



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

Thiago Teixeira Sá

**CloudReports: uma extensão gráfica do framework
CloudSim para a simulação de ambientes de
computação distribuída baseados no paradigma da
Computação em Nuvem**

FORTALEZA – CEARÁ
DEZEMBRO 2010

Autor:

Thiago Teixeira Sá

Orientador:

Prof. Danielo Gonçalves Gomes, Dr.

CloudReports: uma extensão gráfica do framework CloudSim para a
simulação de ambientes de computação distribuída baseados no
paradigma da Computação em Nuvem

Monografia de Conclusão de Curso
apresentada à Coordenação do Curso
de Graduação em Engenharia de
Teleinformática da Universidade
Federal do Ceará como parte dos
requisitos para obtenção do grau de
Engenheiro de Teleinformática.

FORTALEZA – CEARÁ

DEZEMBRO 2010

THIAGO TEIXEIRA SÁ

**CloudReports: uma extensão gráfica do framework CloudSim para a
simulação de ambientes de computação distribuída baseados no
paradigma da Computação em Nuvem**

Esta Monografia foi julgada adequada para a obtenção do diploma de Engenheiro do Curso de Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

Thiago Teixeira Sá

Banca Examinadora:

Prof. Daniello Gonçalves Gomes, Dr.
Orientador

Prof. José Marques Soares, Dr.
Universidade Federal do Ceará

Prof. Cideley Teixeira de Souza, Dr.
Instituto Federal de Educação, Ciência e
Tecnologia do Ceará

Fortaleza, 10 de Dezembro de 2010

Resumo

Esta monografia descreve a criação de um conjunto de extensões para o *framework* de simulação CloudSim, visando a concepção de um simulador gráfico para ambientes computacionais distribuídos baseados no paradigma da Computação em Nuvem. Através desta nova ferramenta, usuários sem experiência em programação podem criar um número virtualmente ilimitado de cenários de simulação por meio de uma interface gráfica de fácil utilização e realizar simulações através dos recursos oferecidos pelo *framework* CloudSim. Os cenários são compostos por um provedor de Infraestrutura como Serviço (IaaS) e por uma quantidade arbitrária de clientes, que utilizam os recursos computacionais disponíveis para a alocação de máquinas virtuais e a execução de tarefas. O simulador oferece flexibilidade para a criação de *datacenters* com quantidade variável de recursos, incluindo, ainda, a customização individual dos *hosts* que o compõem. Este aspecto flexível também é aplicado à criação de clientes, dado que os mesmos podem dispor de máquinas virtuais com configurações distintas e inteiramente customizáveis. Ademais, a funcionalidade de geração de relatórios facilita a compreensão dos resultados de simulação por meio de gráficos explanativos e listagens de características dos ambientes simulados. Os testes iniciais com a ferramenta indicam resultados satisfatórios, que, atrelados ao grande potencial atribuído ao campo da Computação em Nuvem, podem significar uma importante contribuição para as atividades de pesquisa nesta área.

Palavras-chave: Computação em Nuvem, Sistemas Distribuídos, Virtualização, Ferramentas de simulação.

Abstract

This monograph describes the development of a set of extensions for the CloudSim simulation framework towards the creation of a graphic simulator for distributed computing environments based on the Cloud Computing paradigm. It is designed so that users with no programming skills can create a virtually unlimited number of environments through an easy-to-use graphic user interface and perform simulations by using the features provided by the CloudSim framework. These environments are composed by an Infrastructure as a Service (IaaS) provider and an arbitrary amount of clients, which use the available computing resources in order to deploy virtual machines and execute tasks. The simulator provides flexibility to create datacenters with a variable amount of resources. It also supports customization of each one of the hosts individually. This flexibility also applies to the creation of clients, given that they can own virtual machines with different and entirely customizable configurations. Moreover, the reports generation feature makes it easier to understand the simulation results by displaying explanatory charts and lists of the simulated environments' characteristics. The initial tests with the tool indicate satisfactory results, which, combined with the high potential attributed to the Cloud Computing field, might represent an important contribution to research activities on this subject.

Keywords: Cloud Computing, Distributed Systems, Virtualization, Simulation tools.

Este trabalho é dedicado integralmente à minha família,
a quem tudo devo.

Agradecimentos

Ao meu pai Luiz, pelos valores ensinados, pela paciência e pelo apoio imprescindível durante os longos anos de graduação.

Às minhas tias Meire Céli, Maria e Marlúcia, pelo suporte e amor recebidos desde a infância.

Aos meus irmãos André e Lucas, pela amizade e companheirismo.

À minha namorada Ingrid, pela compreensão e apoio incondicional em todos os momentos.

Ao Professor Danielo Gonçalves Gomes, pela confiança e incentivo durante todo o processo de orientação acadêmica.

Aos colegas do curso de graduação de Engenharia de Teleinformática e do Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat), que em inúmeras ocasiões influenciaram positivamente a execução deste trabalho.

*A viagem não acaba nunca. Só os viajantes acabam.
E mesmo estes podem prolongar-se em memória,
em lembrança, em narrativa.
Quando o visitante sentou na areia da praia e disse:
“Não há mais o que ver”, saiba que não era assim.
O fim de uma viagem é apenas o começo de outra.
É preciso ver o que não foi visto,
ver outra vez o que se viu já,
ver na primavera o que se vira no verão,
ver de dia o que se viu de noite,
com o sol onde primeiramente a chuva caía,
ver a seara verde, o fruto maduro,
a pedra que mudou de lugar,
a sombra que aqui não estava.
É preciso voltar aos passos que foram dados,
para repetir e para traçar caminhos novos ao lado deles.
É preciso recomeçar a viagem.
Sempre.*

José Saramago

Sumário

Lista de Figuras	x
Lista de Tabelas	xi
Lista de Algoritmos	xii
Lista de Siglas	xii
1 Introdução	1
1.1 Contextualização	1
1.2 Revisão bibliográfica	2
1.3 Objetivos	3
1.3.1 Objetivo geral	3
1.3.2 Objetivos específicos	3
1.4 Metodologia utilizada	4
1.5 Estrutura da monografia	5
2 Computação em Nuvem	6
2.1 Introdução	6
2.2 Contexto histórico	8
2.3 Arquitetura	11
2.3.1 Modelos relacionados	11
2.3.1.1 Sistemas distribuídos	11

