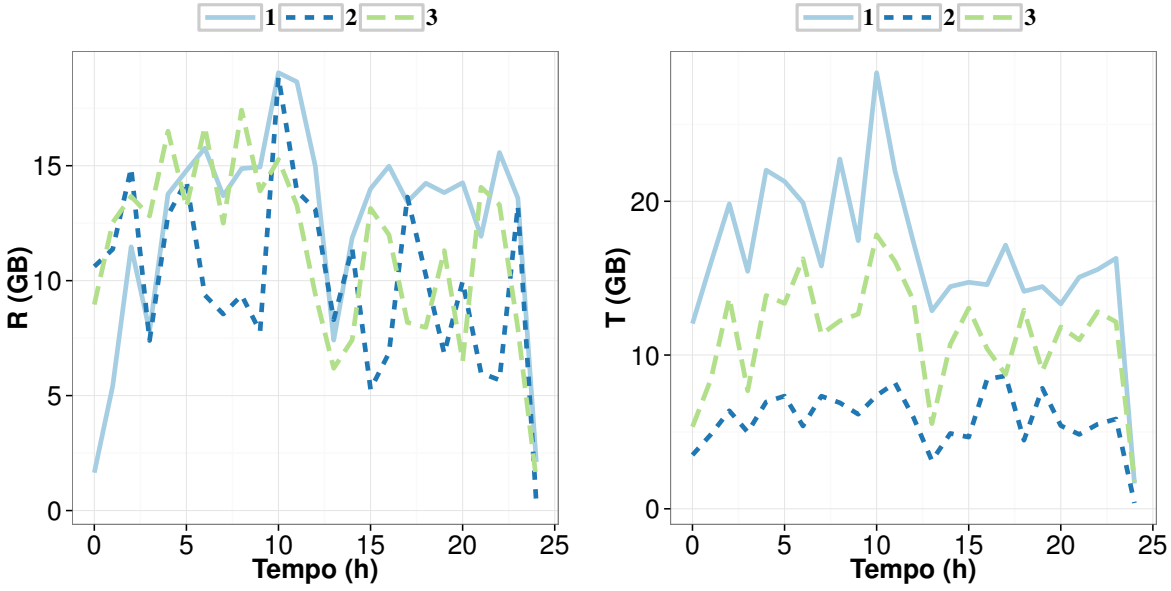
(c) VM #3

Figura 7.5: Vazão de entrada e saída medida por hora para cada uma das 3 máquinas virtuais.



(a) q (b) k

Figura 7.6: Estimativa das variáveis de custo calculadas por hora para cada VM.



(a) R (b) T

Figura 7.7: Estimativa das variáveis de volume de dados calculadas por hora para cada VM.

As penalidades serão estimadas por um determinado período de provimento do serviço. Cada penalidade representa o prejuízo financeiro que essas anomalias representam. As penalidades retornam ao cliente como crédito e representam um acréscimo nos custos do provedor. O modelo de penalidades para as anomalias detectadas deve ser definido com base no modelo de negócio dos serviços. Com base em quanto realmente custa aquele tráfego.

Nesta avaliação serão calculados por quantos segundos o serviço esteve indisponível. Por questões de simplicação, o número de segundos em que o tráfego apresentou anomalias será chamado de número de anomalias (# anomalias). O número de anomalias acima do limite máximo aceitável será chamado de número de anomalias excedentes.

Para o caso de DaaS, as penalidades serão definidas com base no modelo proposto por Xiong *et al.* [Xiong et al. 2011b], que atribui uma penalidade fixa a cada anomalia detectada. Esse valor fixo é uma função do tipo de SLA da VM que está sujeita à anomalia.

Se houver uma anomalia em um centro de dados que deve executar a VM u_z, o valor de penalidade correspondente ao SLA da VM é $p_{SLA}(z)$ e será acrescido ao custo de execução daquele centro de dados ($v_i = v_i + p_{SLA}(z)$), porque o provedor credita esse valor ao cliente. Se ocorrer uma anomalia no centro de dados de onde a fonte de dados seria recuperada, então o custo de recuperação da fonte de dados também é acrescido com o valor de penalidade correspondente ao SLA da VM ($t_j = t_j + p_{SLA}(z)$). Os valores das penalidades de cada anomalia estão apresentados na Tabela 7.3.

Tabela 7.3: Modelo de penalidades para anomalias detectadas.

Tipo	**Penalidade para DaaS (€/s)**
Golden	$0,1$
Silver	$0,05$
Bronze	$0,025$

Sem perda de generalidade, assume-se que o serviço de IaaS foi contratado com um único SLA, que deve atender ao SLA do tipo *Gold* para simplificar o modelo e não haver necessidade do provedor de DaaS renegociar o custo dos serviços a cada vez que vender um serviço de DaaS. As penalidades de IaaS são definidas conforme um valor proporcional ao tempo em que o serviço estiver indisponível.

7.5 Injeção de Anomalias de Tráfego

A probabilidade de violação dos SLAs foi definida com base no estudo realizado por Goudarzi *et al.* [Goudarzi et al. 2012], que tratou as violações de SLA como uma função tipo de SLA contratado e não como um valor fixo, que traria prejuízos maiores aos SLAs mais restritivos. O percentual de violação de SLA (p_{anom}) é 50% a mais do que o percentual máximo aceitável de violação para cada tipo de SLA contratado, que é dado pela Equação 7.1.

$$p_{anom} = 1,5 \cdot (1 - p_{SLA}(z)) \tag{7.1}$$

7.6 Escalonamento baseado em Custo Integrado ao Mecanismo de Detecção de Anomalias

O Algoritmo 7.1 apresenta a estratégia de escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias para alocar a máquina virtual u_z, que utiliza a fonte de dados s_z como entrada para o processamento de uma consulta. O algoritmo recebe como entrada os vetores contendo todos os custos de execução da VM ($\vec{Ce}_z$) e outro vetor contendo os custos de recebimento dos dados ($\vec{Cd}_z$).

Percebe-se que antes do escalonador realizar a escolha dos centros de dados, ocorrem chamadas a métodos que atualizam a incidência de anomalias e os custos. Na Linha 2 é chamado um método que consulta o detector de anomalias. Há ainda a possibilidade desse método ser chamado por um módulo externo, assim que uma anomalia for detectada. Se forem detectadas anomalias em algum centro de dados, o custo dessas anomalias é calculado. O método chamado na Linha 3 acrescenta aos custos de execução dos centros de dados que estão em falha o custo estimado para as anomalias de tráfego identificadas.

Esse acréscimo de custo para o provedor de serviços serve para auxiliar o mecanismo de escalonamento baseado em custo. Suponha que o centro de dados d_i apresente um percentual de anomalias de tráfego para a meta de disponibilidade. Se o custo da penalidade causada por anomalias for inferior aos custos de execução nos demais centros de dados, então mesmo com um acréscimo de custo, a VM u_z será executada no centro de dados d_i. O método chamado na Linha 4 atualiza os custos de anomalias para o acesso a fontes de dados nos

centros de dados em falha.

Após a atualização dos custos das possíveis anomalias, o mecanismo de escalonamento é o mesmo que foi apresentado no Algoritmo 6.1 da Seção 6.2.3, que retorna os centros de dados que minimizam os custos de execução e de acesso à fonte de dados, respectivamente.

Algoritmo 7.1: Algoritmo de escalonamento baseado em custo para a VM u_z integrado ao mecanismo de detecção de anomalias.

Entrada: u_z, s_z, $\vec{Ce}_z$, $\vec{Cd}_z$, D

Saída: $\{d_i, d_j\}$

1 **início**

```
/* método chamado quando ocorrem anomalias          */
```

2 **atualize_anomalias_encontradas()**

```
/* adiciona custos de anomalias para execução de VMs */
```

3 **atualize_custos_de_execucao()**

```
/* adiciona custos de anomalias para fontes de dados */
```

4 **atualize_custos_de_fontes_de_dados()**

5 $d_i \longleftarrow \underset{d_i \in D}{\operatorname{argmin}} \mathrm{Ce_z}$;

6 **se** ***(contains(***d_i***,***s_z***) == 1)*** **então**

7 $t_i \longleftarrow 0$;

8 $d_j \longleftarrow \underset{d_j \in D}{\operatorname{argmin}} \mathrm{Cd_z}$;

Resultado: $\{d_i, d_j\}$

7.7 Impacto do Custo de Anomalias

Nesta seção serão analisados os custos e as penalidades obtidos com a aplicação do escalonamento baseado em custo apresentado no Capítulo 6, que será chamado de escalonamento baseado em custo **padrão**, e o escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, ou simplesmente escalonamento **integrado**.

No Apêndice F encontram-se detalhes sobre as decisões propostas pelos dois tipos de escalonamento em estudo, os custos e as penalidades referentes a cada VM. As Tabelas F.6, F.7 e F.8 apresentam o impacto das anomalias para o escalonamento padrão para as VMs 1, 2 e 3. O mecanismo de escalonamento integrado é detalhado nas Tabelas F.9, F.10 e F.11.

As anomalias foram injetadas seguindo uma função de probabilidade de acordo com o tipo de SLA (vide Equação 7.1). Para o cálculo das penalidades, apenas as anomalias que excedem o número máximo aceitável são contabilizadas. As anomalias foram injetadas no centro de dados de menor custo. Neste estudo, esse centro de dados é o d_1. A Tabela 7.4 apresenta o número de anomalias que foi injetado no centro de dados d_1, o número máximo aceitável de anomalias para cada VM e o número de anomalias excedente.

Tabela 7.4: Número de anomalias injetadas, número máximo aceitável de anomalias e número de anomalias excedentes.

	VM 1 (Máx. 3,6)		**VM 2 (Máx. 6,98)**		**VM 3 (Máx. 14,796)**	
Tempo (h)	**#Anomalias**	**# Anomalias Excedentes**	**#Anomalias**	**# Anomalias Excedentes**	**#Anomalias**	**#Anomalias Excedentes**
0	3	0,0	7	0,02	9	0,000
1	5	1,4	10	3,02	21	6,204
2	4	0,4	10	3,02	19	4,204
3	8	4,4	17	10,02	27	12,204
4	8	4,4	12	5,02	24	9,204
5	2	0,0	7	0,02	15	0,204
6	5	1,4	11	4,02	23	8,204
7	3	0,0	6	0,00	21	6,204
8	7	3,4	11	4,02	22	7,204
9	5	1,4	13	6,02	23	8,204
10	9	5,4	16	9,02	29	14,204
11	5	1,4	10	3,02	18	3,204
12	2	0,0	8	1,02	18	3,204
13	5	1,4	13	6,02	32	17,204
14	2	0,0	9	2,02	18	3,204
15	6	2,4	10	3,02	25	10,204
16	4	0,4	12	5,02	28	13,204
17	3	0,0	7	0,02	17	2,204
18	3	0,0	8	1,02	27	12,204
19	6	2,4	12	5,02	20	5,204
20	2	0,0	5	0,00	20	5,204
21	6	2,4	11	4,02	22	7,204
22	11	7,4	17	10,02	22	7,204
23	6	2,4	11	4,02	18	3,204
24	3	0,0	6	0,00	13	0,000
Total:	**123**	**42,4**	**259**	**88,44**	**531**	**168,692**

Os custos do processamento das consultas nas VMs dos dois mecanismos de escalona-

mento ao longo de cada hora estão apresentados na Figura 7.8. O escalonamento integrado apresenta casos em que o valor excede o praticado pelo escalonamento padrão. Isso ocorre porque o risco de violação de SLAs ao executar uma VM neste centro de dados aumenta com a incidência de anomalias. Contudo, é difícil observar essa diferença no gráfico. Em alguns casos ela é muito discreta. Por exemplo, para a VM 1, no tempo 3h, o custo do serviço de DaaS para o escalonamento padrão é 0,666026 (Tabela F.6) e no integrado é de 0,863315 (Tabela F.9). O custo deste centro de dados para o provedor aumenta devido à incidência de anomalias e ele não se torna elegível pelo mecanismo de escalonamento.

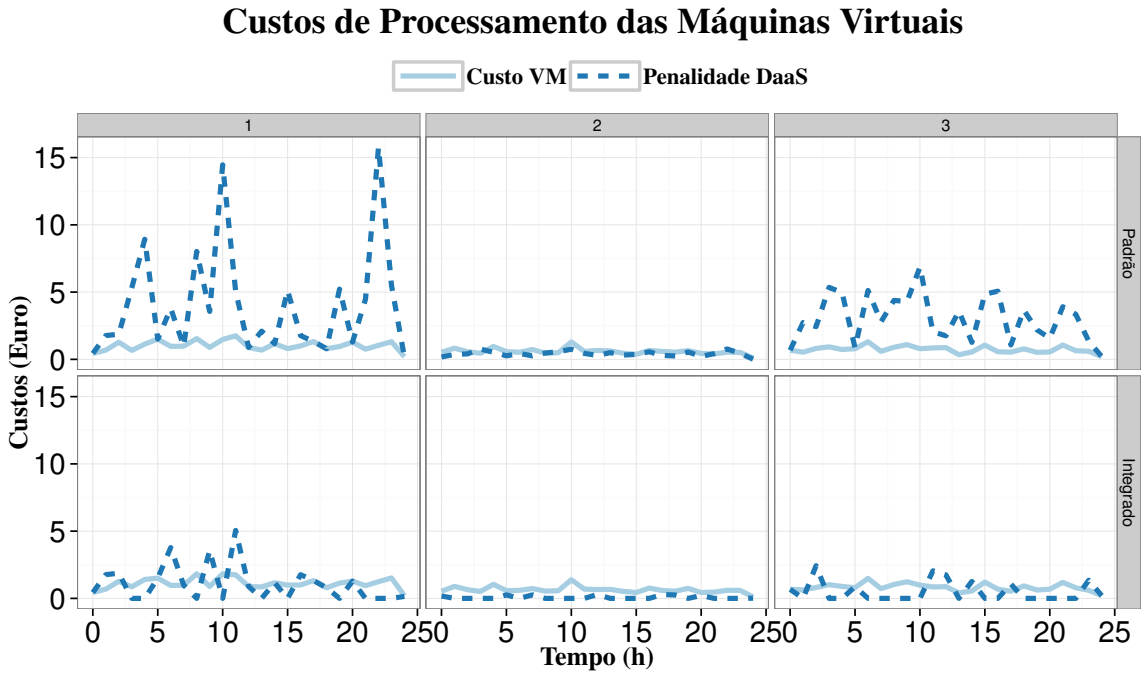
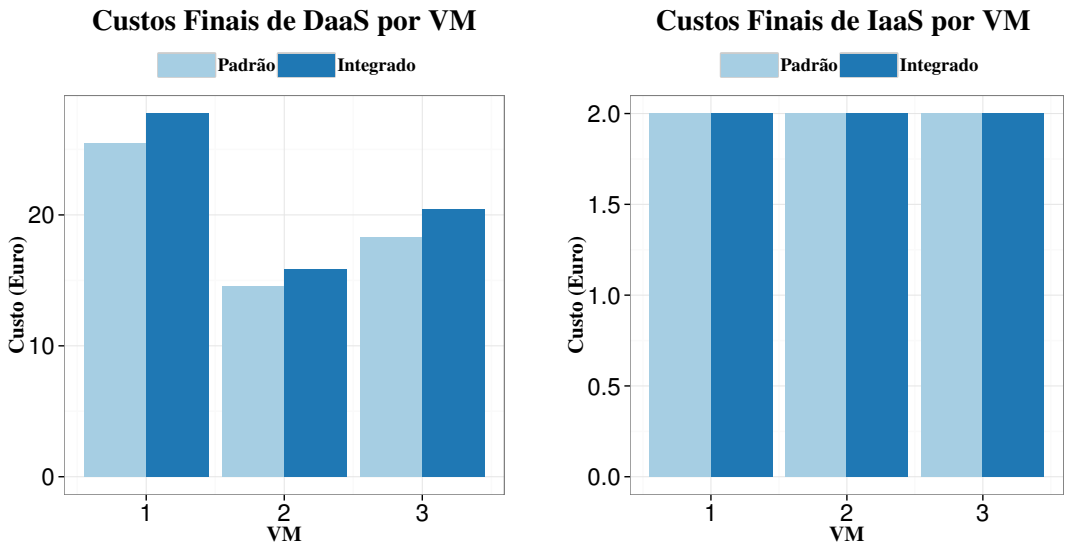


Figura 7.8: Custos das 3 VMs utilizando os dois mecanismos de escalonamento.

Por uma questão de fluidez do texto, as tabelas referentes à decisões de escalonamento, custos para os serviços e penalidades do escalonamento padrão e do integrado a cada hora de execução foram suprimidas desta seção. Elas se encontram no Apêndice F. As tabelas das VMs dos tipos 1, 2 e 3 do escalonamento padrão são respectivamente: F.6, F.7 e F.8. As Tabelas F.9, F.10 e F.11 correspondem os resultados obtidos para as VMs dos tipos 1, 2 e 3 do escalonamento integrado, respectivamente.

Em geral, o acréscimo nos custos totais quando se emprega a estratégia de escalonamento integrado é cerca de 10% maior do que a do escalonamento padrão. Esses valores podem ser

verificados na Figura 7.9a. Um sumário com os custos totais de cada VM referentes às 24 horas de experimento na Tabela 7.5, bem como o custo médio por hora e o desvio padrão por hora.



(a) DaaS (b) IaaS

Figura 7.9: Custo total para os serviços de DaaS e IaaS de cada VM.

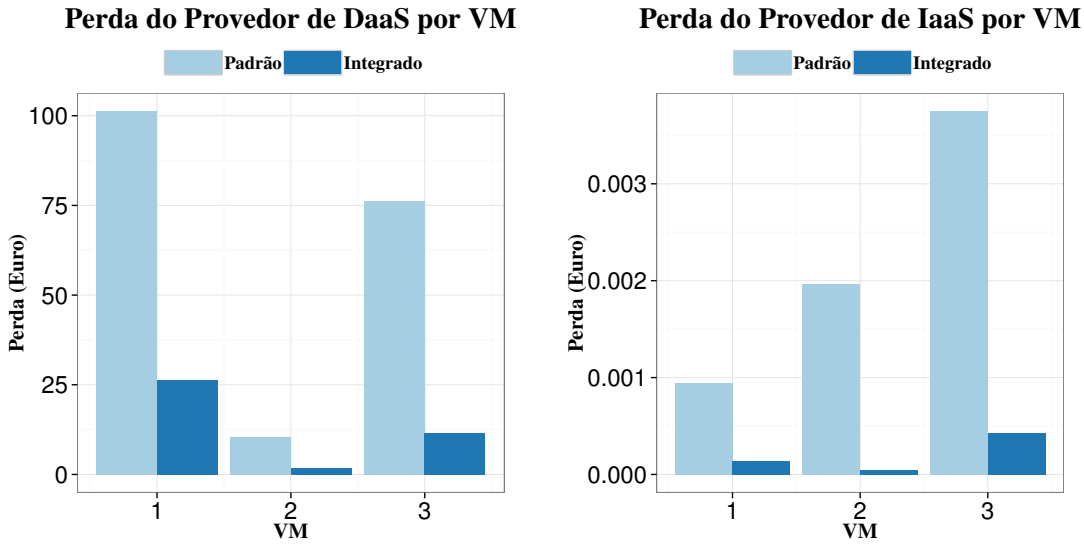
Os custos de IaaS pagos pelo provedor de DaaS são determinados pelo número de horas de utilização da infraestrutura, logo eles são os mesmos independentemente do mecanismo de escalonamento das VMs. Neste caso, o custo total de cada VM é €2 e o custo por hora é €0,08 (Figura 7.9b).

A Figura 7.10 trata das penalidades aplicadas aos serviços de DaaS e IaaS durante o tempo de execução das VMs. Nem sempre é vantajoso financeiramente escolher o centro de dados com menor custo. Quando o valor da penalidade no centro de dados de menor custo é superior à diferença de custo entre esse centro de dados e outro, então a alocação da VM é feita em um centro de dados que não minimiza os custos.

Os resultados referentes às penalidades de DaaS apresentam uma grande diferença, principalmente quando comparada à diferença de custo de implantação entre as duas abordagens. As penalidades de DaaS no total para VM #1 no escalonamento integrado, por exemplo, correspondem a aproximadamente 26% das penalidades do escalonamento padrão, ou seja, são cerca de 4 vezes inferiores (385% menores). As penalidades geradas pelo escalonamento

Tabela 7.5: Custos totais das VMs para DaaS utilizando o escalonamento integrado e o escalonamento padrão,

VM	Escalonador	Custo Total	Custo Médio por Hora	σ por Hora
1	Padrão	25,493689	1,019747542	0,364567546
1	Integrado	27,744306	1,109772238	0,401932738
2	Padrão	14,551529	0,582061180	0,218658274
2	Integrado	15,863584	0,634543367	0,232929210
3	Padrão	18,299410	0,731976402	0,249611926
3	Integrado	20,428799	0,817151970	0,284202724



(a) DaaS

(b) IaaS

Figura 7.10: Perda total para os serviços de DaaS e IaaS em cada VM.

para as VMs # 2 e # 3 são cerca de 17% e 15% das penalidades do escalonamento padrão, o que corresponde a uma redução superior a 500% e 600%, respectivamente. A Tabela 7.6 apresenta o detalhamento das penalidades aplicadas ao provedor de DaaS durante as 24 horas de experimento.

Tabela 7.6: Perdas totais resultantes das penalidades aplicadas aos serviços de DaaS.

VM	Escalonador	Custo Total	Custo Médio por Hora	σ por Hora
1	Padrão	101.24581070	4.049832430	4.083368474
1	Integrado	26.31741745	1.052696698	1.356620302
2	Padrão	10.42951276	0.417180510	0.185687972
2	Integrado	1.75480942	0.070192377	0.115633703
3	Padrão	76.35332412	3.054132965	1.762380658
3	Integrado	11.65437355	0.466174942	0.751157379

As penalidades do provedor de IaaS estão na Tabela 7.7. Note que o escalonamento integrado reduz perdas no provedor de IaaS, pois a indisponibilidade com anomalias independe do nível de abstração do serviço. Logo, quando o serviço é alocado em um centro de dados livre de falhas, a perda no nível de IaaS também é reduzida.

Tabela 7.7: Perdas totais resultantes das penalidades aplicadas aos serviços de IaaS.

VM	Escalonador	Custo Total	Custo Médio por Hora	σ por Hora
1	Padrão	0.000942222	3.77E-05	4.43E-05
1	Integrado	0.000142222	5.69E-06	1.16E-05
2	Padrão	0.001961618	7.85E-05	6.80E-05
2	Integrado	4.58E-05	1.83E-06	6.22E-06
3	Padrão	0.003748711	0.000149948	0.000101995
3	Integrado	0.000431733	1.73E-05	3.16E-05

A Tabela 7.8 apresenta um sumário com as penalidades e os ganhos relativos entre o valor obtido pelo escalonamento baseado em custo integrado ao mecanismo de anomalias de tráfego e o escalonamento baseado em custo padrão.

Tabela 7.8: Perdas e custos relativos entre os valores obtidos pelo escalonamento integrado e o escalonamento padrão baseado em custo.

	Integrado / Padrão			
VM	**Perda DaaS**	**Perda IaaS**	**Custo IaaS**	**Custo DaaS**
1	0,259935866	0,1509434	1	1,0882814
2	0,168254209	2,34E-02	1	1,0901661
3	0,152637409	0,1151685	1	1,1163638

Para apresentar uma ideia da perda total gerada pelas penalidades aplicadas ao provedor de DaaS e IaaS ao longo das horas nas duas abordagens de escalonamento, serão apresentados gráficos da função de densidade acumulada das perdas (Figura 7.11). O prejuízo causado com anomalias de tráfego é bastante reduzido ao longo de todo o processo de execução, quando aplicado o mecanismo de escalonamento integrado.

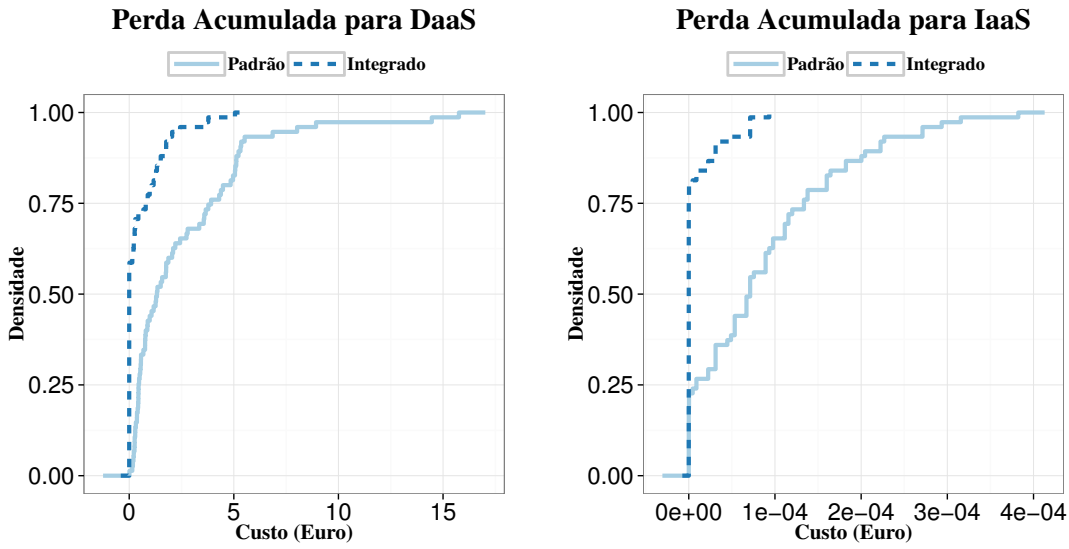
Até 75% do trabalho total computado não houve perda para DaaS utilizando o escalonamento integrado. Por outro lado, este valor era em torno de €5 para o escalonamento padrão. Com este mesmo valor representou a perda total do escalonamento integrado (100%). Quando a perda acumulada para DaaS no escalonamento padrão chegou em 100%, os custos com penalidades eram superiores a €15. O gráfico de perda acumulada para DaaS é apresentado na 7.11a.

Quanto à perda para IaaS, o comportamento é semelhante. A perda total foi inferior a €10^{-4} para o escalonamento integrado e superior a €$4 \cdot 10^{-4}$ para o escalonamento padrão. Esses resultados são mostrados na Figura 7.11b.

7.8 Conclusões

O escalonamento integrado reduziu as penalidades a aproximadamente 26%, $16,8\%$ e $15,3\%$ do valor que foi obtido pelo escalonamento padrão para as VMs 1, 2 e 3, respectivamente. Houve uma redução também nas penalidades no serviço de IaaS também, pois as VMs foram alocadas a centros de dados livres de falhas de rede.

Os custos de IaaS foram definidos de forma fixa, portanto já era esperado que não hou-



(a) DaaS

(b) IaaS

Figura 7.11: Perda acumulada para os serviços de DaaS e IaaS em cada VM.

vesse modificação no valor dos custos de utilização dos serviços. Os custos da utilização dos serviços de DaaS são mais altos, pois em alguns casos, as VMs não foram alocadas aos centros de dados de menor custo. Contudo, mesmo com essa mudança, o acréscimo de custo ficou em torno de 10%.

Capítulo 8

Trabalhos Relacionados

"O medo tem alguma utilidade, mas a covardia não."

Mahatma Gandhi

Os trabalhos relacionados não resolvem o problema de investigar anomalias no tráfego da rede de serviços de computação em nuvem de modo a alimentar o sistema de gerenciamento com informações estratégicas. Eles propõem arquiteturas ou ferramentas em diferentes camadas de serviços, tais como alocação de recursos, negociação de SLA, gestão, implantação, estimativa de métricas e desenvolvimento de novos serviços.

Os trabalhos relacionados não advogam a ideia de um serviço de detecção integrado ao sistema de contabilidade gerencial, aplicado apenas ao tráfego que as partes interessadas necessitam para a tomada de decisões das entidades envolvidas na contratação do serviço, a cliente e a provedora.

Este trabalho difere dos trabalhos do estado da arte por investigar soluções que dão suporte a decisões gerenciais tanto da parte cliente como da provedora de serviços. Este trabalho apresenta um arcabouço para a prática de monitoramento e análise de tráfego podendo ser estendido e as técnicas de monitoramento e detecção de anomalias de tráfego, de outras metas técnicas relacionadas ao desempenho dos serviços e da estimativa dos custos de tráfego normal e anômalo podem ser substituídas sem perda de generalidade do modelo proposto.

Nas próximas seções deste capítulo serão recapituladas algumas das principais técnicas comentadas nesta tese que procuram atender a diferentes problemas levantados ao longo deste trabalho, como detecão de violações de SLAs, detecção de anomalias baseada em

comportamento e tarifação de serviços de computação em nuvem.

8.1 Detecção de Violações de SLAs

Deri e Fusco [Deri e Fusco 2013] propuseram uma arquitetura para prover monitoramento *online* de tráfego em centros de dados. Contudo, eles concentraram o trabalho em como distribuir sondas para capturar o tráfego em vários centros de dados e em como correlacionar os dados processados de forma distribuída em um nó centralizado. Eles utilizaram ferramentas de código aberto parar realizar esse processamento.

Serrano *et al.* [Serrano et al. 2013] definiram um conjunto de normas para inferir restrições em um SLA de serviços de computação em nuvem. Eles introduziram o CSLA (*Cloud Service Level Agreement*) e uma linguagem de descrição intuitiva para especificar objetivos de acordos utilizando pesos.

A computação em nuvem é uma evolução das redes de telecomunicações, segundo Mikkilineni e Sarathy [Mikkilineni e Sarathy 2009]. Ela pode ser vista como uma infraestrutura de telecomunicações de redes inteligentes (IN - *Intelligent Network*). Mikkilineni e Sarathy [Mikkilineni e Sarathy 2009] propuseram um modelo de camadas de *software* para a criação, a garantia e a entrega de serviços em nuvem. Nessa arquitetura é definida uma camada para a mediação dos recursos virtuais chamada de VRML (*Virtual Resources Mediation Layer*) que é responsável por:

I. Mediar a alocação dinâmica dos recursos na rede;

II. Gerenciar falhas, configuração, contabilidade, desempenho e segurança (FCAPS - *Fault, Configuration, Accounting, Performance and Security*);

III. Permitir o desenvolvimento ágil de serviços convergentes largamente escaláveis e interoperáveis.

Mikkilineni e Sarathy [Mikkilineni e Sarathy 2009] argumentam que a camada VRML permite o desenvolvimento de uma rede de colaboração de serviços (SCN - *Service Collaboration Network*) com a participação de múltiplos provedores e propõem um modelo de referência para computação em nuvem que inclui o gerenciamento FCAPS que define vários

papéis no provimento da infraestrutura de *sofware*, de componentes, virtual e física como serviços.

Habib *et al.* [Habib et al. 2003] projetaram um sistema chamado SLAM (*Service Level Agreement Monitor*) para verificar métricas de QoS para serviços diferenciados. O sistema opera em roteadores de borda e auxilia no balanceamento de carga da rede.

8.2 Detecção de Anomalias baseada em Entropia

Wang, Talwar, Schwan e Ranganathan [Wang et al. 2010] realizaram um experimento para demonstrar a viabilidade e a acurácia do método EbAT, comparando-o a métodos baseados em detecção por limiar. As anomalias consideradas nesse trabalho foram erros em operações, falhas de *hardware* ou *software* e super/subprovisionamento de recursos. Os autores fazem um levantamento de técnicas tradicionais de detecção e criticam o fato de serem fundamentadas em complexas análises estatísticas ou a falta de escalabilidade dos métodos que realizam mineração em grandes quantidades de dados com métricas desagregadas.

Uma contribuição do trabalho foi a categorização de técnicas aplicadas a detecção de anomalias, classificando-as em: i) abordagens baseadas em limiar; ii) métodos estatísticos. Exemplos de técnicas com abordagens baseadas em limiar são: FAR (*false-alarm incremental rate*) e detecção pós-fato. Os resultados obtidos demonstram que a técnica EbAT apresenta um desempenho superior aos métodos baseados em limiares com um aumento médio de 18,9% com relação à métrica F1. Houve ainda uma redução em torno de 50% no número de falsos alarmes quando comparada ao método quase-ótimo FAR[Wang et al. 2010].

Quanto às limitações da técnica proposta por Wang *et al.*, estão a dificuldade de configurar os parâmetros de detecção em meio a diferentes tipos de aplicações de rede, o que requer conhecimento sobre padrões normais de tráfego e, em alguns casos, intervenção humana para realizar identificação visual de discrepâncias, ou configuração dos níveis de variações que caracterizam de fato anomalias.

Técnicas de aprendizagem de máquina não-supervisionadas também podem ser aplicadas para a detecção de anomalias. Esse método assume que os dados possuem uma distribuição de probabilidade conhecida, comumente a Gaussiana. Os valores com probabilidade muito baixa são considerados anomalias. O objetivo dessa análise de probabilidades é encontrar

um limiar de probabilidade que maximiza a acurácia da detecção e que será utilizado como valor-base para a estimativa dos valores anômalos [Ng 2014].

Smith *et al.* [Smith et al. 2010] propuseram um mecanismo autonômico para detecção de anomalias em sistemas de computação em nuvem. Os autores definiram um conjunto de técnicas que envolvem a *transformação dos dados* para uniformizar o formato dos dados para a análise, a *extração de características* para reduzir o tamanho dos dados e a *aprendizagem não-supervisionada* usando agrupamento (*clustering*) para detectar nós que estão se comportando de uma forma discrepante dos demais (*outliers*).

No mecanismo autonômico proposto por Smith *et al.* [Smith et al. 2010], as anomalias são calculadas com base no comportamento do sistema. No total são monitoradas 52 variáveis em cada nó do sistema de computação em nuvem, que caracterizam estatísticas sobre o tempo de execução do nó, incluindo seus processadores, memória, dispositivos de E/S, conexões de rede e discos. Esses valores são coletados em nível de sistema operacional. Os estados de tempo de execução em um nó são definidos pelos valores amostrados para as variáveis em cada ponto no tempo.

Os resultados obtidos por Smith *et al.* [Smith et al. 2010] demonstram que o modelo proposto pode ser usado para aumentar a confiança no funcionamento de sistemas de computação em nuvem de larga escala. No entanto, esse modelo não atua no nível de monitoramento do tráfego da rede, mas realiza coleta de variáveis em nível de sistema operacional. Além disso, a técnica proposta realiza amostragens e os dados são coletados em intervalos de 5 minutos. Naturalmente, há um intervalo para detecção e perda de informações que poderiam interferir nos resultados.

Benetazzo *et al.* [Benetazzo et al. 2007] propuseram a análise de tráfego agregado por meio da determinação das curvas de taxa de intervalo empíricas (RIC – *rate-interval curves*), que consiste em dividir as medições de vazão em quantis, visando delinear propriedades de escala e outros diagnósticos. O método baseado em RIC caracteriza o tráfego da rede sem a necessidade de conhecimento do modelo dos fluxos *a priori*.

Nychis *et al.* [Nychis et al. 2008] avaliaram a eficiência da técnica de detecção de anomalias baseada em entropia para diferentes métricas. Eles concluíram que a porta e o endereço possuem distribuições fortemente correlacionadas com a capacidade de detecção. Ambas as métricas apresentaram resultados semelhantes quando aplicadas para detectar anomalias no

tráfego da rede.

Quan *et al.* [Quan et al. 2009] compararam dois métodos de detecção baseada em entropia para classificar comportamentos de rede: *entropia de rede* e *entropia de rede normalizada relativa* (NRNE – *normalized relative network entropy*). Duas distribuições de probabilidades diferentes podem compartilhar o mesmo valor de entropia, mesmo tendo vetores de probabilidade discrepantes, o que é um problema. Para evitar esse problema, os autores empregaram o conceito de entropia relativa, ou desvio de *Kullback-Leibler* (KL), que representa a diferença entre duas distribuições de probabilidade. O NRNE apresentou um melhor desempenho; em contrapartida, requer um maior número de atributos de entrada para detectar anomalias.

8.3 Tarifação de Serviços de Computação em Nuvem

Diversos mecanismos de tarifação foram propostos na literatura e utilizados para a cobrança de serviços de telecomunicações [Barachi et al. 2008][Oumina e Ranc 2007][Dressler et al. 2004] [F´oll et al. 2005][Farrell et al. 2000][Glass et al. 2000]. As RFCs 2906 [Farrell et al. 2000] e 2977 [Glass et al. 2000] fazem um levantamento e especificação dos requisitos para o provimento dos serviços de Autenticação, Autorização e Contabilização (AAA - *Authentication, Authorization, and Accounting*).

Föll et al. [F´oll et al. 2005] propõem um mecanismo de contabilização e cobrança orientado a serviço que desabilita alguns componentes gerais definidos nas arquiteturas convencionais e ativa componentes necessários à contabilização de um serviço particular. Esse trabalho aplica-se a estudar cobrança por tipo de serviço que é provido em redes 3G e B3G (*Beyond 3G*), envolvendo a contabilização em redes móveis celulares, atacando o problema de custos entre células e atribuição de valores aos serviços prestados por diferentes estações-base que podem prover serviços em trajetos percorridos por clientes em mobilidade. Os autores seguem a abordagem proposta pelo grupo de trabalho NSIS da IETF que procura especificar um protocolo para configuração do serviço de contabilização [Dressler et al. 2004].

Schwarzkopf *et al.* [Schwarzkopf et al. 2013] propuseram uma nova arquitetura de escalonamento chamada de escalonador *Omega*, que emprega compartilhamento de estado e controle de concorrência otimista e livre de retenção para proporcionar extensibilidade e

escalabilidade. O escalonador Omega fornece diretrizes para a alocação de recursos e não estratégias propriamente ditas.

Justafort e Pierre [Justafort e Pierre 2012] propuseram a alocação de VMs entre centros de dados, também chamados de ambientes *internuvens*, com foco em minimizar a utilização de largura de banda e o atraso entre VMs. Eles desenvolveram uma solução baseada em Programação Linear Inteira Mista (*MILP*). Como essa solução não é escalável, eles desenvolveram heurísticas para lidar com o problema de alocação de VMs, que é NP-difícil. Contudo, eles não levam em consideração o preço final das instâncias.

Kantarci *et al.* [Kantarci et al. 2012] propuseram um esquema para alocação de VMs intra e inter centro de dados para sistemas em nuvem de larga escala com foco em minimizar o consumo de energia. Os centros de dados estão entre os principais contribuintes para a emissão do gás GHC (*Green-House Gas* – GHG) dentre os serviços de TI. O *trade-off* está em como garantir SLAs versus a minimização de despesas operacionais. Os centros de dados mais eficientes energeticamente são escolhidos para hospedar as VMs. Essa abordagem também aplica Programação Linear Inteira Mista. Os resultados obtidos mostraram uma redução de 10% do consumo total de energia e melhoria da utilização de recursos.

Patel e Sarje [Patel e Sarje 2012] propuseram um método de provisionamento de VMs com objetivo de minimizar violações de SLA e aumentar os lucros dos prestadores de serviços em nuvem. Eles desenvolveram um mecanismo baseado em limiares para balanceamento de carga em nuvens federadas. Foram considerados modelos de preços para VMs sob demanda e para instâncias reservadas. Os resultados mostraram que o modelo pode diminuir a taxa de violação de SLAs, mas acrescentou um problema de alocação de recursos e de balanceamento de carga entre os centros de dados.

Zaman e Grosu [Zaman e Grosu 2013] desenvolveram um mecanismo de alocação de recursos baseado em leilões para provisionamento e alocação dinâmicos de VMs com objetivo de minimizar o preço total de VMs sob a restrição de que um trabalho executado em uma VM deve ser concluído dentro de um determinado prazo D.

Li *et al.* [Li et al. 2012] desenvolveram um algoritmo adaptativo para encontrar o melhor plano de alocação com objetivo de maximizar a disponibilidade de recursos de IaaS, evitando superutilização de recursos. Li *et al.* argumentam que essas práticas são as principais responsáveis para o aumento do lucro.

Xiong *et al.* [Xiong et al. 2011b] também trataram do problema de sobrecarga do sistema. Eles desenvolveram o ActiveSLA, um sistema de controle de admissão para sistemas de Banco de Dados-como-um-Serviço com objetivo de minimizar violações de SLAs e maximizar o lucro. Em outro trabalho, Xiong *et al.* [Xiong et al. 2011a] resolveram esse problema utilizando uma abordagem diferente, por meio de um sistema de gestão de recursos ciente de custos.

Capítulo 9

Considerações Finais

"Combati o bom combate, terminei a minha carreira, guardei a fé."

II Timóteo 4,7

Clientes de serviços de computação em nuvem não possuem controle sobre a infraestrutura física que suporta a execução dos serviços. É importante que informações gerenciais relevantes ao negócio do cliente estejam disponíveis a ele e que possam ser acessadas de modo normatizado. O processo de notificação de anomalias de tráfego em sistemas de computação em nuvem pode fornecer informações úteis tanto para o cliente e quanto para o provedor, ou mesmo a uma entidade externa que realiza auditorias de qualidade dos serviços.

Nesta tese foi proposto um mecanismo de governança de rede para serviços de computação em nuvem que normatiza que informações relevantes ao negócio sobre o desempenho dos serviços estejam disponíveis de modo padronizado tanto aos clientes, quanto aos provedores que foi apresentação no Capítulo 1 ([Oliveira et al. 2015a]).

Este trabalho contribuiu com o projeto de uma arquitetura (TADE) para detecção e gerenciamento de anomalias de tráfego para serviços de computação em nuvem baseada em acordos de nível de serviço que possui uma API aberta e acessível tanto para o provedor quanto para o cliente baseada no modelo de governança de rede proposto ([Oliveira et al. 2014a]). A arquitetura TADE foi descrita no Capítulo 3.

Outra contribuição deste trabalho foi uma análise da técnica de detecção baseada em entropia pura, EbAT, (Capítulo 4) e um melhoramento à essa técnica por meio de uma abor-

dagem híbrida, que une aprendizagem de máquina à EbAT que foi proposta no Capítulo 5 ([Oliveira et al. 2014b]).

Um produto do doutorado sanduíche na Universidade Técnica de Dresden foi o levantamento de um modelo de custo de máquinas virtuais (VMs) para DaaS e um mecanismo de escalonamento baseado nesses custos. Esse modelo foi proposto no Capítulo 6. A modelagem das VMs foi inicialmente feita de modo aleatório e o modelo foi validado por meio de simulações. Esse modelo foi utilizado como estudo de caso para validar o modelo de tarifação confiável para serviços de computação em nuvem ([Oliveira et al. 2015b] [Oliveira et al. 2015c]).

O Capítulo 7 reúne todos os trabalhos apresentados nos capítulos anteriores. Eles atuam em conjunto para dar suporte a um modelo de tarifação confiável, que é baseado em um processo de transparência no desempenho dos serviços de computação em nuvem e em estimativas de perdas financeiras causadas por tráfego anômalo que, quando identificadas, são acardas pelo provedor desse serviço.

O modelo de tarifação confiável foi validado por meio de uma análise de custos da incidência de anomalias no provimento de serviços de computação em nuvem em diferentes níveis. O escopo de uma anomalia nesse contexto é uma violação de um SLA de disponibilidade em um intervalo de 1 segundo, que é um tempo curto para disponibilidade no nível de aplicação e alto para o nível de rede. O objetivo deste trabalho é analisar o impacto do custo de tráfego anômalo quando um provedor de DaaS é também cliente de IaaS.

O modelo de tarifação confiável foi incorporado a um mecanismo de escalonamento baseado em custo que leva em conta os resultados dessa análise de impacto. Os recursos são escalonados para os centros de dados que minimizem o custo total de prestação do serviço, o que envolve o *trade-off* entre o escalonamento baseado em custo (proposto anteriormente) e o prejuízo causado pela perda financeira devido ao não cumprimento dos SLAs.

A metodologia empregada na avaliação do modelo de tarifação confiável envolveu experimentação em ambiente real e simulação. A experimentação foi utilizada para caracterizar o comportamento de execução das VMs e o tráfego recebido e enviado por elas no processamento de DaaS em uma nuvem privada. As métricas de execução estimadas foram empregadas para calibrar o modelo de tarifação confiável proposto.

A implementação do modelo de tarifação confiável proposto nesta tese envolveu 6 fases.

Na Fase 1 foi feito um experimento real com uma aplicação de DaaS para analisar como as VMs se comportam na prática. As variáveis do modelo analítico foram ajustadas conforme essas estimativas.

Na Fase 2 ocorreu o processo de detecção de anomalias. Durante o provimento dos serviços de DaaS é feito o monitoramento do tráfego e quando ocorre uma violação, essas anomalias são identificadas, contabilizadas e publicadas em um canal de comunicação compartilhado entre cliente e provedor (arquitetura TADE). Essa fase segue o modelo de governança de rede proposto para sistemas de computação em nuvem.

A injeção de anomalias foi feita na Fase 3. Na Fase 4 realizou-se a predição do impacto do custo de uma anomalia para o tipo de serviço a ser escalonado, caso a VM viesse a ser executada em um centro de dados (DC) que estivesse apresentando problemas de disponibilidade.

Na Fase 5 foi feito o escalonamento baseado nos custos de cada DC de modo integrado ao módulo de detecção de anomalias implementado. Na Fase 6 foram analisados os resultados obtidos para os custos dos serviços e das perdas financeiras após o emprego de um escalonamento integrado à governança de rede para serviços de computação em nuvem. Verificou-se qual o ganho do escalonamento utilizar as estimativas de perda financeira em detrimento de apenas se basear nos custos pontuais de cada DC.

Quanto a balanceamento de carga, o esquema de custos pode ser ajustado de modo que DCs subutilizados tenham seus custos diminuídos e os com maior utilização se tornem mais altos. No entanto, essa questão está fora do escopo desta tese.

Os códigos fontes, materiais e informações suplementares referentes a esta tese de doutorado podem ser encontrados em um repositório púlico no seguinte endereço: `https://anacristina@bitbucket.org/anacristina/thesis.git`. O sistema de controle de versões do repositório é o *git*. Para baixá-lo via linha de comanho, utilizar os seguintes comandos (para sistemas operacionais Linux e afins):

```
1 $ mkdir /path/to/your/project
2 $ cd /path/to/your/project
3 $ git init
4 $ git remote add origin https://anacristina@bitbucket.org/anacristina/
    thesis.git
```

Bibliografia

[Alharkan e Martin 2012] Alharkan, T. e Martin, P. (2012). IDSaaS: Intrusion Detection System as a Service in Public Clouds. *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pages 686–687.

[Amazon 2015] Amazon (2015). Amazon web services. http://aws.amazon.com/pt. Acessado em: Outubro, 2015.

[AMQP 2015] AMQP (2015). Amqp: Advanced message queueing protocol. http://www.amqp.org. Acessado em: Outubro, 2015.

[Antonello et al. 2012] Antonello, R., Fernandes, S., Kamienski, C., Sadok, D., Kelner, J., GóDor, I., Szabó, G., e Westholm, T. (2012). Deep packet inspection tools and techniques in commodity platforms: Challenges and trends. *J. Netw. Comput. Appl.*, 35(6):1863–1878.

[ao Brasileira de Normas Técnicas 2009] ao Brasileira de Normas Técnicas, A. (2009). *ABNT NBR ISO/IEC 38500: Information Technology Corporative Governance*. ABNT.

[Arfeen et al. 2013] Arfeen, M. A., Pawlikowski, K., McNickle, D., e Willig, A. (2013). The role of the weibull distribution in internet traffic modeling. In *Teletraffic Congress (ITC), 2013 25th International*, pages 1–8.

[Armbrust et al. 2010] Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., e Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53(4):50–58.

[Armbrust et al. 2009] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., e Zaharia, M. (2009). Above the

clouds: A berkeley view of cloud computing. Technical Report No. UCB/EECS-2009-28, Univ. of California, Berkeley.

[Arshadi e Jahangir 2014a] Arshadi, L. e Jahangir, A. H. (2014a). Benford's law behavior of internet traffic. *J. Netw. Comput. Appl.*, 40:194–205.

[Arshadi e Jahangir 2014b] Arshadi, L. e Jahangir, A. H. (2014b). An empirical study on tcp flow interarrival time distribution for normal and anomalous traffic. *International Journal of Communication Systems*, pages n/a–n/a.

[Barachi et al. 2008] Barachi, M. E., Glitho, R., e Dssouli, R. (2008). Charging for multi-grade services in the ip multimedia subsystem. In *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST '08. The Second International Conference on*, pages 10 –17.

[Basili et al. 2002] Basili, V., Caldiera, G., e Rombach, H. (2002). *The Approach. Encyclopedia of Software Engineering*. John Wiley and Sons, 2. ed. edition. pp. 578-583.

[Bauer et al. 2001] Bauer, D., Cannady, J., e Garcia, R. (2001). Detecting anomalous behavior: optimization of network traffic parameters via an evolution strategy. In *Southeast-Con 2001. Proceedings. IEEE*, pages 34 –39.

[Benetazzo et al. 2007] Benetazzo, L., Giorgi, G., e Narduzzi, C. (2007). On the analysis of communication and computer networks by traffic flow measurements. *Instrumentation and Measurement, IEEE Transactions on*, 56(4):1157–1164.