2.3.1.2	Grades computacionais	13
2.3.1.3	Computação Utilitária	14
2.3.2	Principais tecnologias	15
2.3.2.1	Virtualização	15
2.3.2.2	<i>Web services</i>	17
2.4	Modelos de serviço	18
2.4.1	Infraestrutura como Serviço	19
2.4.2	Plataforma como Serviço	22
2.4.3	<i>Software</i> como Serviço	22
2.5	Modos de implantação	24
2.5.1	Nuvem pública	24
2.5.2	Nuvem privada	24
2.5.3	Nuvem comunitária	25
2.5.4	Nuvem híbrida	25
2.6	Segurança	25
2.7	<i>Middlewares</i> e <i>framework</i> de simulação	27
2.7.1	CloudSim	27
2.8	Perspectivas futuras	30
2.9	Resumo	31
3	CloudReports	32
3.1	Material	32
3.1.1	Ferramentas de programação	32
3.1.2	CloudSim	33
3.1.3	JFreeChart	33
3.2	Análise descritiva	34
3.3	Ambiente de simulação	35
3.4	Arquitetura	36
3.4.1	Extensões	38
3.4.2	Gerência de simulação	39
3.4.3	Registro de entidades	39

3.4.4	Geração de relatórios	40
3.4.5	Interface Gráfica de Usuário	41
3.5	Entidades funcionais	41
3.6	Algoritmos de decisão	41
3.6.1	<i>Cheapest Datacenter</i>	42
3.6.2	<i>Fastest Datacenter</i>	43
3.6.3	<i>Lowest Latency</i>	43
3.7	Resumo	44
4	Resultados	45
4.1	Cenário simulado	45
4.1.1	Provedor	46
4.1.2	Clientes	50
4.2	Resultados obtidos	54
4.2.1	Máquinas virtuais alocadas por <i>datacenter</i>	54
4.2.2	<i>Cloudlets</i> executadas por <i>datacenter</i>	55
4.2.3	Custos por <i>datacenter</i>	56
4.2.4	Recursos utilizados por <i>datacenter</i>	56
4.2.5	<i>Cloudlets</i> executadas por máquina virtual	57
4.2.6	Recursos utilizados por máquina virtual	58
4.2.7	Tempo de execução por <i>cloudlet</i>	59
4.3	Resumo	61
5	Conclusões	63
5.1	Contribuições	63
5.2	Limitações	64
5.3	Perspectivas para trabalhos futuros	65
	Referências Bibliográficas	69

Lista de Figuras

1.1	Metodologia utilizada para a execução deste trabalho.	4
2.1	O Motor Analítico de Babbage (MUSEUM, 2010)	9
2.2	Um sistema distribuído típico. Adaptada de (TANENBAUM, 2006) .	12
2.3	Tipos de virtualização. Adaptada de (SOLTER <i>et al.</i> , 2009)	17
2.4	Comunicação entre cliente e nuvem através de uma interface de <i>web service</i>	18
2.5	Visão em camadas dos modelos de serviço da Computação em Nuvem. Adaptada de (STANOEVSKA-SLABEVA <i>et al.</i> , 2010)	19
2.6	Arquitetura em camadas do CloudSim. (CALHEIROS <i>et al.</i> , 2010b) .	28
2.7	Escalonamento de máquinas virtuais e <i>cloudlets</i> de acordo com o tipo de compartilhamento de recursos: (a) Domínio do espaço para máquinas virtuais e <i>cloudlets</i> , (b) Domínio do espaço para máquinas virtuais e domínio do tempo para <i>cloudlets</i> , (c) Domínio do tempo para máquinas virtuais e domínio do espaço para <i>cloudlets</i> e (d) Domínio do tempo para máquinas virtuais e <i>cloudlets</i> . Adaptada de (CALHEIROS <i>et al.</i> , 2009)	29
2.8	Fluxo de comunicação entre entidades do CloudSim. Adaptada de (CALHEIROS <i>et al.</i> , 2010b)	30
3.1	Tela inicial do CloudReports.	35
3.2	Ambiente de simulação típico do CloudReports.	37
3.3	Estrutura em módulos do CloudReports.	37
3.4	Diagrama UML das classes que implementam políticas de <i>broker</i> . . .	38
3.5	Diagrama UML das classes que compõem um registro de <i>datacenter</i> . .	39

3.6	Diagrama UML das classes que compõem um registro de cliente. . . .	40
4.1	Tela de exibição do <i>log</i> de simulação.	54
4.2	Máquinas virtuais alocadas por <i>datacenter</i>	55
4.3	<i>Cloudlets</i> executadas por <i>datacenter</i>	56
4.4	Custos produzidos nos <i>datacenters</i> simulados.	57
4.5	Recursos utilizados nos <i>datacenters</i> de (a) São Paulo, (b) Rio de Janeiro, (c) Brasília e (d) Fortaleza.	58
4.6	<i>Cloudlets</i> executadas pelas máquinas virtuais do (a) Cliente 1, (b) Cliente 2, (c) Cliente 3 e (d) Cliente 4.	59
4.7	Recursos utilizados pelas máquinas virtuais (a) VM0, (b) VM1, (c) VM2 e (d) VM3 do Cliente 3.	60
4.8	Tempos de execução de <i>cloudlets</i> do Cliente 1.	61
4.9	Tempos de execução de <i>cloudlets</i> do (a) Cliente 2 e (b) Cliente 3. . .	62
4.10	Tempos de execução de <i>cloudlets</i> do Cliente 4.	62