[Brauckhoff et al. 2006] Brauckhoff, D., Tellenbach, B., Wagner, A., May, M., e Lakhina, A. (2006). Impact of packet sampling on anomaly detection metrics. *Proceedings of the 6th ACM SIGCOMM on Internet measurement - IMC '06*, page 159.

[Burrows 2006] Burrows, M. (2006). The chubby lock service for loosely-coupled distributed systems. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, OSDI '06, pages 335–350, Berkeley, CA, USA. USENIX Association.

[Buyya 2009] Buyya, R. (2009). Market-Oriented Cloud Computing: Vision, Hype, and Reality of Delivering Computing as the 5th Utility. *9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, pages 1–1.

[Buyya et al. 2009] Buyya, R., Ranjan, R., e Calheiros, R. N. (2009). Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. *2009 International Conference on High Performance Computing & Simulation*, pages 1–11.

[Callado et al. 2009] Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., e Sadok, D. (2009). A survey on internet traffic identification. *Communications Surveys Tutorials, IEEE*, 11(3):37 –52.

[Callado et al. 2010] Callado, A., Kelner, J., Sadok, D., Kamienski, C. A., e Fernandes, S. (2010). Better network traffic identification through the independent combination of techniques. *J. Netw. Comput. Appl.*, 33(4):433–446.

[Chagas e Oliveira 2013] Chagas, H. e Oliveira, A. C. (2013). Desenvolvimento de mecanismos para melhoria do desempenho da análise de métricas de rede em tempo real. In *VIII Congresso Norte-Nordeste de Pesquisa e Inovação (CONNEPI)*.

[Chang et al. 2006] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., e Gruber, R. E. (2006). Bigtable: A distributed storage system for structured data. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7*, OSDI '06, pages 15–15, Berkeley, CA, USA. USENIX Association.

[Chen 2001] Chen, T. M. (2001). Increasing the observability of internet behavior. *Commun. ACM*, 44(1):93–98.

[Chhetri et al. 2012] Chhetri, M. B., Vo, Q. B., e Kowalczyk, R. (2012). Policy-Based Automation of SLA Establishment for Cloud Computing Services. *2012 12th IEEEACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pages 164–171.

[Cisco 2005] Cisco (2005). Cisco ios ip service level agreements: User guide. GNU Free License.

[Cloud&Heat 2015] Cloud&Heat (2015). Cloud & heat iaas pricing.

https://www.cloudandheat.com/en/pricing-iaas.html. Acessado em: 3 de Novembro de 2015.

[Collingridge 2015] Collingridge, D. (2015). Validating a questionnaire. http://www.methodspace.com/profiles/blogs/validating-a-questionnaire. Acessado em: Outubro, 2015.

[Costa et al. 2012] Costa, P., Migliavacca, M., Pietzuch, P., e Wolf, A. L. (2012). Naas: network-as-a-service in the cloud. In *Proceedings of the 2nd USENIX conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, Hot-ICE'12, pages 1–1, Berkeley, CA, USA. USENIX Association.

[Costa et al. 2009] Costa, R., Brasileiro, F., Filho, G. L., e Sousa, D. M. (2009). Oddci: on-demand distributed computing infrastructure. In *Proceedings of the 2nd Workshop on Many-Task Computing on Grids and Supercomputers*, MTAGS '09, pages 10:1–10:10, New York, NY, USA. ACM.

[Costa et al. 2010] Costa, R., Brasileiro, F., Lemos, G., e Sousa, D. (2010). Just in time clouds: Enabling highly-elastic public clouds over low scale amortized resources. Technical report, Federal University of Campina Grande / Federal University of Paraíba. http://www.lsd.ufcg.edu.br/relatorios_tecnicos/TR-3.pdf.

[Cramér 1946] Cramér, H. (1946). *Mathematical Methods of Statistics*. Princeton University Press, Princeton. ISBN 0-691-08004-6.

[Cusumano 2010] Cusumano, M. (2010). Cloud computing and SaaS as new computing platforms. *Communications of the ACM*, 53(4):27.

[Dean e Ghemawat 2008] Dean, J. e Ghemawat, S. (2008). MapReduce : Simplified Data Processing on Large Clusters. *Communications of the ACM*, 51(1):1–13.

[Dekkers 2007] Dekkers, P. (2007). Complex Event Processing: CORDYS, simplifying Business. Master's thesis, Radboud University Nijmegen, The Netherlands.

[Deri e Fusco 2013] Deri, L. e Fusco, F. (2013). Realtime microcloud-based flow aggregation for fixed and mobile networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 96–101.

[Dóra et al. 2013a] Dóra, P. M., Oliveira, A. C., e Moura, J. A. B. (2013a). A baseline for quality management in software projects. In *2nd International Conference on Informatics and Computer Sciences (CICCI) in conjuction with the 15th International Convention and Fair Informática*.

[Dóra et al. 2013b] Dóra, P. M., Oliveira, A. C., e Moura, J. A. B. (2013b). Improving quality in agile development processes. In *8th International Conference on Software Engineering and Applications*.

[Dóra et al. 2014a] Dóra, P. M., Oliveira, A. C., e Moura, J. A. B. (2014a). Selecting frameworks for multi-agent systems development for the oil industry. In *Congreso Internacional de Ingeniería Informática y Sistemas de Información (CIIISI)*.

[Dóra et al. 2014b] Dóra, P. M., Oliveira, A. C., e Moura, J. A. B. (2014b). Simultaneously improving quality and time-to-market in agile development. In Cordeiro, J. e van Sinderen, M., editors, *Communications in Computer and Information Science*, volume 457, pages 84–98. Springer Berlin Heidelberg, xii edition.

[Dressler et al. 2004] Dressler, F., Carle, G., Fan, C., Kappler, C., e Tschofenig, H. (2004). NSLP for Accounting Configuration Signaling. Internet-Draft draft-dressler-nsis-accounting-nslp-00.txt, IETF.

[Duan et al. 2015] Duan, Y., Fu, G., Zhou, N., Sun, X., Narendra, N., e Hu, B. (2015). Everything as a service (xaas) on the cloud: Origins, current and future trends. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 621–628.

[Duffield et al. 2005] Duffield, N., Lund, C., e Thorup, M. (2005). Learn more, sample less: control of volume and variance in network measurement. *Information Theory, IEEE Transactions on*, 51(5):1756 – 1775.

[Eaton et al. 2008] Eaton, J. W., Bateman, D., e Hauberg, S. (2008). *GNU Octave Manual Version 3*. Network Theory Ltd.

[Esper 2013] Esper (2013). Esper - complex event processing. http://esper.codehaus.org. Acessado em: Agosto, 2013.

[Eucalyptus 2014] Eucalyptus (2014). Eucalyptus: Open source aws compatible private clouds. https://www.eucalyptus.com. Acessado em: Outubro, 2014.

[Farrell et al. 2000] Farrell, S., Vollbrecht, J., Calhoun, P., Gommans, L., Gross, G., de Bruijn, B., de Laat, C., Holdrege, M., e Spence, D. (2000). Aaa authorization requirements - rfc 2906. Internet Engineering Task Force.

[F´oll et al. 2005] F´oll, U., Fan, C., Carle, G., Dressler, F., e Roshandel, M. (2005). Service-oriented accounting and charging for 3g and b3g mobile environments. In *9th IFIP/IEEE International Symposium on Integrated Network Management*.

[Fraga et al. 2013] Fraga, E., Brilhante, J., Costa, R., Brasileiro, F., Spohn, M., Gomes, R., Senger, H., Bignatto, P., Desani, D., Pereira, A., Garcia, V., Oliveira, A. C., Chagas, H., Ferreira, A., Navaux, P., Carvalho, O., Macêdo, R., Sá, A., e Trinta, F. (2013). Just-in-time clouds: An approach to federate private clouds. In *Saloon of Tools of the Brazilian Symposium on Computer Networks and Distributed Systems (SBRC) (In Portuguese)*, pages 1109–1116.

[Gerhardt e Silveira 2009] Gerhardt, T. E. e Silveira, D. T., editors (2009). *Métodos de Pesquisa*. Editora da UFRGS, 1 edition. ISBN: 978-85-386-0071-8.

[Glass et al. 2000] Glass, S., Hiller, T., Jacobs, S., e Perkins, C. (2000). Mobile ip authentication, authorization, and accounting requirements - rfc 2977. Internet Engineering Task Force.

[GoGrid 2015] GoGrid (2015). Gogrid. http://www.gogrid.com. Acessado em: Outubro, 2015.

[Gonzalez et al. 2007] Gonzalez, J. M., Paxson, V., e Weaver, N. (2007). Shunting: a hardware/software architecture for flexible, high-performance network intrusion prevention. In *Proceedings of the 14th ACM conference on Computer and communications security*, CCS '07, pages 139–149, New York, NY, USA. ACM.

[Goudarzi et al. 2012] Goudarzi, H., Ghasemazar, M., e Pedram, M. (2012). SLA-based Optimization of Power and Migration Cost in Cloud Computing. *2012 12th IEEE/ACM*

International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), pages 172–179.

[Goyal 2009] Goyal, P. (2009). The Virtual Business Services Fabric: An Integrated Abstraction of Services and Computing Infrastructure. *2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises*, pages 33–38.

[Grembergen e Haes 2010] Grembergen, W. V. e Haes, S. D. (2010). A research journey into enterprise governance of it, business/it alignment and value creation. *International Journal of IT/Business Alignment and Governance (IJITBAG)*, 1(1):1–13.

[Grossman 2009] Grossman, R. (2009). The Case for Cloud Computing. *IT Professional*, 11(2):23–27.

[Habib et al. 2003] Habib, A., Fahmy, S., Avasarala, S. R., Prabhakar, V., e Bhargava, B. (2003). On detecting service violations and bandwidth theft in qos network domains. *Computer Communications*, 26(8):861 – 871. Performance evaluation of {IP} networks and services.

[Hadoop] Hadoop. Apache hadoop. hadoop.apache.org. Acessado em: Novembro, 2015.

[Handley et al. 2001] Handley, M., Paxson, V., e Kreibich, C. (2001). Network intrusion detection: evasion, traffic normalization, and end-to-end protocol semantics. In *Proceedings of the 10th conference on USENIX Security Symposium - Volume 10*, SSYM'01, pages 9–9, Berkeley, CA, USA. USENIX Association.

[Hping 2014] Hping (2014). Hping 3.

[Hurwitz et al. 2013] Hurwitz, J., Nugent, A., Halper, F., e Kaufman, M. (2013). *Big Data For Dummies*. For Dummies, 1st edition.

[Husson et al. 2010] Husson, F., LÃª, S., e PagÃ¨s, J. (2010). *Exploratory Multivariate Analysis by Example Using R*. Chapman & Hall/CRC. ISBN 978-1-4398-3580-7.

[IBM 2013] IBM (2013). Ibm infrastructure as a service. http://www-935.ibm.com/services/us/en/cloud-enterprise. Acessado em: Agosto, 2013.

[IBM 2015] IBM (2015). Iaas cloud computing platform guide for managers. http://searchcio.techtarget.in/tutorial/IaaS-cloud-computing-platform-guide-for-managers. Acessado em: Outubro, 2015.

[Ishibashi et al. 2007] Ishibashi, K., Kawahara, R., Tatsuya, M., Kondoh, T., e Asano, S. (2007). Effect of sampling rate and monitoring granularity on anomaly detectability. In *IEEE Global Internet Symposium, 2007*, pages 25 –30.

[Jacob e Brodley 2006] Jacob, N. e Brodley, C. (2006). Offloading ids computation to the gpu. In *Computer Security Applications Conference, 2006. ACSAC '06. 22nd Annual*, pages 371 –380.

[Jain 1991] Jain, R. (1991). *The Art of Computer Systems Performance Analysis: techniques for experimental design, measurement, simulation, and modeling*. Wiley.

[Jones et al. 2015] Jones, E., Oliphant, T., Peterson, P., et al. (2015). SciPy: Open source scientific tools for Python. [Online; Acessado em 2015-10-23].

[Justafort e Pierre 2012] Justafort, V. e Pierre, S. (2012). Performance-aware virtual machine allocation approach in an intercloud environment. In *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pages 1–4.

[Kamiyama e Mori 2006] Kamiyama, N. e Mori, T. (2006). Simple and accurate identification of high-rate flows by packet sampling. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, pages 1 –13.

[Kantarci et al. 2012] Kantarci, B., Foschini, L., Corradi, A., e Mouftah, H. (2012). Inter-and-intra data center vm-placement for energy-efficient large-scale cloud systems. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 708–713.

[Katashita et al. 2007] Katashita, T., Yamaguchi, Y., Maeda, A., e Toda, K. (2007). Fpga-based intrusion detection system for 10 gigabit ethernet. *IEICE - Trans. Inf. Syst.*, E90-D(12):1923–1931.

[Khatri e Brown 2010] Khatri, V. e Brown, C. V. (2010). Designing data governance. *Commun. ACM*, 53(1):148–152.

[Lacerda et al. 2009] Lacerda, T., Fernandes, S. L., Oliveira, A. C., Sadok, D., e Kelner, J. (2009). Performance-driven development of deep packet inspection systems on commodity platforms. In *VIII Workshop em Desempenho de Sistemas Computacionais e de Comunicaćão. In: , Anais do XXIX Congresso Brasileiro de Computação*, Bento Gonçalves, RS.

[Lakhina et al. 2004] Lakhina, A., Crovella, M., e Diot, C. (2004). Characterization of network-wide anomalies in traffic flows. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, IMC '04, pages 201–206, New York, NY, USA. ACM.

[Laplante et al. 2008] Laplante, P., Zhang, J., e Voas, J. (2008). What's in a name? distinguishing between saas and soa. *IT Professional*, 10(3):46 –50.

[Lê et al. 2008] Lê, S., Josse, J., e Husson, F. (2008). Factominer: An r package for multivariate analysis. *Journal of Statistical Software*, 25(1).

[Lehtonen e Pahkinen 2004] Lehtonen, R. e Pahkinen, E. (2004). *Practical Methods for Design and Analysis of Complex Surveys*. John Wiley and Sons, England, 2nd edition edition. ISBN: 978-0-470-84769-5.

[Li et al. 2012] Li, J., Wang, Q., Jayasinghe, D., Malkowski, S., Xiong, P., Pu, C., Kanemasa, Y., e Kawaba, M. (2012). Profit-based experimental analysis of iaas cloud performance: Impact of software resource allocation. In *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, pages 344–351.

[Mai et al. 2006] Mai, J., Chuah, C.-N., Sridharan, A., Ye, T., e Zang, H. (2006). Is sampled data sufficient for anomaly detection? In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 165–176, New York, NY, USA. ACM.

[Mendes 2013] Mendes, C. M. M. (2013). *A Method Based on DEMO for Managing Service Quality*. PhD thesis, Technical University of Lisbon.

[Microsoft 2013] Microsoft (2013). Windows azure.

http://www.microsoft.com/industry/government/guides/cloud_computing/5-PaaS.aspx. Acessado em: Agosto, 2013.

[Mikkilineni e Sarathy 2009] Mikkilineni, R. e Sarathy, V. (2009). Cloud computing and the lessons from the past. In *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE '09. 18th IEEE International Workshops on*, pages 57 –62.

[Moura et al. 2015] Moura, J. A. B., Dóra, P., e Oliveira, A. C. (2015). Selecting frameworks for multi-agent systems development for the oil industry. *Revista Cubana de Ciencias Informáticas*, 9(1):78–93.

[Ng 2014] Ng, A. (2014). Machine learning: Anomaly detection. Lecture Notes, Coursera Course, Standford University.

[NumPy 2015] NumPy (2015). Numpy web site. http://www.numpy.org. Acessado em: Outubro, 2015.

[Nychis et al. 2008] Nychis, G., Sekar, V., Andersen, D. G., Kim, H., e Zhang, H. (2008). An empirical evaluation of entropy-based traffic anomaly detection. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, pages 151–156, New York, NY, USA. ACM.

[Oliveira et al. 2014a] Oliveira, A. C., Chagas, H., Spohn, M., Gomes, R., e Duarte, B. J. (2014a). Efficient network service level agreement monitoring for cloud computing systems. In *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pages 1–6.

[Oliveira et al. 2015a] Oliveira, A. C., Dora, P., Spohn, M., e Oliveira, K. (2015a). From the dark net to the cloudy data: Cloud network governance guidelines. In *XXXIV International Conference of the Chilean Society of Computer Science (SCCC)*, Santiago, Chile.

[Oliveira e Ferreira 2012] Oliveira, A. C. e Ferreira, A. (2012). Implementação de um detector de anomalias de tráfego de rede baseado na entropia de métricas para sistemas de computação em nuvem. In *VII Congresso Norte-Nordeste de Pesquisa e Inovação (CONNEPI)*.

[Oliveira et al. 2015b] Oliveira, A. C., Fetzer, C., Martin, A., e Spohn, M. (2015b). Optimizing query prices for data-as-a-service. In *Big Data (BigData Congress), 2015 IEEE International Congress on*, pages 289–296.

[Oliveira et al. 2015c] Oliveira, A. C., Fetzer, C., Martin, A., e Spohn, M. (2015c). Optimizing virtual machine scheduling for data-as-a-service. *International Journal of BigData (IJBD)*. Artigo convidado para publicação.

[Oliveira et al. 2014b] Oliveira, A. C., Spohn, M., Gomes, R., Quoc, D. L., e Duarte, B. J. (2014b). Improving network traffic anomaly detection for cloud computing services. In *9th International Conference on Systems and Networks Communications (ICSNC)*.

[Openstack 2015] Openstack (2015). *Openstack Training Guidelines Icehouse*. Openstack Foundation. http://docs.openstack.org/icehouse/training-guides/content/operator-computer-node.html.

[Oppenheimer 2004] Oppenheimer, P. (2004). *Top-Down Network Design*. Cisco Press, 2 ed. edition.

[Oracle 2015] Oracle (2015). Oracle cloud platform. http://www.oracle.com/us/solutions/cloud/platform/overview/index.html. Acessado em: Outubro, 2015.

[Ostinato 2015] Ostinato (2015). Ostinato: Packet/traffic generator and analyzer. http://code.google.com/p/ostinato. Acessado em: Outubro, 2015.

[Oumina e Ranc 2007] Oumina, H. e Ranc, D. (2007). Towards a real time charging framework for complex applications in 3gpp ip multimedia system (ims) environment. In *Next Generation Mobile Applications, Services and Technologies, 2007. NGMAST '07. The 2007 International Conference on*, pages 145 –150.

[Parkhill 1966] Parkhill, D. (1966). *The Challenge of the Computer Utility*. Addison-Wesley Educational Publishers Inc, US.

[Patel e Sarje 2012] Patel, K. S. e Sarje, A. (2012). Vm provisioning method to improve the profit and sla violation of cloud service providers. In *Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference on*, pages 1–5.

[Paxson et al. 2006] Paxson, V., Asanović, K., Dharmapurikar, S., Lockwood, J., Pang, R., Sommer, R., e Weaver, N. (2006). Rethinking hardware support for network analysis and intrusion prevention. In *Proceedings of the 1st USENIX Workshop on Hot Topics in Security*, HOTSEC'06, pages 11–11, Berkeley, CA, USA. USENIX Association.

[Paxson et al. 2007] Paxson, V., Sommer, R., e Weaver, N. (2007). An architecture for exploiting multi-core processors to parallelize network intrusion prevention. In *Sarnoff Symposium, 2007 IEEE*, pages 1 –7.

[PF_RING 2015] PF_RING (2015). Pf_ring: High-speed packet capture, filtering and analysis. http://www.ntop.org/products/pf_ring. Acessado em: Outubro, 2015.

[Quan et al. 2009] Quan, Q., Hong-Yi, C., e Rui, Z. (2009). Entropy based method for network anomaly detection. In *Dependable Computing, 2009. PRDC '09. 15th IEEE Pacific Rim International Symposium on*, pages 189–191.

[Quoc et al. 2015] Quoc, D. L., Fetzer, C., Fellber, P., Rivière, É., Schiavoni, V., e Sutra, P. (2015). Unicrawl: A practical geographically distributed web crawler. In *8th IEEE International Conference on Cloud Computing (CLOUD'15)*. IEEE Computer Society.

[Quoc et al. 2011] Quoc, D. L., Jeong, T., Roman, H. E., e Hong, J. W.-K. (2011). Traffic dispersion graph based anomaly detection. In *Proceedings of the Second Symposium on Information and Communication Technology*, SoICT '11, pages 36–41, New York, NY, USA. ACM.

[Quoc et al. 2014] Quoc, D. L., Yazdanov, L., e Fetzer, C. (2014). Dolen: User-side multi-cloud application monitoring. In *Future Internet of Things and Cloud*. IEEE.

[R Core Team 2015] R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[RabbitMQ 2015] RabbitMQ (2015). Rabbitmq: Advanced message queueing protocol. https://www.rabbitmq.com. Acessado em: Outubro, 2015.

[Rackspace 2015] Rackspace (2015). Rackspace. http://www.rackspace.com. Acessado em: Outubro, 2015.

[Rademakers 2012] Rademakers, T. (2012). *Activiti in Action: Executable business processes in BPMN 2.0*. Manning Publications, 1 edition. ISBN-13: 978-1617290121. ISBN-10: 1617290122.

[Rijsbergen 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition.

[RNP 2010] RNP (2010). Centro de pesquisa e desenvolvimento em tecnologias digitais para informação e comunicação (ctic): Projeto jit cloud (in portuguese). http://www.ctic.rnp.br/web/ctic/jitcloud. Acessado em: Outubro, 2015.

[Roesch 1999] Roesch, M. (1999). Snort - lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX conference on System administration*, LISA '99, pages 229–238, Berkeley, CA, USA. USENIX Association.

[Salesforce 2015] Salesforce (2015). Paas: Platform as a service. https://www.salesforce.com/paas/overview/. Acessado em: Outubro, 2015.

[Salfner et al. 2010] Salfner, F., Lenk, M., e Malek, M. (2010). A survey of online failure prediction methods. *ACM Computing Surveys*, 42(3):1–42.

[Schwarzkopf et al. 2013] Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M., e Wilkes, J. (2013). Omega: Flexible, scalable schedulers for large compute clusters. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys'13, pages 351–364, New York, NY, USA. ACM.

[SciPy 2015] SciPy (2015). Scipy web site. http://scipy.org. Acessado em: Outubro, 2015.

[Serrano et al. 2013] Serrano, D., Bouchenak, S., Kouki, Y., Ledoux, T., Lejeune, J., Sopena, J., Arantes, L., e Sens, P. (2013). Towards qos-oriented sla guarantees for online cloud services. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 50–57.

[Shetty 2013] Shetty, S. (2013). Auditing and analysis of network traffic in cloud environment. In *Proceedings of the 2013 IEEE Ninth World Congress on Services*, SERVICES '13, pages 260–267, Washington, DC, USA. IEEE Computer Society.

[Smith et al. 2010] Smith, D., Guan, Q., e Fu, S. (2010). An Anomaly Detection Framework for Autonomic Management of Compute Cloud Systems. *2010 IEEE 34th Annual Computer Software and Applications Conference Workshops*, pages 376–381.

[Snort 2013] Snort (2013). Snort intrusion detection and prevention system.

[Sommer et al. 2009] Sommer, R., Paxson, V., e Weaver, N. (2009). An architecture for exploiting multi-core processors to parallelize network intrusion prevention. *Concurr. Comput. : Pract. Exper.*, 21(10):1255–1279.

[Telecom 2007] Telecom, N. (2007). Service level management white paper. http://www.nexustelecom.com/documents/whitepapers.

[Travassos 2002] Travassos, G. (2002). *Introdução à engenharia de software experimental*. RT-ES-590/02. UFRJ.

[Vallentin et al. 2007] Vallentin, M., Sommer, R., Lee, J., Leres, C., Paxson, V., e Tierney, B. (2007). The nids cluster: scalable, stateful network intrusion detection on commodity hardware. In *Proceedings of the 10th international conference on Recent advances in intrusion detection*, RAID'07, pages 107–126, Berlin, Heidelberg. Springer-Verlag.

[van der Walt et al. 2011] van der Walt, S., Colbert, S. C., e Varoquaux, G. (2011). The numpy array: A structure for efficient numerical computation. *Computing in Science Engineering*, 13(2):22–30.

[Voas e Zhang 2009] Voas, J. e Zhang, J. (2009). Cloud computing: New wine or just a new bottle? *IT Professional*, 11(2):15 –17.

[Wang 2009] Wang, C. (2009). Ebat: online methods for detecting utility cloud anomalies. In *Proceedings of the 6th Middleware Doctoral Symposium*, MDS '09, pages 4:1–4:6, New York, NY, USA. ACM.