Lista de Tabelas

2.1	Tipos de instâncias oferecidas pelo Amazon EC2 (AMAZON, 2010).	21
2.2	Obstáculos para o crescimento e a adoção da Computação em Nuvem (STANOEVSKA-SLABEVA <i>et al.</i> , 2010)	26
4.1	Características dos <i>datacenters</i> simulados.	47
4.2	Características dos <i>hosts</i> simulados.	48
4.3	Características dos enlaces cuja fonte é o <i>datacenter</i> localizado em Fortaleza.	48
4.4	Características dos enlaces cuja fonte é o <i>datacenter</i> localizado em Brasília.	49
4.5	Características dos enlaces cuja fonte é o <i>datacenter</i> localizado no Rio de Janeiro.	49
4.6	Características dos enlaces cuja fonte é o <i>datacenter</i> localizado em São Paulo.	50
4.7	Configurações das máquinas virtuais dos clientes simulados.	51
4.8	Perfis de utilização de recursos dos clientes simulados.	52
4.9	Características dos enlaces cuja fonte é o Cliente 1.	52
4.10	Características dos enlaces cuja fonte é o Cliente 2.	53
4.11	Características dos enlaces cuja fonte é o Cliente 3.	53
4.12	Características dos enlaces cuja fonte é o Cliente 4.	53

Lista de Algoritmos

3.1	Política de decisão do <i>broker Cheapest Datacenter</i>	42
3.2	Política de decisão do <i>broker Fastest Datacenter</i>	43
3.3	Política de decisão do <i>broker Lowest Latency</i>	43

Lista de Siglas

- QoS** *Quality of Service* (Qualidade de Serviço)
- IaaS** *Infrastructure as a Service* (Infra-estrutura como Serviço)
- PaaS** *Platform as a Service* (Plataforma como Serviço)
- SaaS** *Software as a Service* (*Software* como Serviço)
- API** *Application Programming Interface* (Interface de Programação de Aplicativos)
- PC** *Personal Computer* (Computador Pessoal)
- VLSI** *Very Large Scale Integration* (Integração em Escala Muito Alta)
- PDA** *Personal Digital Assistant* (Assistentes Digitais Pessoais)
- CAD** Computação de Alto Desempenho
- XML** *Extensible Markup Language* (Linguagem de Marcação Extensível)
- SOA** *Service-Oriented Architecture* (Arquitetura Orientada a Serviços)
- SLA** *Service Level Agreement* (Acordo de Nível de Serviço)
- CaaS** *Communication as a Service* (Comunicação como Serviço)
- MaaS** *Management as a Service* (Gerenciamento como Serviço)
- DaaS** *Data as a Service* (Dados como Serviço)
- ITaaS** *Information Technology as a Service* (Tecnologia da Informação como Serviço)
- XaaS** *Everything as a Service* (Tudo como Serviço)
- EC2** *Elastic Compute Cloud*
- EBS** *Elastic Block Store*

MTTR *Mean Time To Repair* (Tempo Médio Para Reparo)

OCBCL *Oracle Corporation Binary Code License*

CDDL *Common Development and Distribution License*

GPL *General Public License*

JVM *Java Virtual Machine* (Máquina Virtual Java)

IDE *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)

MIPS *Millions of Instructions Per Second* (Milhões de Instruções por Segundo)

GUI *Graphical User Interface* (Interface Gráfica de Usuário)

SAN *Storage Area Network* (Área de Armazenamento em Rede)

Capítulo 1

Introdução

Este capítulo visa apresentar o contexto no qual o trabalho realizado está inserido, assim como definir seus objetivos e justificar seus propósitos. Na Seção 1.1, é apresentada uma contextualização do problema tratado. A Seção 1.2 discute as principais fontes bibliográficas que traduzem o estado da arte no que concerne à simulação de ambientes computacionais em nuvem. Os objetivos geral e específicos são indicados na Seção 1.3. Por fim, a Seção 1.4 realiza uma breve descrição da metodologia empregada e a Seção 1.5 descreve o formato no qual esta monografia está organizada.

1.1 Contextualização

A Computação em Nuvem propõe a integração de modelos tecnológicos para o provimento de infraestrutura de *hardware*, plataformas de desenvolvimento e aplicações na forma de serviços disponíveis remotamente e em escala global. Neste novo paradigma de utilização de recursos computacionais, clientes abrem mão da administração de uma infraestrutura própria e dispõem de serviços oferecidos por terceiros, delegando responsabilidades e assumindo custos estritamente proporcionais à quantidade de recursos que utilizam.

Os ambientes computacionais em nuvem caracterizam-se por serem intrinsecamente distribuídos e compostos por recursos heterogêneos, além de servirem concomitantemente a uma grande diversidade de clientes com requisitos de Qualidade de Serviço (QoS) distintos. Ademais, os *datacenters* que compõem uma nuvem de recursos devem suportar um número virtualmente ilimitado de

aplicações, o que os obriga a lidar com demandas variantes por processamento, armazenamento de dados e utilização de banda de comunicação.

Dado o elevado potencial e complexidade deste cenário, e tendo em vista a crescente adoção de serviços em nuvem observada no mercado e a intensa atividade de pesquisa acadêmica nesta área, torna-se evidente a necessidade de ferramentas que viabilizem uma análise detalhada do funcionamento destes ambientes e provejam meios para o desenvolvimento de técnicas que aperfeiçoem o emprego dos recursos computacionais envolvidos.

Como a utilização de *testbeds* reais limita a realização de experimentos do ponto de vista da escalabilidade e torna difícil a reprodução dos resultados obtidos (CLOUDSIM, 2010), faz-se necessária a existência de alternativas para a simulação de ambientes computacionais em nuvem através das quais a validação de novas técnicas seja realizada de forma escalável e facilmente reproduzível. O emprego de ferramentas de simulação elimina ainda a necessidade de uma infraestrutura real atrelada aos experimentos e todos os custos relacionados à utilização da mesma.

1.2 Revisão bibliográfica

O campo da Computação em Nuvem tem recebido grande atenção da academia desde sua eclosão. Devido a seu caráter recente, muitas informações conflitantes são frequentemente encontradas na literatura. Contudo, seu grande potencial e o papel chave que certamente desempenhará em um futuro próximo em diversas áreas da tecnologia é tratado no mundo acadêmico de forma próxima à unanimidade.

De acordo com Armbrust *et al.* (2009), a Computação em Nuvem “trata-se do sonho de longa data da computação como bem utilitário e tem o potencial para transformar grande parte da indústria de Tecnologia da Informação (TI), tornando o software ainda mais atrativo oferecido como um serviço”. Alguns dos indícios de que a Computação em Nuvem representa o novo paradigma que tornará possível a transformação definitiva dos recursos computacionais em bens utilitários são discutidos em Buyya, Yeo e Venugopal (2008a), Buyya *et al.* (2008b) e Buyya (2009).

Uma alternativa para a simulação de ambientes de computação distribuída é apontada em Legrand, Marchal e Casanova (2003), que trata do framework de simulação de grades computacionais SimGrid. Em Sulistio *et al.* (2008), uma extensão à ferramenta GridSim é apresentada. Também voltado para a simulação

de grades computacionais, o GridSim é o precursor do *framework* de simulação empregado neste trabalho, denominado CloudSim e com sua mais recente versão apresentada em Calheiros, Buyya e Rose (2010a).

Por fim, as oportunidades e desafios que surgem com o advento da simulação de ambientes computacionais em nuvem são discutidos em Buyya, Ranjan e Calheiros (2009), e alguns dos resultados já obtidos através do uso destas ferramentas são mostrados em Beloglazov e Buyya (2010), Calheiros *et al.* (2010b) e Kim, Beloglazov e Buyya (2009).

1.3 Objetivos

Esta seção visa descrever os objetivos deste trabalho através de um panorama geral de seus propósitos e da descrição de pontos específicos que devem ser atendidos.

1.3.1 Objetivo geral

O principal objetivo deste trabalho consiste na criação de uma extensão do *framework* CloudSim, visando a concepção de um simulador gráfico para ambientes de computação distribuída baseados no paradigma da Computação em Nuvem. Através desta nova ferramenta, estudantes e profissões da área poderão realizar experimentos e desenvolver novas técnicas que visem uma maior eficiência no emprego dos recursos computacionais existentes nestes ambientes. Um dos requisitos para o simulador consiste em uma interface gráfica de fácil utilização e de recursos para a geração de relatórios de simulação de forma automatizada. Estes relatórios terão como meta facilitar a compreensão dos resultados obtidos e serão compostos por gráficos ilustrativos e listagens de características dos ambientes simulados.

1.3.2 Objetivos específicos

Os objetivos específicos deste trabalho são enumerados a seguir:

- i. Análise das soluções existentes para a simulação de sistemas computacionais em nuvem;
- ii. Estudo do funcionamento e identificação de falhas e possíveis melhorias relativas ao *framework* de simulação escolhido;
- iii. Modelagem de uma interface gráfica de fácil utilização;

- iv. Criação de um módulo de *software* para o registro das entidades criadas por meio da interface gráfica do simulador;
- v. Implementação de políticas de decisão para a alocação de máquinas virtuais em *datacenters*;
- vi. Implementação de melhorias para o *framework* de simulação utilizado;
- vii. Criação de um recurso de geração automática de relatórios de simulação;
- viii. Integração da aplicação desenvolvida com o *framework* de simulação utilizado.

1.4 Metodologia utilizada

A metodologia empregada para a execução deste trabalho pode ser descrita na forma de um processo composto pelas seis etapas ilustradas na Figura 1.1.



Figura 1.1: Metodologia utilizada para a execução deste trabalho.

Primeiramente, uma revisão bibliográfica referente à área da Computação em Nuvem foi realizada com o intuito de familiarização com o estado da arte no que concerne às tecnologias de simulação de ambientes computacionais que seguem esse modelo. Em seguida, deu-se início a uma pesquisa por *softwares* que possibilitassem a simulação desses ambientes, quando se constatou a ausência de aplicações gráficas com esta função. O passo seguinte consistiu na escolha das ferramentas que seriam utilizadas para a implementação do simulador, descritas na Seção 3.1. Após esta seleção, a codificação do simulador foi realizada. Nas duas etapas posteriores, testes e melhorias se deram de forma concomitante. O último passo do processo, mostrado em linha tracejada na Figura 1.1, propõe a comparação dos resultados obtidos por meio do simulador com dados extraídos de experimentos práticos. A forma como esta etapa será realizada ainda não foi inteiramente definida.

1.5 Estrutura da monografia

Esta monografia está organizada em cinco capítulos, incluindo-se a Introdução aqui apresentada. O Capítulo 2 aborda os conceitos gerais que definem o novo paradigma de computação distribuída denominado Computação em Nuvem e constitui o embasamento teórico deste trabalho. No Capítulo 3, a aplicação desenvolvida é abordada através da apresentação das ferramentas utilizadas para sua criação e de descrições do ambiente de simulação, dos módulos estruturais que a compõem e das entidades funcionais existentes. O Capítulo 4 analisa os resultados obtidos através do uso do simulador, enquanto o Capítulo 5 apresenta considerações finais acerca do trabalho realizado.

Capítulo 2

Computação em Nuvem

Neste capítulo, serão abordados aspectos que definem o campo de pesquisa e negócios hoje intitulado Computação em Nuvem. As Seções 2.1 e 2.2 trazem definições e uma contextualização histórica, enquanto a Seção 2.3 analisa os elementos que compõem a arquitetura deste modelo. Em seguida, são abordados os principais modelos de serviço e os modos de implantação existentes. As seções finais discutem aspectos relacionados à segurança e perspectivas futuras para a área.

2.1 Introdução

O grande crescimento da conectividade de usuários e empresas em todo o mundo atrelado à evolução da rede mundial de computadores de um simples meio de comunicação a uma plataforma real de negócios e redes sociais tornou necessária a criação de ambientes computacionais capazes de gerenciar dinamicamente uma demanda exponencial por dados e serviços *on-line*. A solução para esta problemática adveio do surgimento de um novo campo de pesquisa e negócios hoje intitulado Computação em Nuvem.

Embora não raro mencionada equivocadamente como uma tecnologia específica, a Computação em Nuvem constitui um ponto de convergência entre modelos tecnológicos diversos. Trata-se de um campo multidisciplinar onde elementos como virtualização, grades computacionais, sistemas distribuídos, técnicas de armazenamento de dados, segurança, serviços web e *utility computing* se unem para a formação de um ambiente computacional complexo, cujo potencial atrai, com igual interesse, o mundo corporativo e a academia.