[Wang et al. 2010] Wang, C., Talwar, V., Schwan, K., e Ranganathan, P. (2010). Online detection of utility cloud anomalies using metric distributions. In *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, pages 96–103. Ieee.

[Wang et al. 2009] Wang, S. C., Yan, K. Q., e Wang, S. S. (2009). Achieving High Efficient Agreement with Malicious Faulty Nodes on a Cloud Computing Environment. *Industrial Engineering*, pages 3–8.

[Weaver et al. 2007] Weaver, N., Paxson, V., e Gonzalez, J. M. (2007). The shunt: an fpga-based accelerator for network intrusion prevention. In *Proceedings of the 2007 ACM/-SIGDA 15th international symposium on Field programmable gate arrays*, FPGA '07, pages 199–206, New York, NY, USA. ACM.

[Weinhardt et al. 2009] Weinhardt, C., Anandasivam, A., Blau, B., e Stöβer, J. (2009). Business models in the service world. *IT Professional*, 11(2):28–33.

[Werner e Jones 2003] Werner, M. L. e Jones, K. H. (2003). *Introduction to Management Accounting: a user perspective*. Pearson Prentice Hall, 2nd edition edition.

[Xi et al. 2009] Xi, L., Zhang, F., e Wang, D. (2009). *Artificial Intelligence and Computational Intelligence: International Conference, AICI 2009, Shanghai, China, November 7-8, 2009. Proceedings*, chapter Optimization of Real-Valued Self Set for Anomaly Detection Using Gaussian Distribution, pages 112–120. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Xiong et al. 2011a] Xiong, P., Chi, Y., Zhu, S., Moon, H. J., Pu, C., e Hacigumus, H. (2011a). Intelligent management of virtualized resources for database systems in cloud environment. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 87–98.

[Xiong et al. 2011b] Xiong, P., Chi, Y., Zhu, S., Tatemura, J., Pu, C., e HacigümüŞ, H. (2011b). Activesla: A profit-oriented admission control framework for database-as-a-service providers. In *Proceedings of the 2Nd ACM Symposium on Cloud Computing*, SOCC '11, pages 15:1–15:14, New York, NY, USA. ACM.

[Zaman e Grosu 2013] Zaman, S. e Grosu, D. (2013). A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds. *Cloud Computing, IEEE Transactions on*, PP(99):1–1.

[Zhang e Zhou 2009] Zhang, L.-J. e Zhou, Q. (2009). CCOA: Cloud Computing Open Architecture. *2009 IEEE International Conference on Web Services*, pages 607–616.

[Zhang et al. 2010] Zhang, Q., Cheng, L., e Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1(1):7–18.

[Ziviani e Duarte 2005] Ziviani, A. e Duarte, O. C. M. B. (2005). Metrologia na internet. In *Minicurso no Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC'2005)*.

Apêndice A

Convenções Adotadas

"Pois a dúvida agrada-me não menos que o prazer."

Dante

Análise de tráfego *off-line*: a análise de tráfego *off-line* não é feita em tempo real. O tráfego de rede é armazenado em arquivos de rastro (*traces*) e em algum momento será analisado.

Análise de tráfego *online*: a análise de tráfego *online* exige alto poder de memória e processamento, pois verifica a presença de anomalias no tráfego de rede que está transitando no ponto de coleta em tempo real.

Anomalia (ou anomalia de tráfego): uma anomalia de tráfego neste trabalho representa uma alteração não esperada nas métricas de desempenho de rede, como tráfego indesejado gerado por um ataque de negação de serviço. Pode ser entendida também como um descumprimento no acordo de nível de serviço (SLA) para metas técnicas relacionadas ao tráfego de rede, como alto atraso dos pacotes.

***DC Provider*:** um centro de dados.

Tráfego de rede: neste contexto, refere-se a todos os pacotes que são produzidos e enviados por serviços de um *DC Provider* que está sendo monitorado para um sorvedouro

local ou remoto, no contexto de *DC Provider*, e os pacotes contendo dados ou informações de controle que são destinado a máquinas virtuais ou reais pertencentes ao *DC Provider*.

Ponto de coleta de tráfego: refere-se ao local em que o tráfego de rede está sendo monitorado e analisado. Geralmente, é representado por uma porta de um roteador ou switch capaz de fazer o espelhamento de todo o tráfego de entrada e saída das demais portas. A investigação do tráfego apenas observa os pacotes que estão transitando no(s) enlace(s) que está(ão) sendo monitorado(s), o que significa que o mecanismo de detecção e análise de tráfego não injeta pacotes na rede. Os serviços que podem ser identificados em um ponto de coleta dependem de como é a topologia lógica da rede. Por exemplo, se um ponto de coleta for instalado em um roteador por onde passa tráfego de uma mais de um *DC Provider*, então ele é capaz de fornecer informações para detectar anomalias em mais de um DC Provider. Neste trabalho, consideramos um detector de anomalias por *DC Provider*, contudo, dependendo de como está estruturada a topologia da rede de computadores, essa análise pode ser estendida, mesmo que seja aplicado um único ponto de coleta. Outra consideração importante é que a implantação do mecanismo de detecção de anomalias de tráfego é opcional para um *DC Provider*, podendo, portanto, haver provedores que não realizam este tipo de monitoramento.

Métrica (ou métrica de um SLA): para o detector de anomalias, uma métrica representa uma meta técnica de rede que foi acordada entre o cliente e o provedor de serviços de computação em nuvem em um SLA. Exemplos: atraso, variação do atraso, largura de banda, tempo de resposta, disponibilidade, utilização, precisão e carga. Para cada métrica, será definida uma faixa de valores permitida ou um valor limiar Quando é estabelecido um valor esperado para a métrica, então esse par é chamado de SLO. Qualquer valor observado no tráfego que não esteja em conformidade com os limites estabelecidos no acordo, é considerado como uma anomalia de tráfego.

Apêndice B

Caracterização da Utilização de Serviços de Computação em Nuvem

> "O que perturba os homens não são as coisas que acontecem, mas sim a opinião que eles têm delas."
>
> Demócrito (461-361 a.C)

Neste trabalho foi realizado um estudo piloto (*survey*) para investigar demandas reais dos clientes de computação em nuvem, a fim de fornecer informações úteis para a especificação de novos modelos e de apoio à tomada de decisão. Os resultados obtidos reforçam a forte dependência dos serviços em nuvem no nível de rede, constatação esta que motivou a elaboração de uma proposta de uma abordagem para a governança da rede na nuvem, avançando assim o estado-da-prática.

É difícil obter informações precisas referentes a SLAs acordados entre clientes e provedores; tanto por muitos clientes desconhecerem os acordos, como por questões de sigilo de informações. No entanto, este levantamento serviu para que fossem identificadas as principais métricas acordadas em SLAs e que devem nortear a governança de rede em sistemas de computação em nuvem.

B.1 Objetivos da Pesquisa

O objetivo geral desta pesquisa é caracterizar os tipos de serviços de computação em nuvem que os clientes utilizam e os níveis de desempenho esperados para os serviços, quando aplicáveis. Os objetivos específicos dessa pesquisa são:

1. Caracterizar o perfil dos clientes dos serviços em nuvem;
2. Caracterizar a utilização de serviços em nuvem;
3. Caracterizar requisitos técnicos para serviços em nuvem estabelecidos nos SLAs;
4. Identificar quais, se houver, abordagens de monitoramento estão sendo empregadas pelos provedores e oferecidas aos clientes;
5. Caracterizar os níveis de aceitação dos serviços de computação em nuvem por parte dos clientes.

B.2 Projeto Experimental

Como a população total de usuários de computação em nuvem e provedores é difícil de precisar, optou-se por realizar um estudo piloto usando um *survey* (questionário) não-supervisionado. O processo de amostragem foi não-probabilístico com base em grupos focais (*focus groups*), onde as pessoas inicialmente selecionadas foram contatos pessoais ou pessoas pertencentes a listas de *e-mail* em comum, permitindo uma amostragem em bola de neve (*snowball sampling*), em que os participantes podem indicar o instrumento de pesquisa para seus próprios contatos pessoais [Lehtonen e Pahkinen 2004]. Para criar, divulgar o questionário via *e-mail*, controlar o número de respostas e administrar os resultados da pesquisa, utilizou-se uma aplicação de computação em nuvem do *Google*, denominada *Google Forms*.

O instrumento de pesquisa foi um questionário qualitativo. O processo de desenvolvimento e aplicação da pesquisa seguiu um processo experimental de quatro etapas:

1. Definição de objetivos;

2. Planejamento de experimentos;

3. Execução do experimento e coleta de dados (estatística descritiva);

4. Análise e interpretação dos dados coletados (estatística inferencial).

A pesquisa realizada para caracterização de utilização de serviços de computação em nuvem compreendeu os seguintes passos:

1. Identificação dos serviços de computação em nuvem utilizados na perspectiva dos clientes;

2. Identificação de metas com base na metodologia GQM, onde cada problema representado por uma questão é identificado em termos de objetivos;

3. Identificação de questões que abrangem todos os objetivos;

4. Identificação de métricas para cada pergunta (a questão pode envolver mais de uma métrica);

5. Desenvolvimento de um formulário *online* com perguntas e seu envio a possíveis respondentes;

6. Tabulação das respostas;

7. Análise dos dados;

8. Proposta de métricas para o desenho de governança da rede da nuvem;

9. Identificação de indicadores de qualidade de serviço para monitoramento de nuvem de acordo com as respostas obtidas;

10. Avaliação dos resultados.

As perguntas do questionário foram mapeadas dentro das metas propostas. Inicialmente, realiza-se a definição das metas que norteiam a criação das questões e, em seguida, a definição de pelo menos uma métrica de avaliação para cada questão, seguindo a metodologia GQM [Basili et al. 2002]. A Tabela I contém a representação das metas (G - *Goal*), questões (Q) e métricas (M) desse estudo.

Tabela B.1: Metas, questões e métricas da pesquisa de caracterização de serviços de computação em nuvem.

Metas	**Questões**	**Métricas**
Meta 1:	**Questão 1:** A que tipo de empresa você está afiliado (ser o mais específico possível)?	$M_{1,i}$ – % dos clientes filiado ao tipo de empresa i
Caracterizar o perfil dos clientes dos serviços em nuvem	**Questão 2:** Qual é a área de atuação de sua empresa?	$M_{2,i}$ – % dos clientes na área de especialização i
Meta 2:	**Questão 3:** Escolha o serviço (s) que melhor atender às suas necessidades.	$M_{3,i}$ – % dos clientes que precisam do serviço i
Caracterizar a utilização de serviços em nuvem	**Questão 4:** Você precisa de algum serviço de nuvem que não está disponível de graça?	$M_{4,i}$ – % dos clientes que não precisam de serviços pagos, para $i = 0$; % dos clientes que precisam de serviços pagos, para $i = 1$.
	Questão 5: Que tipo de serviços que você utiliza?	$M_{5,i}$ – % dos clientes que usam o modelo de negócio i
Meta 3:	**Questão 6:** Você tem algum contrato para usar serviços de computação em nuvem?	$M_{6,i}$ – % dos clientes que não têm um contrato, para $i = 0$; % dos clientes que têm um contrato, para $i = 1$
Caracterizar SLAs	**Questão 7:** Se você paga por serviços na nuvem, como Amazon EC2 ou o Windows Azure, você conhece suas cláusulas de contrato de serviço?	$M_{7,i}$ – % dos clientes que não conhecem seus SLAs, para $i = 0$; % dos clientes que conhecem seus SLAs, para $i = 1$
	Questão 8: Se respondeu "sim" à questão anterior, quais tipos de restrições estão contidas em contrato de serviço? [Múltipla escolha]	$M_{8,i}$ – % dos clientes que contrataram a restrição de SLA i
Meta 4:	**Questão 9:** O seu provedor de computação em nuvem oferece algum sítio na Web onde você possa monitorar o desempenho do serviço contratado?	$M_{9,i}$ – % dos provedores que não oferecem um monitor de desempenho, para $i = 0$; % dos provedores que oferecem um monitor de desempenho, para $i = 1$
Caracterizar a abordagem de monitoramento do serviço disponível	**Questão 10:** Se você respondeu "sim" à pergunta anterior, esse sítio na Web mostra possíveis descumprimentos nos acordos de nível de serviço?	$M_{10,i}$ – % dos clientes que têm acesso a não-conformidade com SLA, se $i = 0$; % dos clientes que têm acesso a não-conformidade com SLA, se $i = 1$
Meta 5:	**Questão 11:** Quão satisfeito você está com o serviço de computação em nuvem que utiliza? Ele atende às suas necessidades técnicas?	$M_{11,i}$ – % dos clientes que estão satisfeitos conforme opção i
Caracterizar os níveis de aceitação dos serviços de computação em nuvem pelos clientes	**Questão 12:** Considerando o tempo, esforço e dinheiro que você gastou com o seu provedor de serviços em nuvem, como você avaliaria o custo-benefício geral?	$M_{12,i}$ – % dos clientes que expressam custo-benefício conforme opção i

B.3 Estatística Descritiva

Depois de remover os dados inconsistentes, obtiveram-se 67 respostas para 12 perguntas. Uma taxa de resposta de cinco pessoas para cada pergunta é aceitável para um estudo piloto. Após isso, as respostas do questionários foram codificadas, sendo mapeadas para números discretos, e analisadas de acordo com estratégia apropriada para o tipo da variável de resposta [Collingridge 2015]. Em termos de Estatística Descritiva, recomenda-se que o estudo das questões representadas por variáveis categóricas e ordinais seja feito com gráficos de pizza ou de barras e que sejam analisadas medidas de tendência central, tais como média, mediana e moda.

A Questão 1 caracterizou o tipo de filiação institucional dos respondentes (Figura B.1). Reuniram-se respostas de empresas de três setores da economia: educação, serviços e indústria. Os resultados para o tipo de instituição tiveram como *moda* as *empresas estatais*.

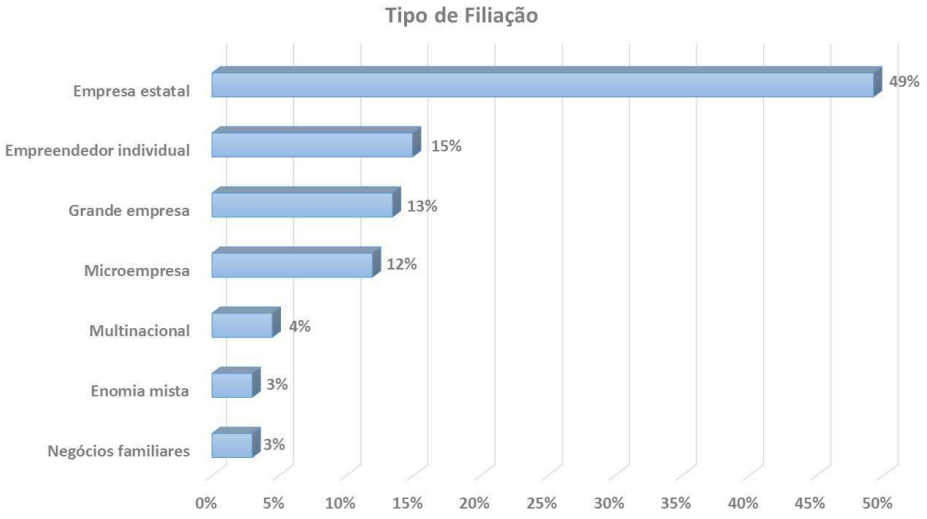


Figura B.1: Diagrama de Pareto com os resultados obtidos para a Questão 1.

Quanto à área de atuação das empresas/instituições, os usuários do setor de *educação* foram a *moda*, dentre eles se encontram as categorias de estudantes, pesquisadores, analistas e professores, que utilizam, em sua maioria, os serviços de computação em nuvem para fins acadêmicos. A classificação dos respondentes deu-se com 69% no setor educacional, seguido por 25% no setor de serviços e 6% na indústria, como representado na Figura B.2. O percentual de respondentes por cada instituição está detalhado na Figura B.3.

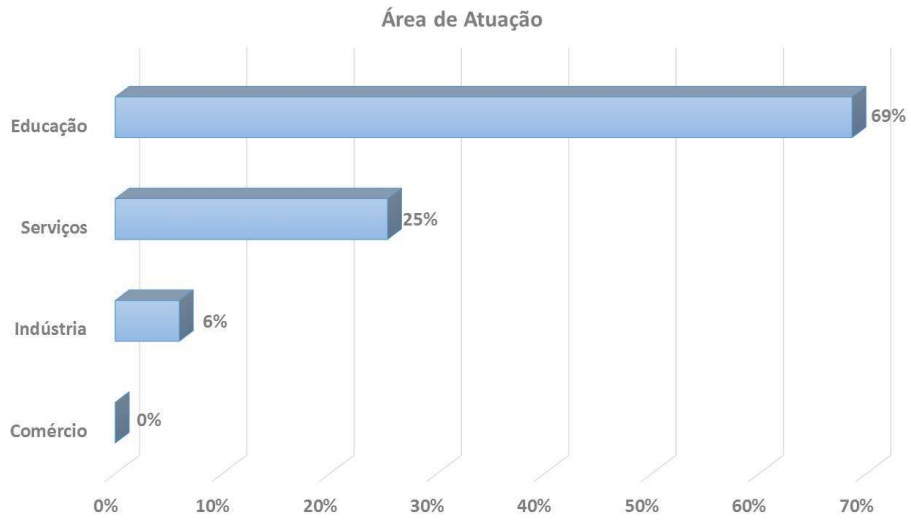


Figura B.2: Diagrama de Pareto com os resultados obtidos para a Questão 2.

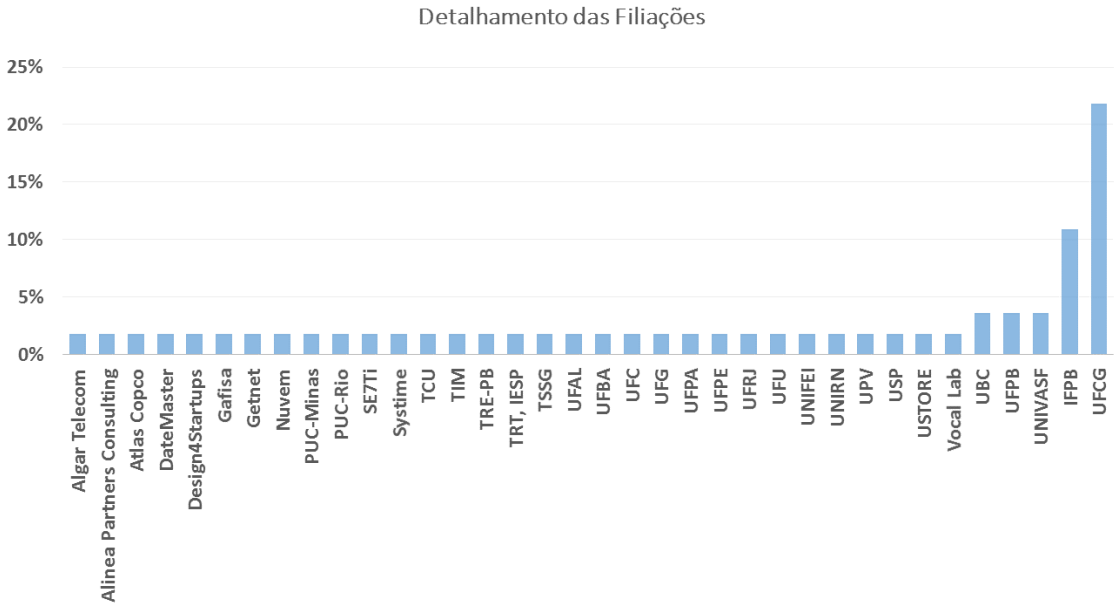


Figura B.3: Diagrama de Pareto com detalhamento do percentual de respondentes por instituição.

Dentre os aplicativos gratuitos mais utilizados estão os disponibilizados pela empresa Google, como Docs, YouTube e Picasa, que são utilizados por 90% dos respondentes. Amazon Elastic Computing, que fornece uma infra-estrutura para aplicações de alto desempenho, é usado por 36% de todos os usuários. Dentre os respondentes, 46% declararam que usaram outros tipos de serviços em nuvem, como o Dropbox (45%), a Apple iCloud (5%) e Evernote (5%). Esses resultados estão apresentados usando diagramas de Pareto (gráficos de barras frequência ordenada) na Figura B.4. Observa-se que o total pode ultrapassar 100%, porque um entrevistado pode escolher mais de uma opção.

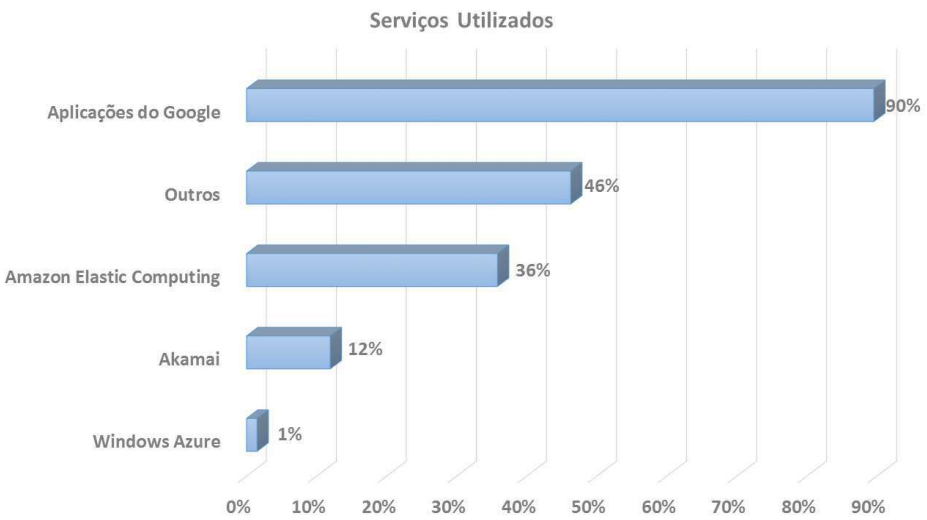


Figura B.4: Diagrama de Pareto com os resultados obtidos para a Questão 3.

Quanto a precisar de serviços pagos, a maioria dos respondentes declarou não precisar de serviços pagos (61%). No entanto, uma parte significativa dos clientes declarou que precisavam de serviços de nuvem pagos (39%) (Figura B.5).

Quanto ao modelo de negócio dos serviços utilizados, mais uma questão de múltipla escolha, 78% dos clientes fazem uso de *Software-as-a-Service* (SaaS). Em segundo lugar vem *Infrastructure-as-a-Service* (IaaS) com 69% dos clientes, *Platform-as-a-Service* (PaaS) usado por 40% deles e 3% dos clientes não sabiam como categorizar os serviços em nuvem, conforme resultados da Figura B.6.

No que toca a contratação de serviços de computação em nuvem, 34% do total de respondentes chegou a contratar serviços (mostrado na Figura B.7).

Os SLAs negociados entre os provedores de serviços em nuvem e os clientes podem

Figura B.5: Gráfico de pizza com os resultados obtidos para a Questão 4.

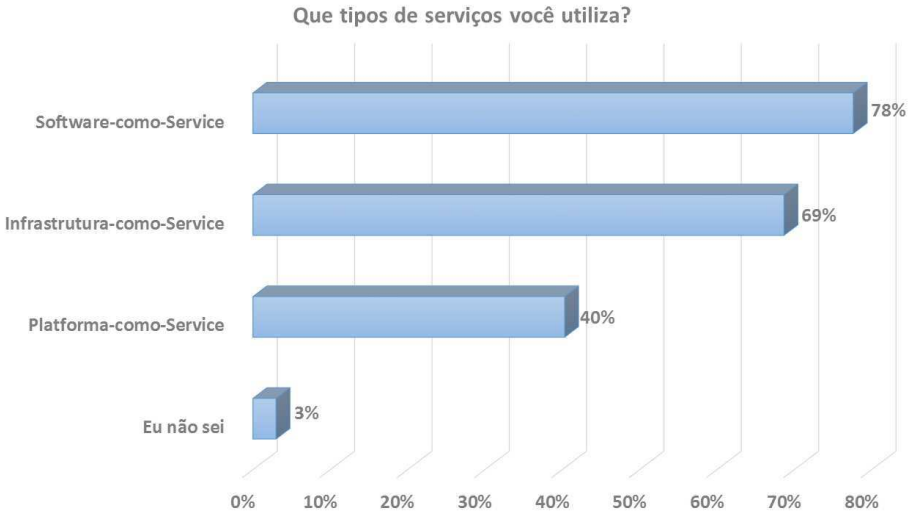


Figura B.6: Diagrama de Pareto com os resultados obtidos para a Questão 5.

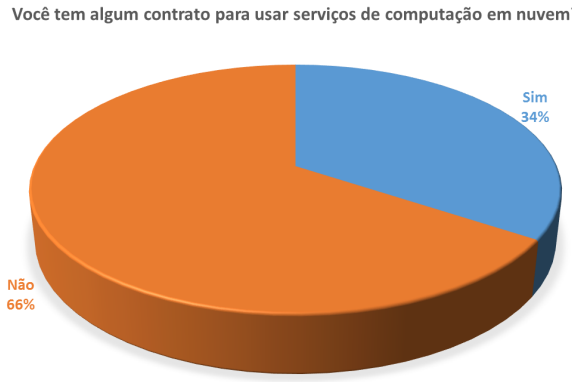


Figura B.7: Gráfico de pizza com os resultados obtidos para a Questão 6.

abranger diversos tipos de restrições, ou objetivos técnicos. Um percentual de 18% do total dos entrevistados declarou ter conhecimento das restrições estabelecidas em seus contratos de serviços de computação em nuvem (Figura B.8). Note que apenas 34% deles possuem algum tipo de contrato.

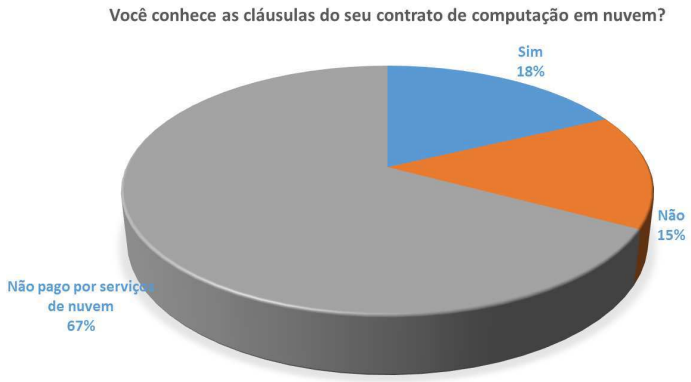


Figura B.8: Gráfico de pizza com os resultados obtidos para a Questão 7.

Dos 18% de clientes que declararam conhecer seus objetivos esperados de nível de serviço (SLOs), alguns forneceram informações adicionais (campo livre para comentários) sobre esses objetivos. Os requisitos de desempenho citados pelos clientes foram: largura de banda, atraso, variação do atraso (*jitter*), tempo de resposta, disponibilidade e tempo médio de reparo após uma falha. Esses requisitos não eram uma questão obrigatória; portanto, nem todos os entrevistados acrescentaram informações sobre SLOs.

Ao observar os tipos de SLOs esperados, a moda é a meta técnica de *disponibilidade* com valor de 18%, seguido por rede com 13% e serviço de atendimento ao cliente com 6%, conforme mostrado na Figura B.9. Essa questão também era uma questão de múltipla escolha.

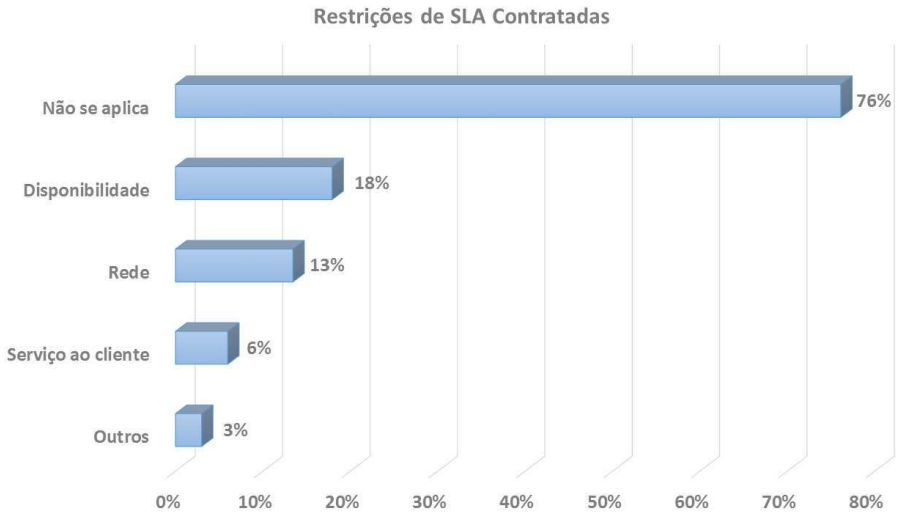


Figura B.9: Diagrama de Pareto com os resultados obtidos para a Questão 8.

Embora disponibilidade também possa ser considerada uma métrica do nível de aplicação, ela também pode ser classificada como uma métrica de rede. Tempo de resposta e segurança também podem ser monitorados no nível de rede. Problemas de usabilidade geralmente envolvem o nível de aplicação, tais como os requisitos relativos à interface gráfica do usuário; no entanto, esta exigência pode também se encaixar no nível da rede; por exemplo, o uso do protocolo DHCP (*Dynamic Host Configuration Protocol*) para configurar automaticamente os nomes das máquinas e endereços IP [Oppenheimer 2004]. Os tipos de restrições citados e os seus percentuais, com relação apenas ao número de respondentes que declararam conhecer os SLOs, estão dispostos na Tabela B.2. Um cliente pode selecionar mais de um requisito de SLA, portanto a soma dos percentuais declarados pode ser superior a 100%.

Dos respondentes, 55% declarou não dispor de qualquer aplicação para monitorar métricas de desempenho de seus serviços e 45% têm acesso a um sítio na Web que fornece algum tipo de informação relacionada ao desempenho (Figura B.10).

A pesquisa apontou que 85% dos prestadores não disponibilizam informações sobre o

Tabela B.2: Levantamento de metas técnicas acordadas em SLAs.

Métrica	**Percentual**
Disponibilidade	56%
Armazenamento	38%
Tempo de resposta	25%
Atraso	19%
Largura de banda	19%
Jitter	13%
Segurança	13%
Usabilidade	6%
Total	188%

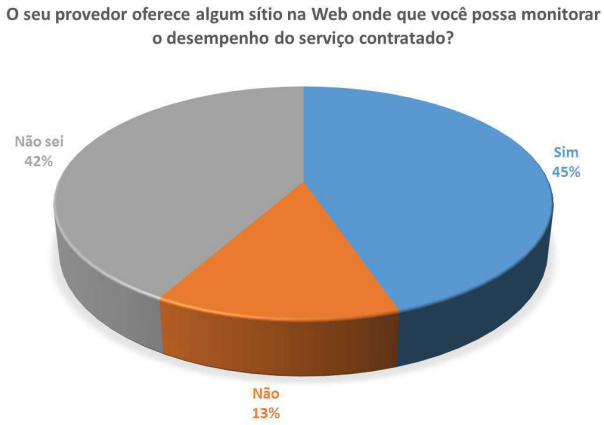


Figura B.10: Gráfico de pizza com os resultados obtidos para a Questão 9.

não cumprimento de acordos de serviço (Figura B.11).

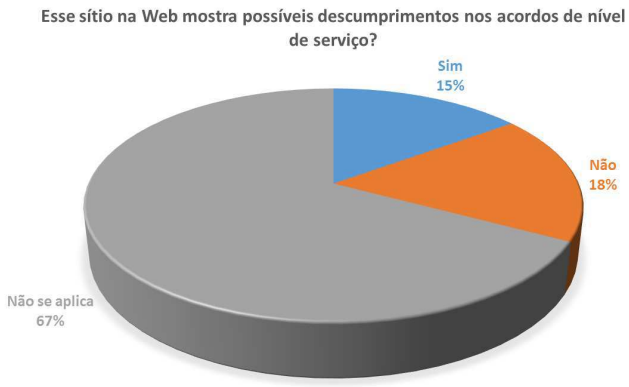


Figura B.11: Gráfico de pizza com os resultados obtidos para a Questão 10.

Quanto ao grau de satisfação dos clientes sobre o atendimento aos requisitos técnicos, a moda corresponde a 60% dos respondentes que afirmam estar muito satisfeitos (Figura B.12).

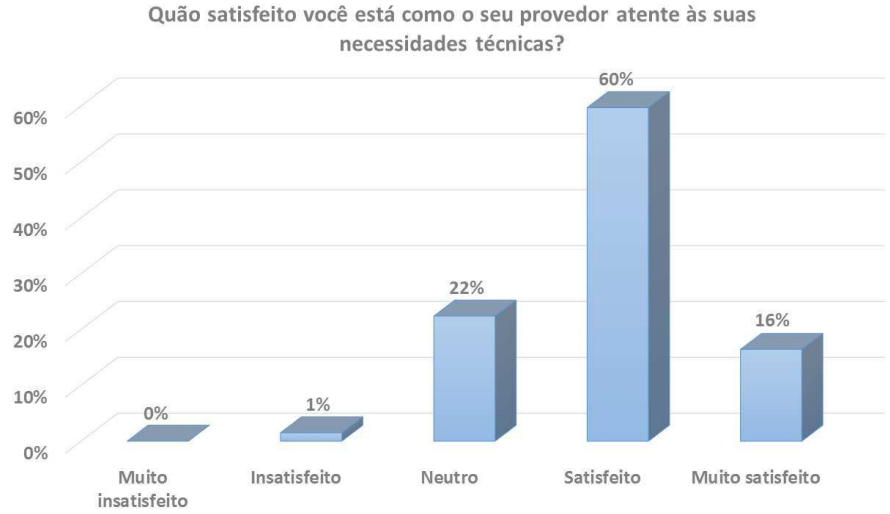


Figura B.12: Resultados obtidos utilizando Escala de Likert para a Questão 11.

A avaliação dos clientes quanto à combinação de tempo, esforço e dinheiro gasto com serviços de nuvem, a maioria dos clientes avaliaram os serviços como dentro do esperado (57%). Apenas 6% classificaram os serviços como pior do que o esperado (Figura B.13).

A Questão 12 continha uma caixa adicional para os clientes expressarem qualquer co-

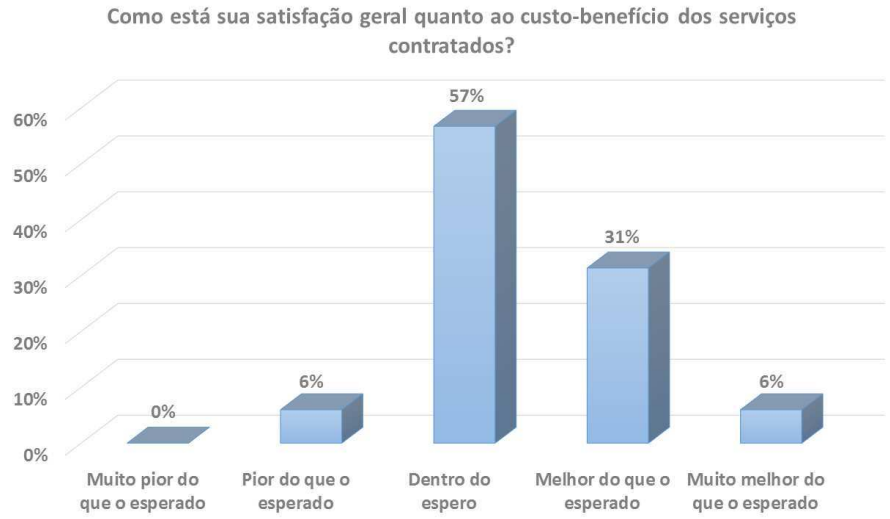


Figura B.13: Resultados obtidos utilizando Escala de Likert para a Questão 12.

mentário relacionado, o que foi feito por 12% deles. Os respondentes sugeriram que: os provedores não cobrassem taxas extras para largura de banda, sendo preferível restringir a taxa após certo montante de utilização; fatias de tempo com granularidade fina; serviço de banco de dados para aplicações específicas; possibilidade de aplicações de computação em nuvem privados (na rede local); serviços precisam ser tão transparentes quanto possível para que o cliente não precise se preocupar com questões de infraestrutura; a possibilidade de fazer configurações avançadas de maneira fácil para domínios e máquinas específicos; suporte para outros modos de pagamento como, por exemplo, para alguns países como o Brasil; aplicativos em execução em um navegador impõem limitações ao uso, portanto, aplicações em nuvem para PCs (*desktop*) precisam ter o seu próprio *software*, como em telefones celulares; existem problemas de compatibilidade entre os serviços, diferentes formatos de dados e protocolos; é difícil realizar *backup* local.

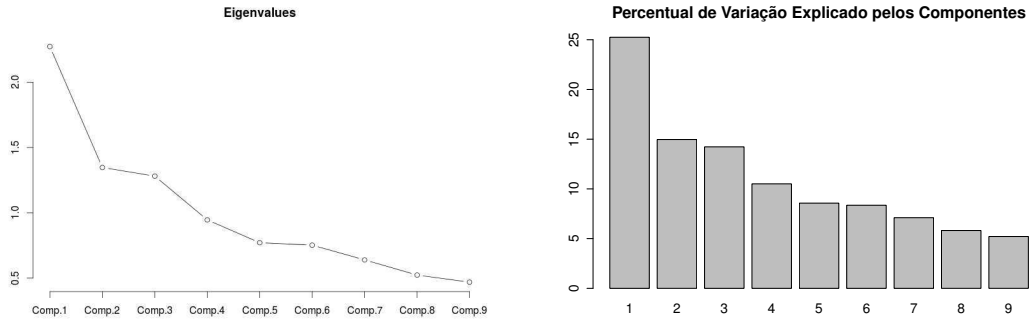
B.4 Estatística Inferencial

Devido ao grande número de variáveis a serem estudadas, realizou-se uma análise multivariada para capturar as correlações entre todas as respostas dadas às perguntas, para projetar as distâncias entre as correlações e para reduzir a dimensionalidade. A metodologia empregada para a análise multivariada foi a *Análise de Componentes Principais* (PCA), onde as ques-

tões são tratadas por *variáveis* e os respondentes são chamados de *indivíduos* [Husson et al. 2010] [Lê et al. 2008].

PCA estuda a tabela de dados normalizados e procura a melhor representação $2D$ para os dados por meio da decomposição de valores únicos (*Single Value Decomposition* – SVD), que consiste no estudo da distância entre 2 pontos visando minimizar a distorção [Lê et al. 2008].

Para reduzir a dimensionalidade dos dados, PCA busca os componentes que melhor explicam a variação dos resultados. Uma estatística que mede variância é chamada de *eigenvalues* (valores próprios). A regra é que devem ser retidos os componentes que têm *eigenvalues* superiores a 1, os demais podem ser removidos sem prejudicar o estudo. Os resultados obtidos neste estudo apontam três componentes principais, que explicam conjuntamente cerca de 55% da variabilidade das respostas. Os valores dos *eigenvalues* e dos percentuais com que cada componente contribui para a variação total dos resultados estão contidos na Tabela B.3. Gráficos complementares sobre esses resultados estão apresentados nas Figuras B.14a e B.14b.



(a) *Eigenvalues* (b) Variabilidade

Figura B.14: Variabilidade dos resultados.

O gráfico *biplot*, também chamado de *círculo de correlação*, é um gráfico tradicional na PCA que visa projetar a melhor representação das variáveis em um plano com duas dimensões. Todas as variáveis ativas do modelo são projetadas no gráfico, mostrando a correlação de cada uma delas com as dimensões. Essa projeção é feita por meio de vetores que representam os coeficientes de correlação entre as variáveis e as dimensões. Esses coeficientes de correlação das variáveis são chamados de *loadings*.

Tabela B.3: *Eigenvalues*, percentual de variância e percentual de variância acumulada

Componente	*eigenvalue*	% variância	% variância acumulada
Comp. 1	2,272535884	25,25039872	25,25039872
Comp. 2	1,346720969	14,96356633	40,21396504
Comp. 3	1,280563258	14,22848064	54,44244568
Comp. 4	0,945707122	10,50785691	64,95030259
Comp. 5	0,771127194	8,568079935	73,51838253
Comp. 6	0,75208797	8,356533003	81,87491553
Comp. 7	0,639272033	7,103022588	88,97793812
Comp. 8	0,523147808	5,812753418	94,79069154
Comp. 9	0,468837762	5,209308462	100

Neste estudo, o *biplot* indica que, se a satisfação geral do cliente é alta, então a satisfação do cliente com os requisitos técnicos também é alta; e que os clientes que contrataram serviços de nuvem tendem a não saber suas cláusulas de SLA. O círculo de correlação encontra-se na Figura B.15 e os respectivos *loadings* de cada variável para cada um dos nove componentes está na Tabela B.4.

Tabela B.4: *Loadings* de cada variável.

Variável	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
area.atuacao	0,3052	-0,3889	0,4586	-0,249	-0,1059	0,0729	-0,0019	-0,6812	-0,0526
existe.contrato	-0,4807	-0,0081	0,2073	-0,3456	-0,1426	-0,2014	0,1683	-2,00E-04	0,7192
tipo.empresa	0,1027	-0,21	0,7293	0,1802	-0,0328	-0,1529	-0,0533	0,5872	-0,094
satisfacao.tecnica	-0,2209	-0,4977	-0,2748	-0,106	-0,7164	0,0822	0,0348	0,1786	-0,2522
conhece.sla	0,4346	-0,1763	-0,2097	-0,3256	0,028	0,1058	-0,6415	0,2527	0,3778
monitoramento.web	0,3476	0,3039	-0,0551	0,2553	-0,4916	-0,6594	-0,0833	-0,1507	0,1117
servicos.pagos	-0,3668	0,2391	0,2292	0,3945	-0,2621	0,3799	-0,5802	-0,2132	0,0712
satisfacao.geral	0,0272	-0,5634	-0,1909	0,6595	0,1961	-0,0427	0,0918	-0,1011	0,3892
sla.web	0,4141	0,2424	0,0572	0,1144	-0,3233	0,576	0,4518	0,1306	0,3096

O componente que melhor explica a variabilidade, ou seja, que possui maior correlação com a primeira dimensão, é “existe.contrato”, que corresponde à Questão 7. Contudo, essa correlação alta é negativa, pois esta variável tem correlação de $-0,7247$ com a dimensão. A variável que possui a maior correlação (positiva) com a segunda dimensão é o “satisfa-

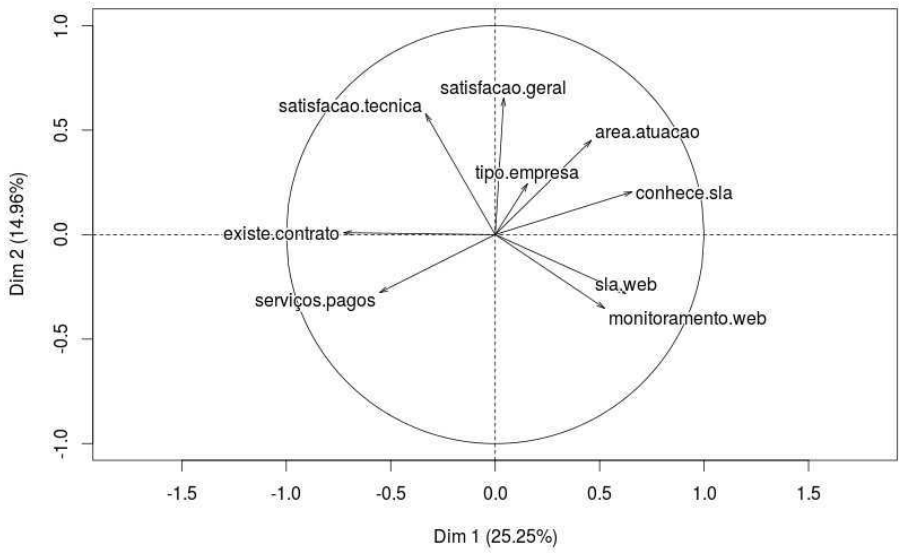


Figura B.15: Círculo de correlações.

cao.geral" (Questão 12), cujo valor é $0,6538$. Complementarmente, na terceira dimensão, a variável é o "tipo.empresa" (Questão 1) com correlação positiva de $0,8253$, mas não está representado no gráfico. Como conclusão, têm-se que as Questões 7, 12 e 1, respectivamente, melhor resumem os resultados do estudo, pois explicam a maior parte da variação das respostas.

Outra medida importante na PCA são os *scores*. Os *scores* são os dados do estudo normalizados (padronizados). Quando os dados, por exemplo, seguem uma distribuição normal, após serem padronizados eles se concentrarão em uma faixa entre -1.95 e 1.95. Se um determinado *score* for maior do que 2, por exemplo, ele é um dado extremamente maior do que a média. Se os *scores* forem aproximados, então os indivíduos possuem as mesmas características (em duas dimensões). Se os pontos estiverem distantes em uma dimensão, significa que são complementares, pois são diferentes em uma componente principal [Lê et al. 2008]. Os *scores* para as nove questões númericas estão apresentados na Tabela B.5.

Com base nos resultados obtidos com a PCA, serão analisados os dois componentes principais identificados. Os clientes serão divididos em categorias: os que pagaram pela contratação de serviços de nuvem ("Sim") e a dos que não o fizeram ("Não"). Será verificado se há diferença estatística sobre a satisfação geral dos clientes dessas duas categorias (com e

Tabela B.5: *Scores* de cada indivíduo.