A origem do termo Computação em Nuvem é controversa. De acordo com (MALIS, 1993), o emprego técnico do termo “nuvem” data dos anos 1990 em referência à capacidade de comutação dinâmica de circuitos para o balanceamento de utilização das redes de telecomunicações, o que reiterava seu caráter virtualizado. No ano de 2001, segundo matéria publicada pelo jornal americano *The New York Times*, a Microsoft utilizou o termo durante uma apresentação sobre o *.Net Framework* para designar o conjunto de computadores que constituem a Internet. Há ainda a suposição de que o termo tenha surgido em 1961 durante uma apresentação realizada por John McCarthy, onde o mesmo previu a utilização de recursos computacionais como um bem utilitário através do uso compartilhado de computadores e aplicações no domínio do tempo (RITTINGHOUSE; RANSOME, 2010).

Segundo o *National Institute of Standards and Technology* (NIST), agência vinculada ao Departamento de Comércio dos Estados Unidos da América e responsável pelo desenvolvimento e aplicação de normas junto à indústria, a Computação em Nuvem é “um modelo que possibilita o acesso via rede de forma conveniente e sob demanda a um conjunto compartilhado de recursos computacionais (como redes, servidores, armazenamento, aplicações e serviços) que podem ser rapidamente providos e liberados com interação com o provedor de serviços e esforço de gerenciamento mínimos. O modelo promove a disponibilidade e é composto por cinco características essenciais, três modelos de serviço, e quatro modos de implantação”.

Embora o texto supracitado seja amplamente aceito, o grande dinamismo e intensa atividade de pesquisa característicos da área fazem com que definições diversas e muitas vezes conflitantes sejam encontradas na literatura. Os três modelos de serviço e modos de implantação serão abordados em seções específicas deste capítulo. As cinco características essenciais são descritas a seguir:

- i. ***Self-service sob demanda:*** O usuário pode alocar recursos computacionais unilateralmente sem a necessidade de interação humana com o provedor do serviço. Empresas que atuam na área de (IaaS) oferecem uma Interface de Programação de Aplicativos (API) própria, que pode ser utilizada para a requisição dinâmica de recursos através de *scripts* personalizados.
- ii. **Amplo acesso à rede:** Infraestrutura computacional, plataformas de desenvolvimento e aplicações são acessadas via rede através de protocolos

padrão. Isto possibilita a utilização dos serviços por máquinas clientes que variam de *desktops* robustos a dispositivos móveis com severas limitações de recursos.

- iii. **Pooling de recursos:** O provedor detém um conjunto de recursos físicos e virtuais que são alocados para usuários e liberados pelos mesmos de forma dinâmica e de acordo com a demanda existente. A alocação de um mesmo recurso é muitas vezes feita para mais de um usuário simultaneamente, prática comumente definida como multi-arrendamento¹. Ademais, o usuário não possui controle ou conhecimento preciso do ponto geográfico de origem do serviço que está utilizando.
- iv. **Elasticidade:** Os recursos providos por um ambiente computacional em nuvem são inerentemente escaláveis. O termo elasticidade é utilizado para transmitir a ideia de que o usuário pode, a qualquer momento, aumentar ou diminuir a quantidade de recursos utilizados. Essa característica cria a ilusão de que os recursos oferecidos são ilimitados e que o usuário pode fazer uso da quantidade que lhe for conveniente.
- v. **Medição de serviços:** Os recursos oferecidos podem ser monitorados, controlados e reportados ao provedor e ao usuário de forma transparente. Este mecanismo permite que sejam cobrados valores referentes ao grau de utilização dos recursos e estipulados por meio de contrato entre provedor e usuário.

2.2 Contexto histórico

É importante compreender a evolução dos sistemas de computação até o momento de eclosão da Computação em Nuvem. A primeira evidência de uma arquitetura computacional foi proposta por Charles Babbage no ano de 1856. Tratava-se de um sistema mecânico movido a uma manivela e que, embora projetado no século XIX, só foi completamente implementado 153 anos depois, em 2002, na cidade de Londres (MUSEUM, 2010). O chamado Motor Analítico (Figura 2.1) representa, ainda hoje, a forma mais básica de arquitetura dos sistemas empregados nos computadores digitais modernos (BREY, 2008). Até anos recentes, quando o trabalho de Babbage ainda não havia sido redescoberto, esta arquitetura era

¹O termo multi-arrendamento é uma tradução livre para a expressão inglesa *multi-tenancy*.

atribuída a John Von Neumann, de onde provém a popular expressão “arquitetura Von Neumann”.

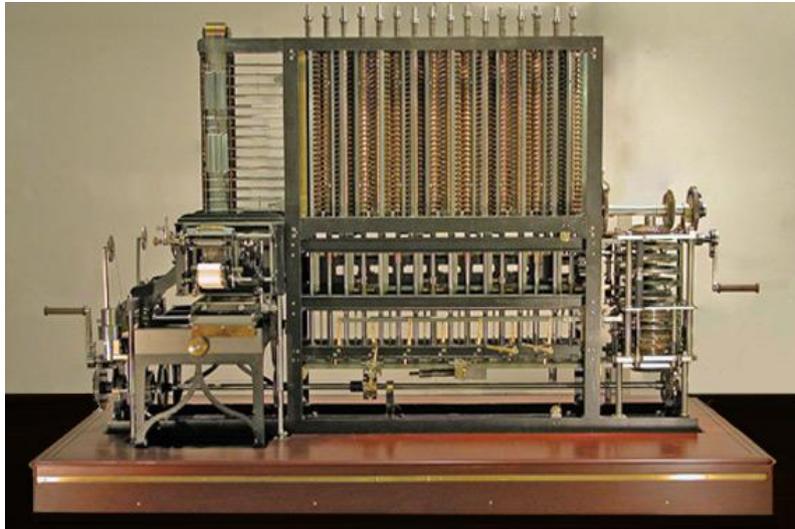


Figura 2.1: O Motor Analítico de Babbage (MUSEUM, 2010)

Segundo (RITTINGHOUSE; RANSOME, 2010), a primeira geração de computadores remonta ao ano de 1943, quando a IBM financiou a criação do Mark I, um computador programável eletromecânico de propósito geral desenvolvido pela Universidade de Harvard. No mesmo ano, em um projeto sob a liderança de Alan Turing, o primeiro dispositivo de computação programável, eletrônico e digital, chamado Colossos, foi concebido na Inglaterra. Era o tempo das válvulas, do armazenamento de dados através de cartões perfurados e de computadores que ocupavam grandes galpões, apesar de oferecerem apenas uma pequena fração do poder computacional de um computador pessoal atual.