Ind.	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
1	-0,7847	-1,1131	0,8951	-1,7022	0,5418	-1,3964	-0,6429	0,3037	0,177
2	1,6636	1,2954	-0,2549	-0,031	0,4497	-0,5208	0,04	0,4433	0,114
3	0,6049	-0,5506	-0,5856	-1,3666	0,7857	1,7315	0,3757	-0,165	-0,5425
4	-0,5808	-0,0563	1,2857	0,4217	-0,1482	0,4031	-0,742	-0,3073	-1,3051
5	1,0829	-0,3606	0,7178	1,4596	-0,9554	0,2624	-1,0171	-0,1648	0,3309
6	0,774	-1,0566	0,3336	1,3114	-1,9571	0,3774	-0,9684	0,0849	-0,0216
7	-1,3938	1,8508	-0,2951	0,5505	-1,2917	-0,2106	0,1445	0,4428	1,0133
8	1,2899	-0,4743	0,6053	-0,6808	0,0129	0,505	0,1613	0,6391	-0,5437
9	-0,9457	-0,2033	1,2582	0,86	-1,6024	1,1139	0,7526	-0,2315	-1,771
10	-1,6223	0,0579	-1,1153	-1,3639	0,8288	-1,3646	-0,5687	1,1061	0,4211
11	-0,3628	-0,6592	0,1688	-2,016	0,4711	1,2408	0,7068	0,0905	0,9313
12	-2,122	1,7935	0,096	0,1803	0,8403	-0,1042	-0,3622	0,2451	0,8063
13	0,5998	-1,2785	-1,1811	-0,4858	1,0852	1,7359	0,5325	-0,5679	0,0646
14	2,1048	0,6641	0,9092	-0,1609	0,2991	-0,57	-0,0087	0,17	-0,0285
15	-0,7867	0,7851	-0,6903	0,3987	0,1624	0,5112	-1,5577	1,0924	-0,5916
16	-0,7054	-2,5345	-1,538	1,3699	0,6549	-1,1496	-0,677	1,8298	-0,437
17	1,1321	0,0451	-0,8096	1,224	-0,1734	-0,6852	0,178	1,8427	0,3061
18	1,3295	-1,2937	0,3277	0,2785	0,2981	0,4429	0,2949	0,4921	0,0223
19	0,2774	-0,4914	1,0419	-1,4087	-0,2874	0,0809	0,5157	0,6387	0,971
20	-0,9416	-1,2625	1,3149	-0,0117	0,1756	1,7584	-0,42	0,2736	1,5204
21	-0,0417	-0,564	1,3588	0,9346	1,9397	1,0022	2,1007	-0,221	-0,8084
22	-0,2469	0,5081	1,8235	0,345	1,1546	1,9868	-0,0901	-0,17	-0,7181
23	1,7116	0,8791	0,4849	-1,3469	-0,9736	-0,3262	-0,0703	0,3108	-0,9062
24	-0,3906	0,4272	1,5283	-0,1575	-1,0788	-1,1648	2,1275	-0,2636	0,1129
25	1,8355	-0,8513	0,2474	0,6501	-0,4175	-0,5172	0,1736	0,2726	0,1849
26	0,4746	-0,745	0,5775	-1,1853	-2,0048	-0,7641	0,4431	0,669	0,781
27	1,3049	-0,037	-1,5523	0,6229	-0,2383	-0,3347	0,2688	0,0342	0,4094
28	-2,8587	0,9295	1,1023	0,4826	0,0588	1,3668	1,184	1,3078	0,0548
29	-1,5835	-0,4584	-0,8521	-1,3654	-1,3267	0,6365	-1,066	-1,2518	0,6132
30	0,8462	0,7125	1,4599	0,2769	0,4768	1,1696	-1,078	-0,048	-0,0451
31	-0,3678	-1,1696	-2,3045	0,0183	0,3133	1,6163	0,5624	1,5077	-0,2079
32	2,7622	1,2366	1,3999	1,0948	2,5877	-0,8622	0,0275	-0,4764	1,2427
33	-2,8411	0,6256	-0,8664	-0,5629	0,7674	-1,5837	1,1877	0,1674	-0,5586
34	-0,8343	1,6139	2,3829	0,8002	-0,6149	-0,5002	0,8882	-0,9507	0,6115
35	1,8355	-0,8513	0,2474	0,6501	-0,4175	-0,5172	0,1736	0,2726	0,1849
36	-0,3232	-1,4787	-0,1089	-1,0567	0,7563	1,1786	0,8404	-0,0565	1,4973
37	0,7164	1,4842	-1,3387	-0,2728	1,316	0,4884	0,21	0,9359	0,0939
38	-2,3694	-0,7076	-2,3344	2,7345	-0,2876	0,4826	0,5843	-0,6583	-0,6689
39	-1,3321	-1,6945	0,9008	-0,3032	0,7891	-1,6014	0,3573	-0,1847	0,2327
40	0,5064	3,1672	-1,4936	-0,5777	0,3059	-0,4191	0,8682	0,0104	-0,7379
41	1,5662	-2,3668	-0,4145	1,4612	-1,1341	-0,4643	0,3559	0,3753	0,3984
42	1,6617	0,2427	-0,4284	-0,5447	-0,6598	-0,2552	0,1097	-0,3479	-0,2582
43	-2,0848	-1,2038	1,3712	0,5063	0,2513	-0,8218	-0,8333	-0,6222	0,3787
44	1,8364	1,2132	-0,9976	-0,6321	0,3848	-0,1704	0,1308	-1,3652	0,2173
45	0,5146	2,3258	-0,6308	0,9868	0,0482	0,2414	-1,1252	0,5349	0,3616

Ind.	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
46	1,6221	1,0622	-0,1507	-1,504	-0,945	-0,193	-0,0239	-0,2009	-0,8243
47	0,5373	0,0165	1,0757	0,1287	-0,5249	1,2846	-1,0293	0,2016	-0,3977
48	-1,9897	0,4663	0,8759	-0,0978	-0,3121	-0,0384	-0,3622	0,2215	0,3112
49	-3,6436	0,4799	-1,3093	1,0488	0,5433	-0,733	0,1771	-0,9288	0,2354
50	-2,3429	0,987	-1,5524	-0,205	-1,535	1,0643	0,3843	0,1443	0,621
51	0,8145	0,1885	-1,1891	0,9884	-0,8697	0,6621	-0,8776	-1,7	0,5768
52	0,5352	-0,3275	-1,0571	-1,2202	0,499	0,0994	-0,3017	-0,8077	-0,7359
53	2,1048	0,6641	0,9092	-0,1609	0,2991	-0,57	-0,0087	0,17	-0,0285
54	1,6067	-1,1217	-1,9372	1,1382	-0,0467	-0,1797	0,4466	-1,4096	0,9968
55	-0,0811	-1,792	0,0744	-1,1225	-0,1596	-0,8573	-0,9485	0,5538	-1,6903
56	1,5275	0,5173	-1,3818	-0,7803	-0,617	-0,0554	0,1795	-1,1156	-0,1353
57	0,4303	3,2368	-0,6709	-0,051	-0,2226	0,3702	-1,2355	0,4261	-0,1635
58	-1,5933	-0,0734	1,7222	-0,3062	-0,4485	-0,021	-0,3876	-0,3077	0,2097
59	2,1444	-0,1554	0,6315	0,7983	0,5842	-0,6322	0,1249	0,023	0,5375
60	1,7959	-0,0319	0,5251	-0,3091	-0,7027	-0,455	0,04	0,4197	-0,3811
61	-0,1764	-0,7287	1,0354	0,9426	1,5912	-0,3698	-2,103	-0,53	-0,273
62	-0,7414	-1,1321	-0,6869	-2,0178	-0,8174	-0,2763	0,0781	-0,3026	0,3853
63	-2,7114	-0,1462	1,9322	-0,0132	-0,0718	-0,9025	-0,1003	-0,8166	-0,6977
64	-0,001	-1,3663	-1,726	-0,4862	1,213	-0,6348	-0,7242	-1,3781	-0,5258
65	-3,0067	0,8782	-0,6794	-0,1932	0,2674	-0,6612	-0,8695	0,0714	0,0977
66	0,2203	0,2561	-0,3501	1,6596	-0,3426	-0,7537	1,9554	-0,1369	-0,6932
67	0,0376	-0,2725	-0,1381	-0,6233	0,4325	-0,1767	0,5183	-0,6374	-1,3281

sem contrato), de acordo com as hipóteses abaixo:

- **Hipótese nula -** H_0**:** os clientes que possuem contratos de computação em nuvem apresentam os mesmos níveis de satisfação geral com os serviços do que os que não possuem contratos;
- **Hipótese alternativa -** H_1**:** os clientes que possuem contratos de computação em nuvem **não** apresentam os mesmos níveis de satisfação geral com os serviços do que os que não possuem contratos.

Como uma grande proporção da freqüência das respostas soma menos de cinco, então é recomendado o uso do teste exato de Fisher, porque fornece um valor p mais preciso do que o teste qui-quadrado. Se o valor p para o teste estatístico é maior do que $0,05$, então aceita-se a hipótese nula de que não há diferença significativa entre as duas categorias que responderam à questão.

O grau de satisfação geral dos usuários de computação em nuvem por categoria e pelo total de respondentes está mostrado na Figura B.16, bem como as estatísticas obtidas com o teste de hipótese. Obteve-se um valor $p = 0,705$; portanto, aceita-se a hipótese nula, que infere que **não** há diferença estatística entre as categorias. As estatísticas complementares são o número de amostras (N), os graus de liberdade (df) e o valor Φ de Cramer, cujos valores são respectivamente $N = 67$, $df = 6$, $\Phi = 0,17$. O valor Φ de Cramer é uma estatística que mede a intercorrelação de duas variáveis discretas e está associada à qualidade de ajuste do teste de hipótese. Se $\Phi = 1$, então as duas variáveis são iguais. Neste estudo, a intercorrelação das categorias é baixa [Cramér 1946].

B.5 Validação

A avaliação da confiabilidade do questionário seguiu a abordagem de *inter-observador*. Utilizou validação por *face* para as perguntas de pesquisa. Houveram dois especialistas que preencheram o questionário (mas as perguntas não foram contabilizadas) para criticá-lo e propor aprimoramentos [Collingridge 2015]. Empregou-se o método de *validade de conteúdo* para verificar se o conteúdo era condizente para que os objetivos fossem atingidos.

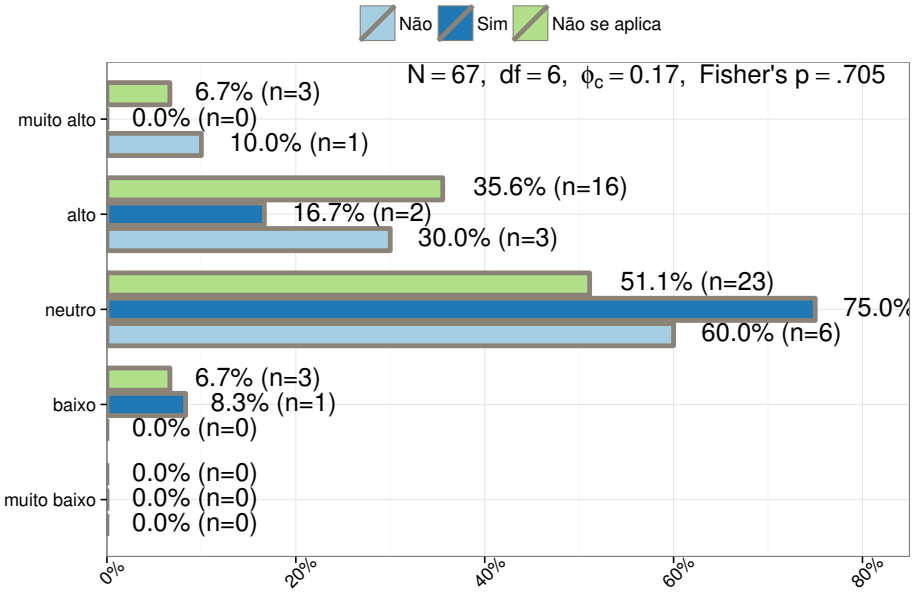


Figura B.16: Comparativo do grau de satisfação geral dentre os clientes com e sem contratos de serviços de computação em nuvem.

Cada objetivo contemplou um conjunto particular de questões, cobrindo os conteúdos esperados [Lehtonen e Pahkinen 2004] [Travassos 2002].

Para validar consistência interna das perguntas do questionário, empregou-se um teste chamado de α de Cronbach (CA). Como resultado, foram descobertos três fatores, ou cargas fatoriais (*factor loadings*), também genericamente chamados de componentes. Esses fatores medem a correlação entre perguntas. Perguntas que medem a mesma coisa devem ser "carregadas" para o mesmo fator. O resultado da análise de consistência interna das questões revelou uma correlação mais elevada entre as questões referentes ao mesmo objetivo, demonstrando que tais questões mensuram o mesmo fator (vide Tabela B.1).

No entanto, o teste CA apresentou um baixo nível de consistência interna, que significa níveis inferiores a $0,6$; porém, não há um consenso neste valor, que indica semelhanças entre as respostas dos indivíduos. Para melhorar a consistência interna, recomenda-se remover do experimento as questões que contribuem para diminuir o valor de CA e reter apenas as questões cujos fatores são superiores a $0,6$. Assumindo esse critério, as Questões 9 e 10 (da Meta 4) não se ajustaram a nenhum dos 3 componentes identificados, demonstrando que fazem parte de um outro fator.

Diante dos resultados obtidos para a análise CA, recomenda-se reaplicar o instrumento e realizar novas análises. Porém, essa nova etapa está fora do escopo deste trabalho, que teve por objetivo obter uma visão geral por meio de um estudo piloto. Os resultados obtidos com a análise CA encontram-se na Tabela B.6.

Tabela B.6: Análise de Consistência Interna do Questionário.

Questão	**No. Questão**	**Componente 1**	**Componente 2**	**Componente 3**
area.atuacao	2	-0.29	0.07	-0.77
existe.contrato	6	0.74	0.18	0
tipo.empresa	1	0.14	-0.11	-0.86
satisfacao.tecnica	11	0.06	0.72	0.12
conhece.sla	7	-0.72	0.03	-0.07
monitoramento.web	9	-0.42	-0.47	0.05
serviços.pagos	0.65	-0.14	0.06	
satisfacao.geral	12	-0.26	0.63	-0.11
sla.web	10	-0.48	-0.48	-0.12
α ***de Cronbach***	–	*-1*	*-0.05*	*0.38*

B.6 Conclusões

Estratégias de governança de nuvem com foco no nível de rede são importantes, pois agregam valor ao negócio. Elas contribuem para transmitir aos clientes maior confiabilidade ao contratar serviços de nuvem, tendo em vista que o tráfego de rede possui papel essencial para a computação em nuvem. Os resultados reforçam a importância da transparência no provimento dos serviços de rede e para o processo de tarifação de serviços.

Apenas 15% dos provedores analisados disponibilizavam algum tipo de informação sobre a não-conformidade com os SLAs de rede, e a maioria dos clientes que contrataram esses serviços não dispunham de conhecimento sobre as cláusulas de seus SLAs.

Por outro lado, o monitoramento *online* de tráfego de rede em alta velocidade é uma atividade muito complexa. Há demanda por uma sistematização complexa de eventos que vai

desde o manuseio do tráfego agregado, à identificação dos fluxos de serviços que requerem monitoramento e análise de conformidade de SLA.

Sem ferramentas para o monitoramento e análise de desempenho, os clientes precisam confiar sem evidências concretas no funcionamento correto dos serviços na nuvem. De outro ponto de vista, também é de interesse dos provedores aumentar o grau de confiança dos clientes nos serviços de computação em nuvem prestados.

Nesta tese, defende-se que disponibilizar metas técnicas sobre o tráfego de rede também aos clientes, ajuda a consolidar os serviços de nuvem em setores econômicos que dependem da informação de forma crítica para o negócio.

Apêndice C

Levantamento de Requisitos da TADE

> "Caiu a chuva, vieram as enchentes, sopraram os ventos e investiram contra aquela casa; ela, porém, não caiu, porque estava edificada na rocha."
>
> Mateus 7, 25

Neste apêndice são apresentados os requisitos funcionais e não-funcionais para o desenvolvimento dos produtos de *software* que implementam a arquitetura para detecção e gerenciamento de anomalias de tráfego para serviços de computação em nuvem (TADE) definida no Capítulo 3. A categorização dos requisitos envolve o tipo de requisito, se funcional ou não-funcional, e a obrigatoriedade do mesmo, de acordo com as prioridades que possuem junto ao processo de detecção de anomalias. Um código representa a categoria do requisito para permitir sua rápida identificação.

A Tabela C.1 contém os símbolos utilizados para identificar os requisitos com base nas propriedades *obrigatoriedade*, *tipo* e *área*.

Tabela C.1: Propriedades utilizadas para definir os requisitos.

Obrigatoriedade	**Tipo**	**Área**	**Descrição da Área do Requisito**
Obrigatório	RF - Requisito Funcional	TEM	Tarifação e Monitoramento
Desejável	RNF - Requisito Não Funcional	TEM	Tarifação e Monitoramento

A categorização propriamente dita dos requisitos é feita na Tabela C.2. A descrição

detalhada de cada requisito é apresentada na Tabela C.3, bem como o levantamento das dependências inter-requisitos, quando houver.

Tabela C.2: Categorização dos requisitos funcionais e não funcionais.

Código	**Título**	**Tipo**	**Categoria**
TEM_RNF1	Interface para acesso aos SLAs estabelecidos	Obrigatório	RNF
TEM_RNF2	Linguagem padrão para especificação de SLAs	Obrigatório	RNF
TEM_RNF3	Interface para notificação de suspeita de anomalias	Desejável	RNF
TEM_RF1	Interpretação de SLAs	Obrigatório	RF
TEM_RF2	Monitoramento do tráfego da rede	Obrigatório	RF
TEM_RF3	Análise de tráfego para reconhecimento de anomalias	Obrigatório	RF
TEM_RF4	Alerta de suspeita de anomalias	Obrigatório	RF
TEM_RF5	Serviço de publicação de anomalias	Desejável	RF

Tabela C.3: Descrição dos requisitos e das dependências inter-requisitos.

Código	Descrição	Dependências
TEM_RNF1	Os SLAs estabelecidos para os serviços serão usados para identificação de anomalias.	
TEM_RNF2	Para facilitar a interpretação dos acordos, faz-se necessária uma padronização da linguagem de especificação de SLAs.	
TEM_RNF3	Em caso de suspeita de anomalias de tráfego, módulos do sistema podem se cadastrar para receberem notificações.	TEM_RF4
TEM_RF1	Os SLAs serão analisados para que os seus dados alimentem o mecanismo de reconhecimento de anomalias de tráfego.	TEM_RNF1 TEM_RNF2
TEM_RF2	Será montado um mecanismo para monitoramento do tráfego em pontos estratégicos da rede.	
TEM_RF3	Implementação de técnicas de detecção de anomalias baseadas no reconhecimento de desvios nos SLAs negociados.	TEM_RF1 TEM_RF2
TEM_RF4	Módulo para notificação de anomalias de tráfego.	TEM_RF3
TEM_RF5	Aplicação Web para acompanhamento dos padrões de tráfego do sistema.	TEM_RF3 TEM_RF4

Apêndice D

Casos de Uso da TADE

> "O juiz não é nomeado para fazer favores com a justiça, mas para julgar segundo a lei."
>
> Platão

O projeto de desenvolvimento dos produtos de *software* da arquitetura para detecção e gerenciamento de anomalias de tráfego para serviços de computação em nuvem (TADE) compreende três casos de uso. Eles definem as funcionalidades dos módulos para processamento de tráfego, análise de pacotes para extração das métricas de desempenho de rede e o processo de notificação de anomalias.

D.1 Planejamento de Versões

A implementação dos produtos de *software* da arquitetura TADE segui um processo de desenvolvimento iterativo e incremental. O projeto de entrega de versões dos produtos de *software* da arquitetura TADE contendo as funcionalidades propostas pelos requisitos definidos iniciou-se pelo desenvolvimento dos dois primeiros casos de uso *coleta de pacotes de rede* (TADE_UC-1) e *processamento de pacotes de rede* (TADE_UC-2) foram desenvolvidos na primeira versão do TADE. Na versão TADE 0.2 foi implementado de modo parcial um módulo para *notificação de anomalias* (TADE_UC-3). Na versão TADE 0.3 foram realizados melhoramentos nos casos de uso TADE_UC-2 e TADE_UC-3. A versão TADE 0.4 apresentou melhoramentos nos casos de uso TADE_UC-2 e TADE_UC-3. A integração da

arquitetura TADE com o projeto de desenvolvimento de um sistema de computação em nuvem federado chamado *JiT Clouds* via canal de comunicação *Publish-Subscribe* ocorreu na quinta versão, bem como refatoramento e testes.

Os nomes dos casos de uso e projeto de versões para a entrega das funcionalidades definidas encontram-se na Tabela D.1.

Tabela D.1: Casos de Uso para implementação da arquitetura TADE.

Versão	Projeto de Implementação de Casos de Uso
TADE 0.1	Coletar Pacotes de Rede (TADE_UC-1)
	Processar Pacotes de Rede (TADE_UC-2)
TADE 0.2	Notificar Anomalias (TADE_UC-3)
TADE 0.3	Incrementos no Processamento de Pacotes de Rede (TADE_UC-2)
	Incrementos na Notificação de Anomalias (TADE_UC-3)
TADE 0.4	Incrementos no Processamento de Pacotes de Rede (TADE_UC-2)
	Incrementos na Notificação de Anomalias (TADE_UC-3)
TADE 1.0	Integração, refatoramento e testes

D.2 Coletar Pacotes de Rede (TADE_UC-1)

Descrição: O mecanismo de detecção de anomalias monitora o tráfego da rede de um determinado *DC Provider*, realiza a coleta e os encaminha para análise em um processador.

Entrada: Os SLAs negociados para os serviços que estão sendo ou serão executados pelos recursos do *DC Provider* e os pacotes que trafegam no(s) enlace(s) de comunicação.

Saída: Pacotes de um determinado serviço são enviados para serem analisados por um processador pré-determinado.

Implementação (Fluxo Principal): O mecanismo faz um levantamento dos SLAs negociados para os serviços escalonados no *DC Provider* e em que recursos eles estão sendo executados. A coleta de pacotes pode ser feita de modo distribuído. Neste caso, os pacotes

serão encaminhados para serem analisados em um processador específico, tendo por princípio que os pacotes de um determinado serviço devem ser destinados ao mesmo processador. Alguns pacotes podem ser armazenados em disco para uma posterior análise *offline* no intuito de realizar auditoria para o serviço de tarifação. Apenas serão coletados os pacotes advindos de serviços para os quais exista um SLA associado.

Diagrama de Sequência: Apresentado na Figura D.1.

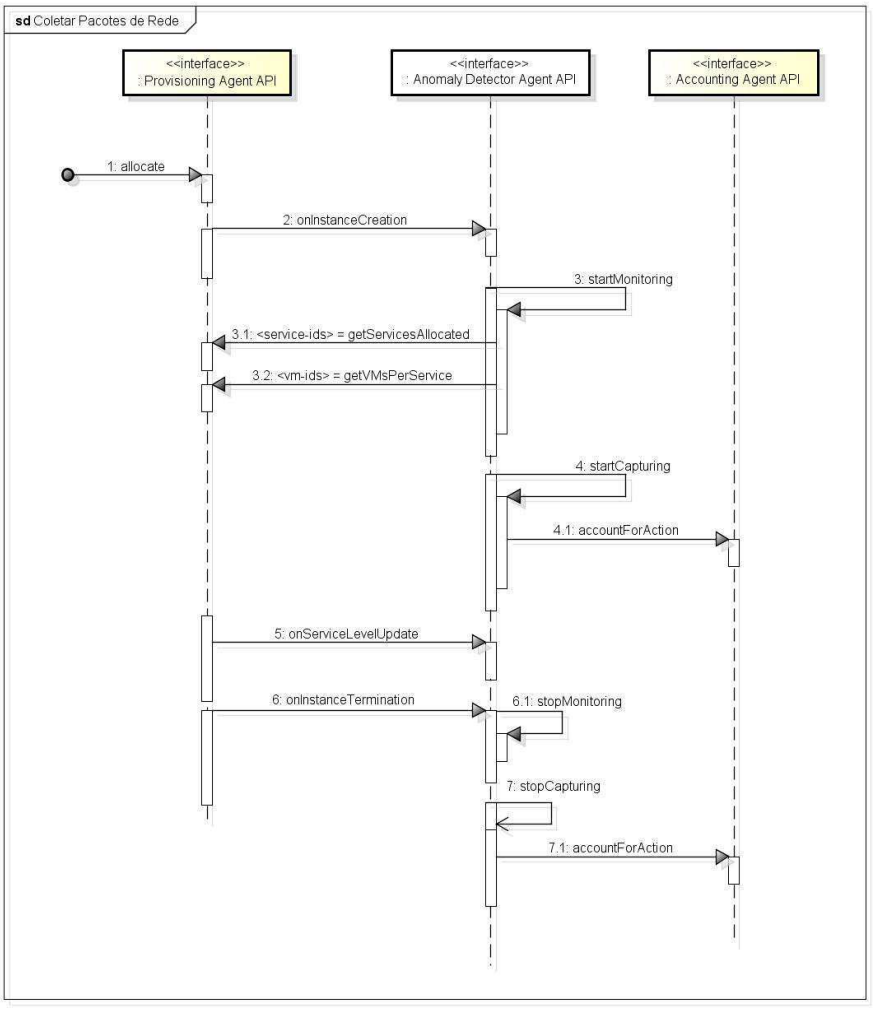


Figura D.1: Diagrama de sequência do caso de uso 1 da TADE.

Requisitos de Origem:

- TEM-2_RNF-1 (*Interface para Acesso aos SLAs Estabelecidos*);

- TEM-2_RNF-2 (*Linguagem Padrão para Especificação de SLAs*);
- TEM-2_RF-1 (*Interpretação de SLAs*);
- TEM-2_RF-2 (*Monitoramento do Tráfego da Rede*).

Casos de Uso Associados:

- MBF_UC-3 (Iniciar Instâncias);
- MBF_UC-5 (Encerrar Instâncias);
- MBF_UC-8 (Adicionar *DC*);
- MBF_UC-12 (Emitir Faturas).

Módulos e Operações:

Provisioning Agent API

allocate(<credential>, <allocation-plan> **)** : <vm-ids>

Realiza a alocação efetiva de instâncias em Resources de acordo com um plano de alocação.

getServicesAllocated() : <*service*-ids>

Requisita todos os serviços que foram alocados no DC Provider que devem ser monitorados.

getVMsPerService(<*service-id*>**)** : <vm-ids>

Requisita os identificadores das máquinas virtuais que foram alocadas para executar um serviço no DC Provider que está sendo monitorado.

Accounting API

accountForAction(<time>, "action", . . . **)** : <void>

Registra no repositório as ações importantes ocorridas durante a operação da Cloud.

Anomaly Detector Agent

startMonitoring(<vm-ids>,<*sla*> **)** : <void>

Inicia o monitoramento de anomalias de uma ou mais máquinas virtuais que compartilhem os mesmos requisitos de nível de serviço.

stopMonitoring(<vm-ids> **)** : <void>

Encerra o monitoramento de anomalias de uma ou mais máquinas virtuais.

startCapturing(<vm-ids>, <slas>, <type_of_capturing> **)** : <void>

Inicia a captura dos pacotes de uma ou mais máquinas virtuais.

stopCapturing(<vm-ids> **)** : <*void*>

Encerra a captura dos pacotes de uma ou mais máquinas virtuais.

onInstanceCreation(<vm-ids>, <**slas**>**)** : <void>

Inicia o monitoramento de anomalias, baseado no acordo de nível de serviço, após a criação de uma ou mais máquinas virtuais.

onInstanceTermination(<vm-ids> **)** : <void>

Executa ações para encerrar o monitoramento de uma ou mais máquinas virtuais após o término de sua execução.

onServiceLevelUpdate(<slas> **)** : <void>

Caso haja alteração em acordo de nível de serviço, este método atualiza os requisitos de monitoramentopara as máquinas virtuais.