A invenção dos transistores marcou o surgimento da segunda geração de computadores. A substituição das válvulas tornava possível a construção de dispositivos menores e mais confiáveis, que predominaram nas décadas de 1950 e 1960. Todavia, ainda se tratavam de máquinas grandes e de custo muito elevado, o que limitava sua utilização a universidades e órgãos governamentais.

Com o surgimento dos circuitos integrados, deu-se início à terceira geração de computadores. As máquinas desta geração, chamadas minicomputadores, eram mais baratas e compactas, tornando possível sua utilização por empresas de menor porte.

O advento dos microprocessadores na década de 1970 fez com que surgissem os computadores que caracterizam a quarta geração. O chamado computador

peçoal (PC) oferecia maior capacidade de processamento, armazenamento de dados, e tamanho bastante reduzido. A integração de circuitos em larga escala (VLSI) colaborou para a massificação comercial destes dispositivos, tornando-os ainda menores, mais baratos e possibilitando o surgimento de uma geração de indivíduos que estaria habituada a lidar com estas máquinas como parte integrante, e muitas vezes imprescindível, de sua vida diária.

A popularização da Internet na década de 1990 e o rápido crescimento da produção e comercialização de dispositivos computacionais móveis foram fatores determinantes para a formação do panorama tecnológico observado na atualidade. À medida que a conectividade dos indivíduos aumentava, uma poderosa plataforma de serviços surgia inteiramente *on-line*. Inicialmente utilizados apenas através de computadores pessoais, estes serviços passaram a ser acessíveis através de telefones celulares, Assistentes Digitais Pessoais (PDA) e um conjunto incrivelmente diversificado de equipamentos eletrônicos. A tecnologia começava a embutir-se onde antes não se imaginava, concretizando o conceito da Computação Ubíqua, termo empregado pela primeira vez por Mark Weiser ainda na década de 1980.

Deste modo, ao final da década de 1990, formava-se um ambiente propício ao surgimento da Computação em Nuvem. A grande quantidade de serviços *web* disponíveis e a necessidade de acessá-los através de dispositivos com recursos físicos cada vez mais limitados somaram-se ao avanço das tecnologias de virtualização de hardware para a promoção de uma evolução na forma de lidar com o consumo de recursos computacionais e com o desenvolvimento e utilização de *software*.

No século XIX, logo após a publicação da obra *A Origem das Espécies*, de Charles Darwin, o zoologista alemão Ernst Haeckel afirmou que “a ontogenia recapitula a filogenia” (TANENBAUM, 2001), o que mais tarde ficaria conhecido como a Teoria da Recapitulação. Com isto, Haeckel propunha que o desenvolvimento de um indivíduo da fecundação à idade adulta se assemelha ao desenvolvimento de sua espécie ao transcorrer de gerações. Algo similar pode ser observado ao se analisar o processo de evolução dos sistemas de computação. Dos antigos *mainframes* ao processo de virtualização de grandes servidores que culminou recentemente no surgimento da Computação em Nuvem, cada geração de computadores procurou se adaptar às necessidades do mercado como uma questão de sobrevivência, que, neste caso, significava a manutenção de sua viabilidade comercial. A analogia se completa ao se observar que, assim como, segundo a teoria de Darwin, as mudanças no

ambiente selecionavam os seres mais aptos e promoviam a criação de novas espécies, a evolução das tecnologias da computação é ditada pelo surgimento de problemáticas que exigem modelos inovadores e aptos a solucioná-las. Por conseguinte, sob o ponto de vista presente ao longo deste documento, a Computação em Nuvem, ao contrário do que muitos apregoam, não se trata de uma revolução, mas de uma evolução tecnológica natural.

2.3 Arquitetura

A Computação em Nuvem é um ponto de convergência entre modelos tecnológicos diversos. Da combinação destes componentes, surge um ambiente computacional complexo e apto a solucionar um vasto conjunto de problemas advindos tanto da academia quanto do mundo corporativo. Nesta seção, serão discutidos os modelos relacionados a este ambiente e suas principais tecnologias.

2.3.1 Modelos relacionados

2.3.1.1 Sistemas distribuídos

O emprego do termo nuvem remete à utilização de serviços de infraestrutura de *hardware* e *software* provindos de um ambiente capaz de oferecer recursos virtualmente ilimitados e de forma “elástica”. Embora não fique claro ao usuário final, este ambiente é formado por um conjunto que *datacenters* que, unidos, tornam possível o *pooling* de recursos, que, juntamente com a elasticidade, constitui uma das principais características da Computação em Nuvem. O aspecto coletivo do funcionamento destes *datacenters* caracteriza o que é denominado um sistema computacional distribuído.

Um sistema distribuído é definido como um conjunto de máquinas independentes que se apresentam ao usuário como um sistema único e consistente (TANENBAUM, 2006) - ou seja, apesar dos componentes de *hardware* serem autônomos, uma camada de *software* exerce a função de oferecê-los como uma única máquina ao usuário final. Já segundo (COULOURIS *et al.*, 2005), os sistemas distribuídos são computadores que se comunicam e coordenam suas ações via rede através da troca de mensagens. Em ambos os casos, um conjunto de máquinas, heterogêneo ou não, age coletivamente para a execução de um objetivo comum, e quem o utiliza sempre o faz seguindo o conceito de unidade que caracteriza o sistema.

Duas das principais vertentes dos sistemas computacionais distribuídos são a computação em *clusters* e as grades computacionais. No primeiro, um agrupamento de máquinas com o mesmo sistema operacional é conectado localmente, enquanto o último é formado por máquinas provenientes de diferentes organizações e indivíduos e que, embora integradas como um único sistema, estão inseridas em domínios administrativos diferentes e possuem características de *hardware* e *software* nitidamente heterogêneas (TANENBAUM, 2006).