D.3 Processar Pacotes de Rede (TADE_UC-2)

Descrição: O mecanismo de detecção de anomalias realiza a análise dos pacotes para identificar a ocorrência de violações nos acordos de nível de serviço.

Entrada: Os SLAs negociados para os serviços que estão sendo ou serão executados pelos recursos do *DC Provider* e os pacotes pré-filtrados que foram coletados pelo mecanismo de detecção de anomalias.

Saída: Alerta sobre a presença de anomalias.

Implementação (Fluxo Principal): A análise de anomalias pode ser feita de modo *online* recebendo os pacotes pré-filtrados, ou de modo *offline* processando o arquivo de rastro dos pacotes que foram armazenados em disco. Com base nos pacotes coletados, nos serviços associados a eles e nos acordos negociados para cada serviço, são extraídas as métricas para os parâmetros que foram acordados nos SLAs, e.g. consumo de banda no enlace de comunicação e atraso entre dois nós na rede. O processamento dos pacotes consiste em comparar as métricas obtidas com os limiares presentes nos SLAs dos serviços. Caso seja identificado um valor que esteja fora dos limiares estabelecidos para um determinado parâmetro, é reportado para o DC Controller um alerta sobre a presença (suspeição) de uma anomalia. Caso contrário, nenhuma mensagem é reportada.

Diagrama de Sequência: Apresentado na Figura D.2.

Requisitos de Origem:

- TEM-2_RNF-1 (*Interface para Acesso aos SLAs Estabelecidos*);
- TEM-2_RNF-2 (*Linguagem Padrão para Especificação de SLAs*);
- TEM-2_RF-1 (*Interpretação de SLAs*);
- TEM-2_RF-2 (*Monitoramento do Tráfego da Rede*);
- TEM-2_RF-3 (*Análise de Tráfego para Reconhecimento de Anomalias*);

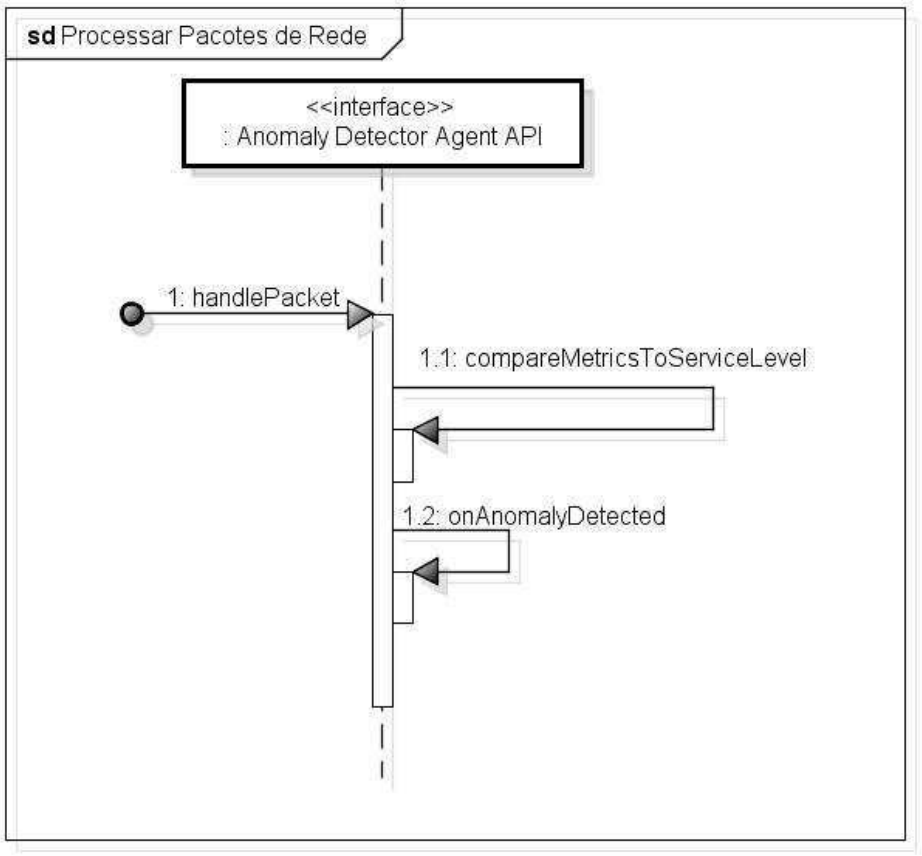


Figura D.2: Diagrama de Sequência do caso de uso 2 da TADE.

- TEM-2_RF-4 (*Alerta de Suspeita de Anomalias*).

Casos de Uso Associados:

- TADE_UC-1 (Coletar Pacotes de Rede);
- MBF_UC-3 (Iniciar Instâncias);
- MBF_UC-5 (Encerrar Instâncias);
- MBF_UC-8 (Adicionar Data Center);
- MBF_UC-12 (Emitir Faturas).

Módulos e Operações:

Anomaly Detector Agent

onAnomalyDetected(<vm-id>, <*anomaly*> **)** : <void>

Dispara a execução de ações de notificação e/ou de recuperação após uma anomalia ter sido detectada.

handlePacket(<**vm-id**> **)** : <metrics>

Quando um pacote é recebido, as métricas associadas a ele são calculadas ou verificadas. Esta ação não precisa ser recalculada a cada pacote, mas pode estar condicionada à ocorrência de um evento.

compareMetricsToServiceLevel(<**metrics**> **)** : <void>

Compara os valores obtidos para as métricas que estão sendo monitoradas com os valores acordados no SLA. Isto é feito para cada máquina virtual sendo monitorada. Se houver discrepância de valores, os valores fora do limiar estabelecido são retornados, caso contrário, o valor será nulo.

D.4 Notificar Anomalias (TADE_UC-3)

Descrição: O mecanismo de detecção de anomalias é responsável por notificar as anomalias identificadas para o *DC Controller*, que se encarregará de executar as ações cabíveis.

Entrada: Saída do caso de uso TADE_UC-2, que representam os alertas sobre a presença de anomalias.

Saída: Informações específicas para caracterizar as anomalias encontradas.

Implementação (Fluxo Principal): A notificação de anomalias consiste em informar ao DC Controller que tipo de anomalia foi detectado (e.g. violação do parâmetro largura de banda presente no SLA), data, hora, duração, volume de dados anômalos, qual o serviço em que está ocorrendo a anomalia e entre quais VMs, incluindo informações sobre os nós de origem e destino. Há ações que serão tomadas a partir dessa identificação. Algumas delas são comuns a todos os tipos de anomalias, como o registro da anomalia no *DC Accounting Agent* e sua inclusão na base de dados do *DC Repository*. Exemplos de ações específicas são o bloqueio de tráfego de rede anômalo e a migração de uma VM. As ações específicas devem estar definidas no SLA do serviço.

Diagrama de Sequência: Apresentado na Figura D.3.

Requisitos de Origem:

- TEM-2_RNF-1 (*Interface para Acesso aos SLAs Estabelecidos*);
- TEM-2_RNF-2 (*Linguagem Padrão para Especificação de SLAs*);
- TEM-2_RNF-3 (*Interface para Notificação de Suspeita de Anomalias*);
- TEM-2_RF-1 (*Interpretação de SLAs*);
- TEM-2_RF-4 (*Alerta de Suspeita de Anomalias*);
- TEM-2_RF-5 (*Serviço de Publicação de Anomalias*).

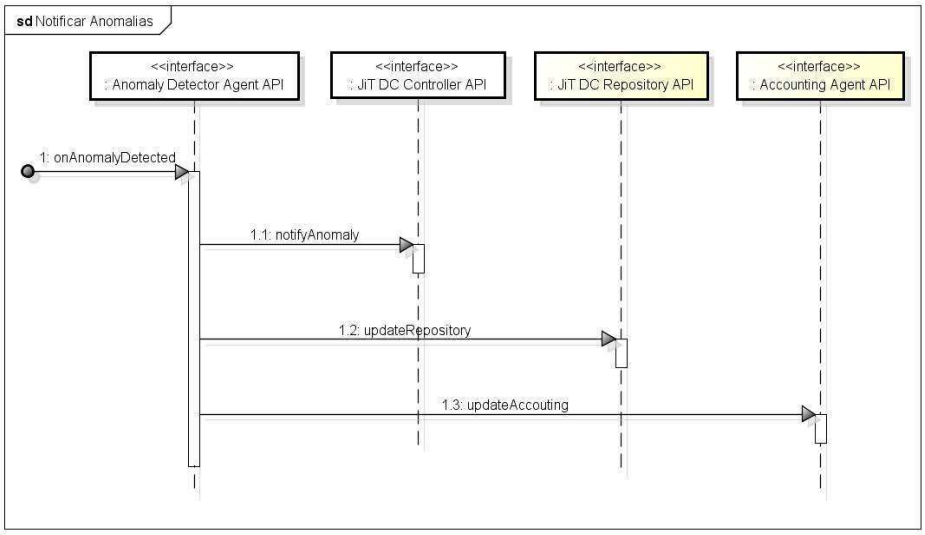


Figura D.3: Diagrama de sequência do caso de uso 3 da TADE.

Casos de Uso Associados:

- TADE_UC-2 (Processar Pacotes de Rede).

Módulos e Operações:

Anomaly Detector Agent

onAnomalyDetected(<vm-id>, <*anomaly*>**)** : <void>

Dispara a execução de ações de notificação e/ou de recuperação após uma anomalia ter sido detectada.

DC Controller API

notifyAnomaly(<vm-id>,<*anomaly*>**)** : <void>

Informações sobre uma anomalia detectada em uma máquina virtual.

DC Repository API

updateRepository(<vm-id>, <*anomaly*>**)** : <void>

Quando anomalias são detectadas, o mecanismo de detecção de anomalias atualiza o estado do sistema no repositório de dados do DC Provider.

Accounting Agent API

updateAccounting(<*vm-id*>, <*anomaly*>**)** : <void>

Quando anomalias são detectadas, o mecanismo de detecção de anomalias atualiza histórico de eventos no DC Accouting Agent.

Apêndice E

Ambiente de Testes da TADE

"O caráter de um homem é o seu destino."

Heráclito

O ambiente de testes que foi utilizado no desenvolvimento e testes da arquitetura TADE consiste de um servidor para análise de anomalias de tráfego e seis máquinas responsáveis por injetar tráfego na rede. O tráfego gerado na rede de testes é espelhado para a(s) interfaces(s) de rede da máquina analisadora, onde é capturado e processado para verificar se está ou não em conformidade com as metas técnicas acordadas no SLA. As máquinas responsáveis pela injeção de tráfego são chamadas de **máquinas geradoras de tráfego** ou *SLAVE* e a **máquina analisadora** foi denominada *ANALYZER*.

A máquina analisadora é capaz de analisar tanto tráfego real, quanto tráfego sintético sem necessidade de modificações, contanto que a carga sintética seja composta por pacotes no mesmo formato que pacotes reais. Desta forma, pode-se aplicar o *ANALYZER* na análise de desempenho da rede de um sistema em produção apenas trocando a fonte de tráfego.

Os servidores *SLAVES* geram os pacotes para o *ANALYZER*, os quais trafegam pela rede de testes, gigabit Ethernet, até o *switch* onde é feito o espelhamento do tráfego para o *ANALYZER*. Na Figura E.1 está ilustrada a topologia de rede do ambiente de testes, mostrando as máquinas de injeção de tráfego e a máquina analisadora de tráfego, bem como a capacidade dos enlaces que as interconectam.

Este ambiente de teses está localizado no laboratório do *Grupo de Pesquisa em Redes Convergentes* (GPRC) no Instituto Federal de Educação, Ciência e Tecnologia da Paraíba

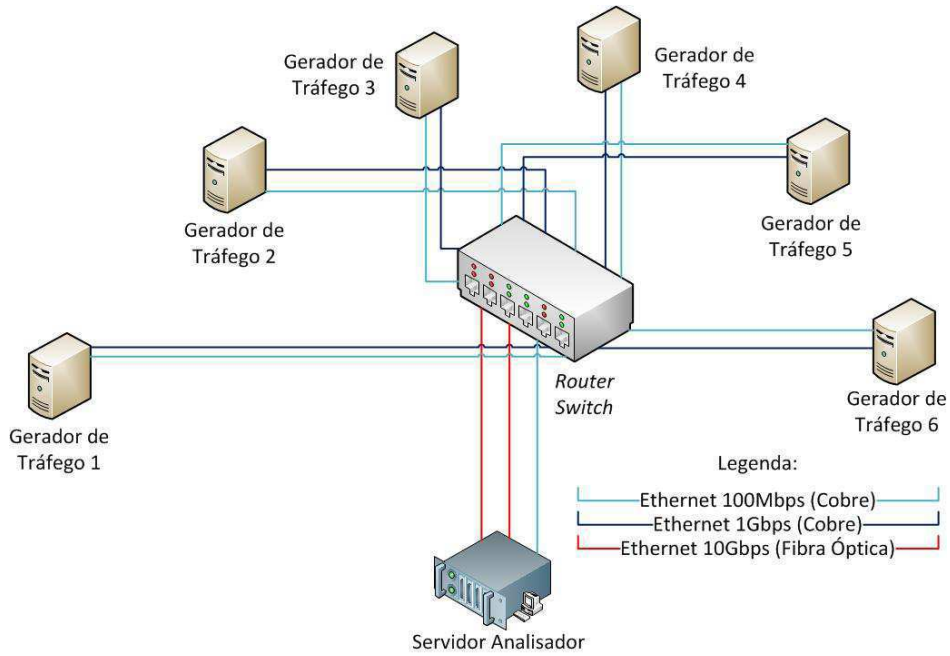


Figura E.1: Topologia do ambiente de testes.

(IFPB), *Campus* Campina Grande.

Pode-se observar a existência de duas LANs virtuais (VLAN) para o ambiente de testes. A VLAN 1 corresponde à rede de testes e a VLAN 2 é faz parte da rede do campus do IFPB, que será chamada de rede administrativa. A rede administrativa interconecta os dispositivos a 100Mbps e é utilizada para realizar tarefas administrativas como configuração e suporte remotos, bem como prover acesso à Internet para garantir atualizações e instalações de softwares armazenados em repositórios externos.

Na Tabela E.1 se encontra um sumário das especificações em termos de *software* dos equipamentos utilizados. O sistema operacional usado para os testes nas máquinas SLAVE foi o Ubuntu Desktop 11.04, porque se mostrou compatível com as bibliotecas e softwares necessários. Já o sistema operacional do ANALYZER é o Ubuntu Server 11.04, por ser mais leve do que a versão desktop, onde é priorizado o desempenho.

Para testes de injeção de tráfego na rede, foi empregado um gerador de tráfego sintético de código aberto chamado Ostinato que pode produzir uma carga de até 10Gbps [Ostinato 2015]. Para realizar captura de pacotes foi utilizado o módulo PF_RING [PF_RING 2015]. Com isso, pode-se realizar a validação do mecanismo de captura de pacotes sem perdas em até 10Gbps.

Tabela E.1: Especificação dos softwares utilizados nas máquinas do ambiente de testes.

#	Software	Analyzer	Slave
1	Ubuntu Server 11.04	X	
2	Ubuntu 11.04		X
3	Gcc 4.5.2	X	X
4	Ostinato 0.5	X	X
5	PF_RING 5.2.0	X	X
6	UTHash 1.9.4	X	

A linguagem de programação utilizada para implementar os algoritmos de detecção de anomalias foi C/C++ por ser de baixo nível e por razões de compatibilidade com as demais API utilizadas. Mais precisamente, nas estações de trabalho utilizadas para o desenvolvimento dos produtos de *software* que implementam a arquitetura TADE são necessários os seguintes *softwares*, bem como a instalação de suas dependências:

1. PF_RING 5.2.0
2. Gcc 4.5.2
3. UTHash 1.9.4
4. Oracle Java JDK 1.7
5. Eclipse Indigo Release

Na Tabela E.2 estão descritas as especificações concernentes ao *hardware* das máquinas injetoras de tráfego.

Na Tabela E.3 estão as especificações de *hardware* da máquina analisadora de tráfego.

Vale ressaltar algumas limitações físicas do ambiente dos testes de validação da implementação da arquitetura TADE, tais como:

1. Como o ambiente de testes é composto por 6 máquinas SLAVE, os experimentos realizados poderão injetar nominalmente até 6 Gbps de tráfego na **rede de testes**. Mesmo

Tabela E.2: Especificações do *hardware* das máquinas SLAVE do ambiente de testes.

Dispositivo	Especificação
Placa-mãe	**Intel S3420GP**
Processador	Intel Xeon X3430 @ 2,40GHz (4 núcleos, 8 threads)
Memória RAM	6GB DDR3
Interface de rede	82578DM Gigabit Network Connection 82574L Gigabit Network Connection

Tabela E.3: Especificação do hardware da máquina ANALYZER (HP ProLiant DL380 G7) do ambiente de testes.

Modelo	Quantidade	Especificação	Capacidade Total
Processador	2	Intel Xeon CPU X5670 @ 2.93GHz	12 núcleos x 2,93 GHz
Memória RAM	1	32GB DDR3	32 GB DDR3
Discos Rígidos	2 (em RAID 0)	Capacidade: 300GB (cada)	600 GB Taxa de Transferência: 6Gb/s cada um, sendo 12Gb/s para leitura/escrita em paralelo
Interface de rede	1	Intel XF SR 10 Gigabit Server Adapter dual EXPX9502AFXSR	2 x 10 Gbps
Interface de rede	4	NetXtreme II BCM5709 Gigabit Ethernet	4 x 1 Gbps

havendo 2 interfaces de rede gigabit, 1 delas está sendo usada exclusivamente na **rede administrativa**;

2. Caso seja aumentado o número de máquinas SLAVE, o ANALYZER é capaz de capturar nominalmente até 20 Gbps na rede de testes;

3. Há uma limitação física para leitura/escrita de *traces* de tráfego no disco rígido, que é 12 Gbps (taxa teórica máxima);

4. É possível utilizar as 2 interfaces 10 Gbps da máquina ANALYZER, pois a mesma possui ainda outras 4 interfaces gigabit que podem ser configuradas para acessarem a rede administrativa.

O chaveamento dos dispositivos teclado, monitor e mouse entre os servidores SLAVE e ANALYZER por meio de um equipamento de *rack* chamado de *switch* KVM. Por fim, na estão listadas as especificações dos equipamentos adicionais presentes no ambiente de testes do centro de dados que implementa a arquitetura TADE na Tabela E.4.

Tabela E.4: Equipamentos adicionais do ambiente de testes.

Equipamento	**Quantidade**	**Marca / Modelo**	**Número de Portas**
Switch KVM	1	TRENDNET / TK-160R	16
Switch de Rede Layer-2	1	3COM / 4210G M/N: 3CRS42G-24-91	24 portas gigabit e 2 *slots* para módulo de expansão 10 Gbps

Apêndice F

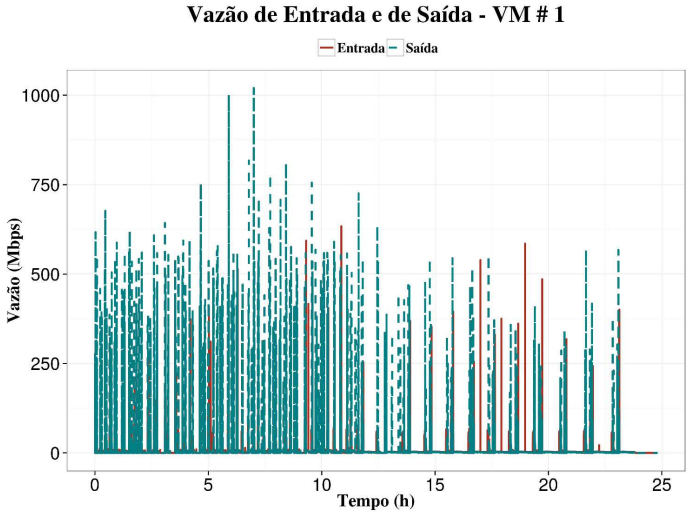
Dados Suplementares do Capítulo 7

“"Muitos dos fracassos da vida ocorrem com pessoas que não perceberam quão próximas estavam do sucesso quando desistiram.”

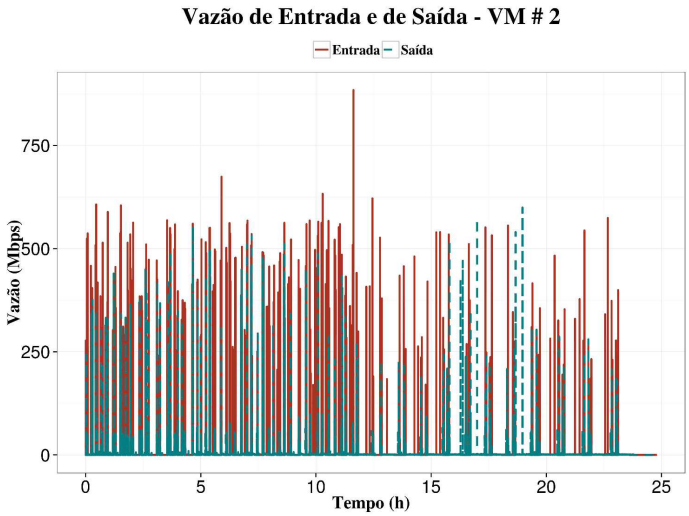
Thomas Edison

Tabela F.1: Estatísticas sobre a vazão do tráfego (Mbps) de entrada e saída das VMs.

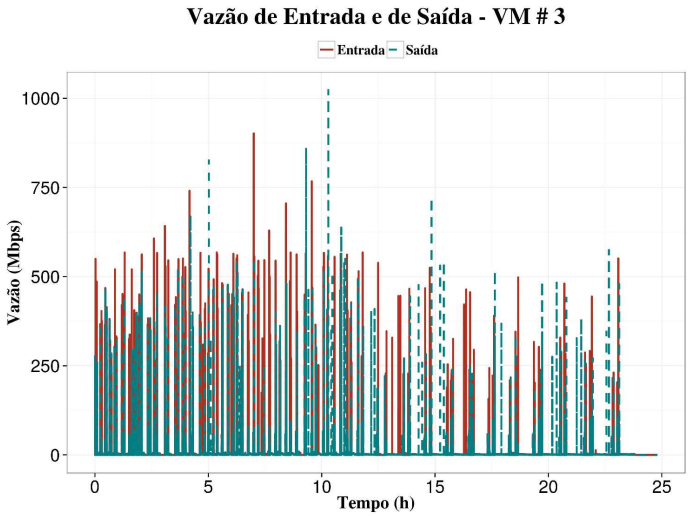
VM ID	Direção	Mínimo	Quantil 1	Mediana	Média	Quantil 3	Máximo
1	Entrada	0,003128	0,05736	1,8010	3,722	2,932	633,6
2	Entrada	0,001320	0,01333	0,2869	2,964	0,792	884,3
3	Entrada	0,002022	0,01478	0,7525	3,383	1,598	901,6
1	Saída	0,001320	0,0231	1,3040	4,911	2,425	1020,0
2	Saída	0,003128	0,02757	0,1582	1,721	1,269	599,6
3	Saída	0,002968	0,03101	0,7649	3,344	1,667	1023,0



(a) VM #1



(b) VM #2



(c) VM #3

Figura F.1: Vazão de entrada e saída das 3 máquinas virtuais.

Tabela F.2: Estimativa das variáveis R, T, q e k calculadas a cada hora de experimento para a VM #1.

VM ID	R (GB)	T (GB)	Tempo Relativo (h)	k (h/GB)	q (GB/h)
1	1,648087	12,04433	0	0,606764	12,04433
1	5,427169	15,99448	1	0,184258	15,99448
1	11,47097	19,83017	2	0,087177	19,83017
1	7,810265	15,44081	3	0,128037	15,44081
1	13,7667	22,02464	4	0,072639	22,02464
1	14,78304	21,29445	5	0,067645	21,29445
1	15,76003	19,90460	6	0,063452	19,9046
1	13,69328	15,7922	7	0,073029	15,7922
1	14,87652	22,74587	8	0,067220	22,74587
1	14,94043	17,43455	9	0,066932	17,43455
1	19,04409	28,35983	10	0,052510	28,35983
1	18,6485	21,95446	11	0,053624	21,95446
1	14,98095	17,33383	12	0,066751	17,33383
1	7,413051	12,87591	13	0,134897	12,87591
1	11,85857	14,45141	14	0,084327	14,45141
1	13,9966	14,73137	15	0,071446	14,73137
1	14,97974	14,57562	16	0,066757	14,57562
1	13,43354	17,14907	17	0,074441	17,14907
1	14,23777	14,14325	18	0,070236	14,14325
1	13,82962	14,45283	19	0,072309	14,45283
1	14,25931	13,32380	20	0,070130	13,32380
1	11,9294	15,06408	21	0,083826	15,06408
1	15,57465	15,56318	22	0,064207	15,56318
1	13,59238	16,29292	23	0,073571	16,29292
1	2,110116	1,64616	24	0,473908	1,64616
1	0,025999	0,01133	25	38,463440	0,01133

Tabela F.3: Estimativa das variáveis R, T, q e k calculadas a cada hora de experimento para a VM #2.