A Figura 2.2 ilustra um sistema distribuído típico, no qual a camada de *middleware* é composta pelo *software* responsável pelo provimento de uma interface de comunicação única e pela coordenação de usuários e aplicações para a utilização de máquinas distintas, cada uma com seu próprio sistema operacional.

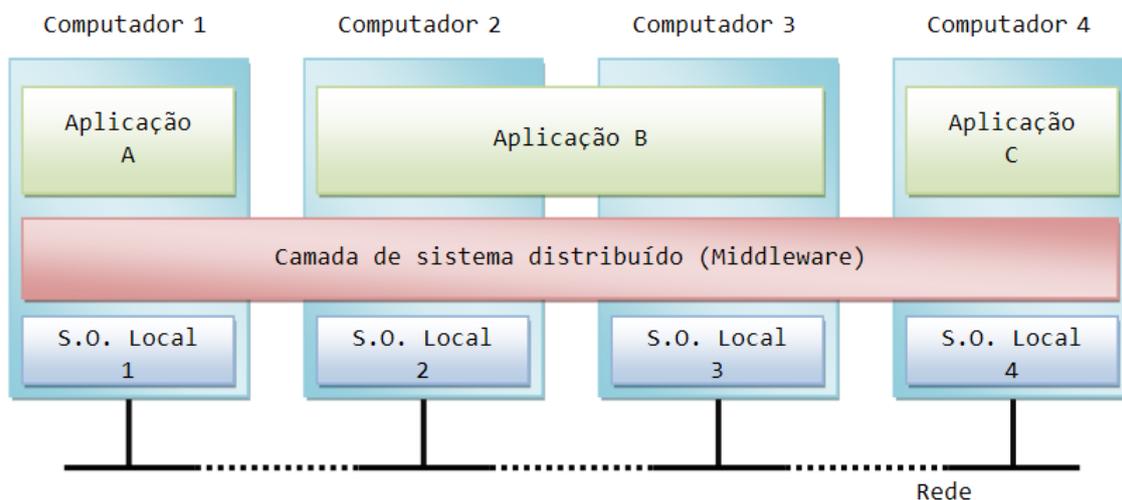


Figura 2.2: Um sistema distribuído típico. Adaptada de (TANENBAUM, 2006)

O conceito de *middleware* é muito utilizado no campo da Computação em Nuvem. Em grande parte, sua utilização se aplica diretamente à função exercida pela camada de *software* que exerce a coordenação e o gerenciamento das máquinas e aplicações que constituem um sistema distribuído. Os serviços de IaaS existentes atualmente funcionam através da utilização de interfaces de web services oferecidas por estes *middlewares*, através das quais os clientes alocam recursos contidos na infraestrutura do provedor. É correto afirmar, portanto, que a utilização do termo nuvem está diretamente relacionada à existência e à utilização de um grande sistema distribuído.

2.3.1.2 Grades computacionais

O termo grades computacionais, ou computação em grades, pode referir-se a diferentes tipos de tecnologia de acordo com o contexto em que está inserido. As áreas de *clusters* computacionais, Computação de Alto Desempenho (CAD), computação *peer-to-peer*, Computação Utilitária e, mais recentemente, Computação em Nuvem têm uma relação particular com o termo e o utilizam com conotações sutilmente divergentes.

No início da década de 1990, as chamadas *grids* nasceram no ambiente acadêmico, quando muitos computadores de alto desempenho foram conectados através de enlaces de comunicação de alta velocidade para servir a aplicações científicas com grande demanda por processamento de dados. Inspirados nas redes de energia elétrica e sua capacidade inerente de infiltração, facilidade de uso e confiabilidade, cientistas iniciaram a construção de um ambiente análogo que permitisse o acesso a recursos computacionais e aplicações em qualquer local e horário sem a necessidade de uma infraestrutura própria (BUYA *et al.*, 2005).

De acordo com (FOSTER, 1998), um ambiente de grades computacionais consiste em uma “infraestrutura de *hardware* e *software* capaz de prover acesso a recursos de computação de alta qualidade de forma segura, consistente, pervasiva² e barata”. Contudo, uma definição mais recente e amplamente aceita pela comunidade científica trata as grades computacionais como “o compartilhamento seguro, confiável e eficiente de recursos computacionais e dados entre instituições organizacionais independentes” (WEISHAUPL *et al.*, 2005).

Dentre os principais tipos de recursos que podem ser compartilhados através das grades computacionais, estão o processamento e armazenamento de dados, largura de banda e ferramentas de software. Há de se ressaltar que as *grids* se diferenciam da computação distribuída por focar em questões como alto desempenho, coordenação, gerenciamento e compartilhamento destes recursos (FOSTER *et al.*, 2001).

Um dos elementos fundamentais na implantação de um ambiente de grades computacionais é o *middleware* utilizado para tal. Trata-se de um *software* específico que provê toda a funcionalidade necessária para que os recursos de uma infraestrutura sejam compartilhados entre diferentes indivíduos e organizações.

²O termo “pervasivo” é um anglicismo referente à expressão inglesa “*pervasive*” e significa “algo dotado da capacidade de imbuir-se, infiltrar-se.”

Através do *middleware*, cria-se uma camada de virtualização e gerência de compartilhamento sobre a qual as aplicações dos usuários são executadas e administradas.

Um dos modos de implantação de ambientes em nuvem consiste nas chamadas nuvens comunitárias. Assim como as primeiras grades computacionais, os exemplos mais conhecidos deste tipo de nuvem são as chamadas *science clouds*, por terem como objetivo servir a aplicações científicas com grande demanda por recursos. Apesar desta estreita relação, todos os modos de implantação da Computação em Nuvem empregam tecnologias e termos inerentes ao campo das grades computacionais. O conceito de compartilhamento de recursos entre usuários e organizações em todo o mundo criado pelas *grids* é adaptado para um panorama onde um provedor compartilha sua infraestrutura entre muitos clientes, utilizando mecanismos de gerência e coordenação muito similares aos existentes nas grades. Ademais, o provimento destes recursos como bens utilitários, presente nas *grids* desde sua concepção e inspirado nas características das redes de energia elétrica, necessita de técnicas apuradas de medição. Alguns dos aspectos relativos este viés da utilização de infraestrutura computacional serão discutidos na Seção 2.3.1.3.