VM ID	R (GB)	T (GB)	Tempo Relativo (h)	k (h/GB)	q (GB/h)
2	10,61587	3,498514	0	0,094199	3,498514
2	11,36677	4,83863	1	0,087976	4,83863
2	14,90098	6,374729	2	0,06711	6,374729
2	7,396339	4,986509	3	0,135202	4,986509
2	12,84582	6,947746	4	0,077846	6,947746
2	14,34296	7,332953	5	0,069721	7,332953
2	9,37329	5,385264	6	0,106686	5,385264
2	8,54807	7,325835	7	0,116985	7,325835
2	9,36574	6,895771	8	0,106772	6,895771
2	7,78597	6,150679	9	0,128436	6,150679
2	18,84980	7,367116	10	0,053051	7,367116
2	13,88888	8,185633	11	0,072	8,185633
2	13,10010	5,941601	12	0,076335	5,941601
2	8,29607	3,131679	13	0,120539	3,131679
2	11,40098	4,901506	14	0,087712	4,901506
2	5,27428	4,66251	15	0,189599	4,66251
2	6,85520	8,428813	16	0,145875	8,428813
2	13,64063	8,646525	17	0,07331	8,646525
2	10,24307	4,461727	18	0,097627	4,461727
2	6,83182	7,83867	19	0,146374	7,83867
2	9,97051	5,408905	20	0,100296	5,408905
2	5,98468	4,83865	21	0,167093	4,83865
2	5,67855	5,5024	22	0,176101	5,5024
2	13,25253	5,841394	23	0,075457	5,841394
2	0,32867	0,376997	24	3,042573	0,376997
2	0,01155	0,019397	25	86,56689	0,019397

Tabela F.4: Estimativa das variáveis R, T, q e k calculadas a cada hora de experimento para a VM #3.

VM ID	R (GB)	T (GB)	Tempo Relativo (h)	k (h/GB)	q (GB/h)
3	8,961143	5,333661	0	0,111593	5,333661
3	12,53372	8,419947	1	0,079785	8,419947
3	13,66313	13,72325	2	0,07319	13,72325
3	12,81193	7,677776	3	0,078052	7,677776
3	16,50281	13,87426	4	0,060596	13,87426
3	13,20717	13,3549	5	0,075716	13,3549
3	16,68905	16,26379	6	0,05992	16,26379
3	12,49895	11,34849	7	0,080007	11,34849
3	17,41888	12,21786	8	0,057409	12,21786
3	13,89933	12,66614	9	0,071946	12,66614
3	15,27317	17,81619	10	0,065474	17,81619
3	13,27482	16,07691	11	0,075331	16,07691
3	9,409881	13,54672	12	0,106271	13,54672
3	6,187625	5,537255	13	0,161613	5,537255
3	7,421836	10,70097	14	0,134738	10,70097
3	13,1441	13,04859	15	0,07608	13,04859
3	11,99501	10,40579	16	0,083368	10,40579
3	8,182603	8,735885	17	0,12221	8,735885
3	7,965315	12,92534	18	0,125544	12,92534
3	11,30689	8,967755	19	0,088442	8,967755
3	6,49812	11,82951	20	0,153891	11,82951
3	14,0645	10,98031	21	0,071101	10,98031
3	13,30138	12,81996	22	0,07518	12,81996
3	7,916834	12,15606	23	0,126313	12,15606
3	1,347675	1,766193	24	0,742019	1,766193
3	0,011355	0,019019	25	88,06938	0,019019

Tabela F.5: Estimativa das variáveis R, T, q e k calculadas a para o período total do experimento.

ID VM	R (GB)	T (GB)	Tempo (h)	k (h/GB)	q (GB/h)
1	$314,0908$	$414,4352$	24,79	$0,078917381$	$16,71971049$
2	$250,1492$	$145,2902$	24,79	$0,099095321$	$5,861165521$
3	$285,4872$	$282,2125$	24,79	$0,086834007$	$11,38412853$

Legenda para as colunas das tabelas que serão apresentadas a seguir:

- **h:** hora que está sendo realizada a análise para obtenção dos dados exibidos;
- **Custo DaaS:** custo do provimento do serviço de DaaS, aplicando a estratégia de escalonamento analisada (veja o título da tabela para identificá-la);
- **Custo E. DaaS:** custo da execução da VM do serviço de DaaS, aplicando a estratégia de escalonamento analisada;
- **Custo D. DaaS:** custo de transferência da fonte de dado do centro de dados de origem até onde a VM foi alocada, aplicando a estratégia de escalonamento analisada;
- **DC E.:** identificador do centro de dados onde a VM foi alocada, aplicando a estratégia de escalonamento analisada;
- **DC D.:** identificador do centro de dados de onde a fonte de dados para processamento da VM foi recuperada, aplicando a estratégia de escalonamento analisada;
- **Custo A. E. DaaS:** custo estimado para as anomalias de tráfego que ocorreram no centro de dados que estava executando a VM, aplicando a estratégia de escalonamento analisada;
- **Custo A. D. DaaS:** custo estimado para as anomalias de tráfego que ocorreram no centro de dados onde a fonte de dados estava localizada, empregando o escalonamento em questão;

- **Custo A. Pena. DaaS:** custo da penalidade a ser aplicada ao provedor de DaaS devido a incidência de anomalias na prestação do serviço. Esse custo representa uma estimativa de prejuízo causado ao cliente;
- **Custo IaaS:** custo do serviço de IaaS para o cliente;
- **Custo A. IaaS:** custo da penalidade a ser aplicada ao provedor de IaaS devido a incidência de anomalias na prestação do serviço. Esse custo representa uma estimativa de prejuízo causado ao cliente.

Tabela F.6: Decisões do escalonamento baseado em custo padrão, detalhamento de custos e perdas para a VM 1.

h	Custo DaaS	Custo E. DaaS	Custo D. DaaS	DC E.	DC Dados	Custo A. E. DaaS	Custo A. D. DaaS	Custo A. DaaS	Custo A. Pena. Fixa	Custo IaaS	Custo A. IaaS
0	0,436618	0,397064	0,039554	1	1	0,397064	0,039554	0,436618	0	0,08	0
1	0,687246	0,491868	0,195378	1	1	0,659868	1,107142	1,76701	0,14	0,08	3,11E-05
2	1,272182	0,583924	0,688258	1	1	0,631924	1,238865	1,870789	0,04	0,08	8,89E-06
3	0,666026	0,478579	0,187446	1	1	1,006579	4,311266	5,317846	0,44	0,08	9,78E-05
4	1,132193	0,636591	0,495601	1	1	1,164591	7,764421	8,929012	0,44	0,08	9,78E-05
5	1,506049	0,619067	0,886983	1	1	0,619067	0,886983	1,506049	0	0,08	0
6	0,963951	0,58571	0,378241	1	1	0,75371	3,025926	3,779637	0,14	0,08	3,11E-05
7	0,979971	0,487013	0,492958	1	1	0,487013	0,492958	0,979971	0	0,08	0
8	1,546492	0,653901	0,892591	1	1	1,061901	6,962213	8,024114	0,34	0,08	7,56E-05
9	0,885	0,526429	0,35857	1	1	0,694429	2,868563	3,562992	0,14	0,08	3,11E-05
10	1,474223	0,788636	0,685587	1	1	1,436636	13,02616	14,46279	0,54	0,08	0,00012
11	1,753817	0,634907	1,11891	1	1	0,802907	4,251857	5,054764	0,14	0,08	3,11E-05
12	0,883555	0,524012	0,359543	1	1	0,524012	0,359543	0,883555	0	0,08	0
13	0,683892	0,417022	0,26687	1	1	0,585022	1,512262	2,097284	0,14	0,08	3,11E-05
14	1,166348	0,454834	0,711514	1	1	0,454834	0,711514	1,166348	0	0,08	0
15	0,797471	0,461553	0,335918	1	1	0,749553	4,36694	5,116493	0,24	0,08	5,33E-05
16	0,997086	0,457815	0,539271	1	1	0,505815	1,258298	1,764113	0,04	0,08	8,89E-06
17	1,32559	0,519578	0,806012	1	1	0,519578	0,806012	1,32559	0	0,08	0
18	0,789144	0,447438	0,341706	1	1	0,447438	0,341706	0,789144	0	0,08	0
19	0,952734	0,454868	0,497866	1	1	0,742868	4,480797	5,223665	0,24	0,08	5,33E-05
20	1,28333	0,427771	0,855559	1	1	0,427771	0,855559	1,28333	0	0,08	0
21	0,755844	0,469538	0,286306	1	1	0,757538	3,721974	4,479512	0,24	0,08	5,33E-05
22	1,042204	0,481516	0,560688	1	1	1,369516	14,39098	15,7605	0,74	0,08	0,000164
23	1,314573	0,49903	0,815543	1	1	0,78703	4,730148	5,517179	0,24	0,08	5,33E-05
24	0,198151	0,147508	0,050643	1	2	0,147508	0	0,147508	0	0,08	0

Tabela F.7: Decisões do escalonamento baseado em custo padrão, detalhamento de custos e perdas para a VM 2.

h	Custo DaaS	Custo E. DaaS	Custo D. DaaS	DC E.	DC Dados	Custo A. E. DaaS	Custo A. D. DaaS	Custo A. DaaS	Custo A. Pena. Fixa	Custo IaaS	Custo A. IaaS
0	0,526291	0,175967	0,350324	2	1	0,176649	0	0,176649	0,00062	0,08	2,76E-07
1	0,830622	0,20545	0,625172	2	1	0,371132	0	0,371132	0,15062	0,08	6,69E-05
2	0,567066	0,239244	0,327821	2	1	0,404926	0	0,404926	0,15062	0,08	6,69E-05
3	0,452782	0,208703	0,244079	2	1	0,759385	0	0,759385	0,50062	0,08	0,000222
4	0,958371	0,25185	0,70652	2	1	0,527532	0	0,527532	0,25062	0,08	0,000111
5	0,57587	0,260325	0,315545	2	1	0,261007	0	0,261007	0,00062	0,08	2,76E-07
6	0,526794	0,217476	0,309319	2	1	0,438158	0	0,438158	0,20062	0,08	8,92E-05
7	0,730312	0,260168	0,470144	2	1	0,260168	0	0,260168	0	0,08	0
8	0,456753	0,250707	0,206046	2	1	0,471389	0	0,471389	0,20062	0,08	8,92E-05
9	0,491252	0,234315	0,256937	2	1	0,564997	0	0,564997	0,30062	0,08	0,000134
10	1,297816	0,261077	1,036739	2	1	0,756759	0	0,756759	0,45062	0,08	0,0002
11	0,584639	0,279084	0,305555	2	1	0,444766	0	0,444766	0,15062	0,08	6,69E-05
12	0,662019	0,229715	0,432303	2	1	0,285397	0	0,285397	0,05062	0,08	2,25E-05
13	0,624181	0,167897	0,456284	2	1	0,498579	0	0,498579	0,30062	0,08	0,000134
14	0,457655	0,206833	0,250822	2	1	0,317515	0	0,317515	0,10062	0,08	4,47E-05
15	0,375627	0,201575	0,174051	2	1	0,367257	0	0,367257	0,15062	0,08	6,69E-05
16	0,66147	0,284434	0,377036	2	1	0,560116	0	0,560116	0,25062	0,08	0,000111
17	0,589317	0,289224	0,300094	2	1	0,289906	0	0,289906	0,00062	0,08	2,76E-07
18	0,535179	0,197158	0,338021	2	1	0,25284	0	0,25284	0,05062	0,08	2,25E-05
19	0,647201	0,271451	0,37575	2	1	0,547133	0	0,547133	0,25062	0,08	0,000111
20	0,437347	0,217996	0,219351	2	1	0,217996	0	0,217996	0	0,08	0
21	0,402945	0,20545	0,197494	2	1	0,426132	0	0,426132	0,20062	0,08	8,92E-05
22	0,532373	0,220053	0,31232	2	1	0,770735	0	0,770735	0,50062	0,08	0,000222
23	0,519066	0,227511	0,291556	2	1	0,448193	0	0,448193	0,20062	0,08	8,92E-05
24	0,108581	0,097735	0,010846	1	1	0	0,010846	0,010846	0	0,08	0

Tabela F.8: Decisões do escalonamento baseado em custo padrão, detalhamento de custos e perdas para a VM 3.

h	Custo DaaS	Custo E. DaaS	Custo D. DaaS	DC E.	DC Dados	Custo A. E. DaaS	Custo A. D. DaaS	Custo A. DaaS	Custo A. Pena. Fixa	Custo IaaS	Custo A. IaaS
0	0,676967	0,206507	0,47046	1	1	0,206507	0,47046	0,676967	0	0,08	0
1	0,534527	0,271319	0,263208	1	1	0,434174	2,304387	2,738561	0,1551	0,08	0,000138
2	0,813077	0,382688	0,430389	1	1	0,493043	1,938183	2,431227	0,1051	0,08	9,34E-05
3	0,928359	0,255733	0,672626	1	1	0,576088	4,77699	5,353079	0,3051	0,08	0,000271
4	0,732418	0,385859	0,346559	1	1	0,627464	4,333719	4,961184	0,2301	0,08	0,000205
5	0,790979	0,374953	0,416026	1	1	0,380308	0,48675	0,867058	0,0051	0,08	4,53E-06
6	1,312214	0,43604	0,876175	1	1	0,651395	4,470244	5,121639	0,2051	0,08	0,000182
7	0,595296	0,332818	0,262478	1	1	0,495673	2,297995	2,793668	0,1551	0,08	0,000138
8	0,89977	0,351075	0,548695	1	1	0,54018	3,842692	4,382872	0,1801	0,08	0,00016
9	1,090204	0,360489	0,729715	1	1	0,575844	3,723005	4,298849	0,2051	0,08	0,000182
10	0,789377	0,46864	0,320737	1	1	0,841495	6,015416	6,856911	0,3551	0,08	0,000316
11	0,850272	0,432115	0,418157	1	1	0,51622	1,534636	2,050856	0,0801	0,08	7,12E-05
12	0,873	0,378981	0,494019	1	1	0,463086	1,285437	1,748523	0,0801	0,08	7,12E-05
13	0,340722	0,210782	0,12994	1	1	0,662387	2,924303	3,58669	0,4301	0,08	0,000382
14	0,553008	0,31922	0,233788	1	1	0,403325	0,858001	1,261327	0,0801	0,08	7,12E-05
15	1,058586	0,36852	0,690065	1	1	0,636375	4,210779	4,847154	0,2551	0,08	0,000227
16	0,564917	0,313022	0,251895	1	1	0,659627	4,409425	5,069052	0,3301	0,08	0,000293
17	0,535706	0,277954	0,257752	1	1	0,335809	0,731156	1,066965	0,0551	0,08	4,9E-05
18	0,784111	0,365932	0,418179	1	1	0,686287	2,969907	3,656195	0,3051	0,08	0,000271
19	0,520267	0,282823	0,237445	1	1	0,419428	1,782022	2,20145	0,1301	0,08	0,000116
20	0,547611	0,34292	0,204691	1	1	0,479525	1,092366	1,571891	0,1301	0,08	0,000116
21	1,063473	0,325087	0,738386	1	1	0,514192	3,398054	3,912245	0,1801	0,08	0,00016
22	0,643048	0,363719	0,279329	1	1	0,552824	2,794687	3,347511	0,1801	0,08	0,00016
23	0,599158	0,349777	0,24938	1	1	0,433882	0,915226	1,349108	0,0801	0,08	7,12E-05
24	0,202343	0,13159	0,070753	1	1	0,13159	0,070753	0,202343	0	0,08	0

Tabela F.9: Decisões do escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, detalhamento de custos e perdas para a VM 1.

h	Custo DaaS	Custo E. DaaS	Custo D. DaaS	DC E.	DC Dados	Custo A. E. DaaS	Custo A. D. DaaS	Custo A. DaaS	Custo A. Pena. Fixa	Custo IaaS	Custo A. IaaS
0	0,436618	0,397064	0,039554	1	1	0,397064	0,039554	0,436618	0	0,08	0
1	0,687246	0,491868	0,195378	1	1	0,659868	1,107142	1,76701	0,14	0,08	3,11E-05
2	1,272182	0,583924	0,688258	1	1	0,631924	1,238865	1,870789	0,04	0,08	8,89E-06
3	0,863315	0,675869	0,187446	2	2	0	0	0	0,44	0,08	0
4	1,408488	0,912887	0,495601	2	2	0	0	0	0,44	0,08	0
5	1,506049	0,619067	0,886983	1	1	0,619067	0,886983	1,506049	0	0,08	0
6	0,963951	0,58571	0,378241	1	1	0,75371	3,025926	3,779637	0,14	0,08	3,11E-05
7	0,979971	0,487013	0,492958	1	1	0,487013	0,492958	0,979971	0	0,08	0
8	1,831443	0,938851	0,892591	2	2	0	0	0	0,34	0,08	0
9	0,885	0,526429	0,35857	1	1	0,694429	2,868563	3,562992	0,14	0,08	3,11E-05
10	1,826541	1,140954	0,685587	2	2	0	0	0	0,54	0,08	0
11	1,753817	0,634907	1,11891	1	1	0,802907	4,251857	5,054764	0,14	0,08	3,11E-05
12	0,883555	0,524012	0,359543	1	1	0,524012	0,359543	0,883555	0	0,08	0
13	0,850403	0,583533	0,26687	2	2	0	0	0	0,14	0,08	0
14	1,166348	0,454834	0,711514	1	1	0,454834	0,711514	1,166348	0	0,08	0
15	0,986248	0,650329	0,335918	2	2	0	0	0	0,24	0,08	0
16	0,997086	0,457815	0,539271	1	1	0,505815	1,258298	1,764113	0,04	0,08	8,89E-06
17	1,32559	0,519578	0,806012	1	1	0,519578	0,806012	1,32559	0	0,08	0
18	0,789144	0,447438	0,341706	1	1	0,447438	0,341706	0,789144	0	0,08	0
19	1,138168	0,640302	0,497866	2	2	0	0	0	0,24	0,08	0
20	1,28333	0,427771	0,855559	1	1	0,427771	0,855559	1,28333	0	0,08	0
21	0,948613	0,662307	0,286306	2	2	0	0	0	0,24	0,08	0
22	1,240962	0,680275	0,560688	2	2	0	0	0	0,74	0,08	0
23	1,522088	0,706545	0,815543	2	2	0	0	0	0,24	0,08	0
24	0,198151	0,147508	0,050643	1	2	0,147508	0	0,147508	0	0,08	0

Tabela F.10: Decisões do escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, detalhamento de custos e perdas para a VM 2.

h	Custo DaaS	Custo E. DaaS	Custo D. DaaS	DC E.	DC Dados	Custo A. E. DaaS	Custo A. D. DaaS	Custo A. DaaS	Custo A. Pena. Fixa	Custo IaaS	Custo A. IaaS
0	0,526291	0,175967	0,350324	1	2	0,176649	0	0,176649	0,00062	0,08	2,76E-07
1	0,894847	0,269675	0,625172	2	2	0	0	0	0,15062	0,08	0
2	0,648188	0,320366	0,327821	2	2	0	0	0	0,15062	0,08	0
3	0,518634	0,274555	0,244079	2	2	0	0	0	0,50062	0,08	0
4	1,045796	0,339276	0,70652	2	2	0	0	0	0,25062	0,08	0
5	0,57587	0,260325	0,315545	1	2	0,261007	0	0,261007	0,00062	0,08	2,76E-07
6	0,597032	0,287714	0,309319	2	2	0	0	0	0,20062	0,08	0
7	0,730312	0,260168	0,470144	1	2	0,260168	0	0,260168	0	0,08	0
8	0,543607	0,33756	0,206046	2	2	0	0	0	0,20062	0,08	0
9	0,56991	0,312972	0,256937	2	2	0	0	0	0,30062	0,08	0
10	1,389854	0,353115	1,036739	2	2	0	0	0	0,45062	0,08	0
11	0,685681	0,380126	0,305555	2	2	0	0	0	0,15062	0,08	0
12	0,662019	0,229715	0,432303	1	2	0,285397	0	0,285397	0,05062	0,08	2,25E-05
13	0,669629	0,213345	0,456284	2	2	0	0	0	0,30062	0,08	0
14	0,522571	0,27175	0,250822	2	2	0	0	0	0,10062	0,08	0
15	0,437914	0,263863	0,174051	2	2	0	0	0	0,15062	0,08	0
16	0,765187	0,388151	0,377036	2	2	0	0	0	0,25062	0,08	0
17	0,589317	0,289224	0,300094	1	2	0,289906	0	0,289906	0,00062	0,08	2,76E-07
18	0,535179	0,197158	0,338021	1	2	0,25284	0	0,25284	0,05062	0,08	2,25E-05
19	0,744426	0,368676	0,37575	2	2	0	0	0	0,25062	0,08	0
20	0,437347	0,217996	0,219351	1	2	0,217996	0	0,217996	0	0,08	0
21	0,46717	0,269675	0,197494	2	2	0	0	0	0,20062	0,08	0
22	0,6039	0,291579	0,31232	2	2	0	0	0	0,50062	0,08	0
23	0,594322	0,302766	0,291556	2	2	0	0	0	0,20062	0,08	0
24	0,108581	0,097735	0,010846	3	1	0	0,010846	0,010846	0	0,08	0

Tabela F.11: Decisões do escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias, detalhamento de custos e perdas para a VM 3.

h	Custo DaaS	Custo E. DaaS	Custo D. DaaS	DC E.	DC Dados	Custo A. E. DaaS	Custo A. D. DaaS	Custo A. DaaS	Custo A. Pena. Fixa	Custo IaaS	Custo A. IaaS
0	0,676967	0,206507	0,47046	1	1	0,206507	0,47046	0,676967	0	0,08	0
1	0,633436	0,370228	0,263208	2	2	0	0	0	0,1551	0,08	0
2	0,813077	0,382688	0,430389	1	1	0,493043	1,938183	2,431227	0,1051	0,08	9,34E-05
3	1,019476	0,34685	0,672626	2	2	0	0	0	0,3051	0,08	0
4	0,888598	0,542039	0,346559	2	2	0	0	0	0,2301	0,08	0
5	0,790979	0,374953	0,416026	1	1	0,380308	0,48675	0,867058	0,0051	0,08	4,53E-06
6	1,493484	0,617309	0,876175	2	2	0	0	0	0,2051	0,08	0
7	0,724955	0,462477	0,262478	2	2	0	0	0	0,1551	0,08	0
8	1,038557	0,489863	0,548695	2	2	0	0	0	0,1801	0,08	0
9	1,233698	0,503984	0,729715	2	2	0	0	0	0,2051	0,08	0
10	0,986947	0,66621	0,320737	2	2	0	0	0	0,3551	0,08	0
11	0,850272	0,432115	0,418157	1	1	0,51622	1,534636	2,050856	0,0801	0,08	7,12E-05
12	0,873	0,378981	0,494019	1	1	0,463086	1,285437	1,748523	0,0801	0,08	7,12E-05
13	0,409364	0,279424	0,12994	2	2	0	0	0	0,4301	0,08	0
14	0,553008	0,31922	0,233788	1	1	0,403325	0,858001	1,261327	0,0801	0,08	7,12E-05
15	1,206096	0,51603	0,690065	2	2	0	0	0	0,2551	0,08	0
16	0,684677	0,432782	0,251895	2	2	0	0	0	0,3301	0,08	0
17	0,535706	0,277954	0,257752	1	1	0,335809	0,731156	1,066965	0,0551	0,08	4,9E-05
18	0,930327	0,512148	0,418179	2	2	0	0	0	0,3051	0,08	0
19	0,624929	0,387484	0,237445	2	2	0	0	0	0,1301	0,08	0
20	0,68232	0,47763	0,204691	2	2	0	0	0	0,1301	0,08	0
21	1,189266	0,45088	0,738386	2	2	0	0	0	0,1801	0,08	0
22	0,788158	0,508829	0,279329	2	2	0	0	0	0,1801	0,08	0
23	0,599158	0,349777	0,24938	1	1	0,433882	0,915226	1,349108	0,0801	0,08	7,12E-05
24	0,202343	0,13159	0,070753	1	1	0,13159	0,070753	0,202343	0	0,08	0