2.3.1.3 Computação Utilitária

Uma das características da Computação em Nuvem consiste na medição dos serviços oferecidos. Os recursos envolvidos devem ser monitorados e seu grau de utilização deve ser reportado tanto ao provedor quanto ao cliente de forma transparente. Nos casos onde há a necessidade de tarifação do modelo, esta característica torna-se vital para a viabilidade comercial do negócio. Assim como no fornecimento de bens utilitários como água, gás, telefonia e energia elétrica, o campo da Computação Utilitária³ busca desenvolver métodos para a provisão de recursos computacionais onde o cliente é cobrado com base na quantidade e na qualidade daquilo que utiliza.

Tendo em vista os tipos de serviços existentes, como entrega de capacidade de processamento, armazenamento de dados e redes de comunicação, e considerando-se que os mesmos devem atender automaticamente a flutuações de demanda, a Computação Utilitária deve atender a questões como o aumento de confiabilidade,

³O termo computação utilitária trata-se de uma tradução livre da expressão *Utility Computing*, muito encontrada na literatura.

melhoria de desempenho e redução do custo total na utilização de recursos (BUNKER, 2006).

A relação entre provedor e cliente atua como agente diretor do comportamento deste modelo de precificação. Em termos microeconômicos, ambas as partes possuem preferências, que representam exigências relacionadas aos resultados esperados de um serviço (BUYYA; BUDENDORFER, 2010). O equilíbrio é alcançado quando se encontra um ponto comum, onde vantagens são apresentadas a todos os envolvidos na negociação. Em geral, estes aspectos são estabelecidos em um Acordo de Nível de Serviço (SLA), que consiste em um contrato bilateral regulatório da interação entre as partes.

Paradigmas como grades computacionais e computação *peer-to-peer* propunham, desde sua concepção, a transformação dos recursos computacionais em um quinto bem utilitário (assim como o provimento de energia, água, gás e telefone). A Computação em Nuvem renova esta proposição através do fornecimento de infraestrutura, plataformas de desenvolvimento e software na forma de serviços baseados em um modelo *pay-as-you-go*⁴ (BUYYA, 2009), termo que sintetiza muitos dos aspectos fundamentais da Computação Utilitária.

2.3.2 Principais tecnologias

2.3.2.1 Virtualização

A prática de executar várias aplicações e servir a muitos usuários através de um único sistema não é recente. Pelo contrário, na década de 1970, quando os computadores eram grandes e extremamente caros, corporações como a IBM vendiam parte de seus recursos computacionais a um grande número de clientes, geralmente empresas de menor porte. À medida que os servidores se tornavam menores e financeiramente mais acessíveis, essas empresas passaram a possuir sua própria infraestrutura, e tornou-se comum a utilização de máquinas dedicadas a aplicações específicas. Porém, com este modelo, o cenário observado era comumente formado por servidores subutilizados, o que gerou o fenômeno conhecido como *server-sprawl*⁵. Nos últimos anos, os custos provenientes de espaço físico, energia, refrigeração e manutenção estão, mais uma vez, revertendo o panorama da utilização

⁴Pagamento de serviços de forma proporcional à taxa de utilização do usuário.

⁵O termo *sprawl* transmite a idéia de espalhamento, visto que a utilização dos recursos está "espalhada" entre os servidores.

de grandes datacenters, e a virtualização surge como peça fundamental para uma melhor utilização dos recursos computacionais disponíveis.

A virtualização consiste na técnica de prover a ilusão de acesso exclusivo a um recurso compartilhado. Com isto, é possível a execução de um grande número de *workloads* em um mesmo sistema de hardware de forma previsível, segura e isolada (SOLTER *et al.*, 2009). As diferentes modalidades existentes de virtualização oferecem benefícios diversos, como uma melhor utilização do sistema através do compartilhamento de recursos físicos, execução de aplicações em ambientes isolados, eliminação de conflitos de domínio (como utilização das mesmas portas TCP), utilização simultânea de vários sistemas operacionais em uma única máquina e isolamento de falhas. A seguir, são descritos os três principais tipos de virtualização:

- i. Virtualização para gerência de recursos.** Ao executar diferentes *workloads* em um único sistema, torna-se aparente a necessidade de gerenciar os recursos de *hardware* utilizados. Esta é a forma mais básica de virtualização, e pode ser exemplificada pelo gerenciamento realizado pelos sistemas operacionais ao executar simultaneamente um grande número de aplicações.
- ii. Virtualização em sistema operacional.** Nesta modalidade, um único sistema operacional gerencia várias instâncias que são executadas como sistemas independentes. Apesar de oferecer um ambiente seguro, isolado e de desempenho satisfatório para diferentes aplicações, todas as unidades virtualizadas dependem do bom funcionamento do sistema hospedeiro, o que representa uma importante limitação.
- iii. Virtualização completa.** Neste tipo de virtualização, sistemas operacionais diversos rodam simultaneamente na mesma máquina com um leve gerenciador de recursos, chamado hipervisor, sendo executado sobre o *hardware*. As instâncias executadas neste modelo são chamadas de máquinas virtuais.

Há dois tipos básicos de hipervisores. O tipo 1 é executado diretamente no *hardware*, enquanto o tipo 2 necessita de um sistema operacional hospedeiro. A virtualização através de hipervisores de tipo 2 difere da virtualização em sistema operacional, pois, na primeira, o sistema hospedeiro não tem ciência da máquina virtual que está sendo executada e gerenciada pelo hipervisor.

A Figura 2.3 apresenta uma comparação entre os diferentes tipos de virtualização, ressaltando a disposição das camadas de *software* envolvidas neste processo. A virtualização para gerência de recursos é mostrada no primeiro exemplo, onde o sistema operacional gerencia diretamente os recursos utilizados pelas aplicações. A virtualização em sistema operacional é ilustrada no segundo exemplo, com aplicações sendo executadas dentro de instâncias. Em seguida, os modos de funcionamento dos dois tipos de hipervisores são exibidos, dando-se ênfase ao fato de, no tipo 1, o hipervisor ser executado diretamente sobre o *hardware*, enquanto um sistema operacional hospedeiro é empregado para a execução do tipo 2.

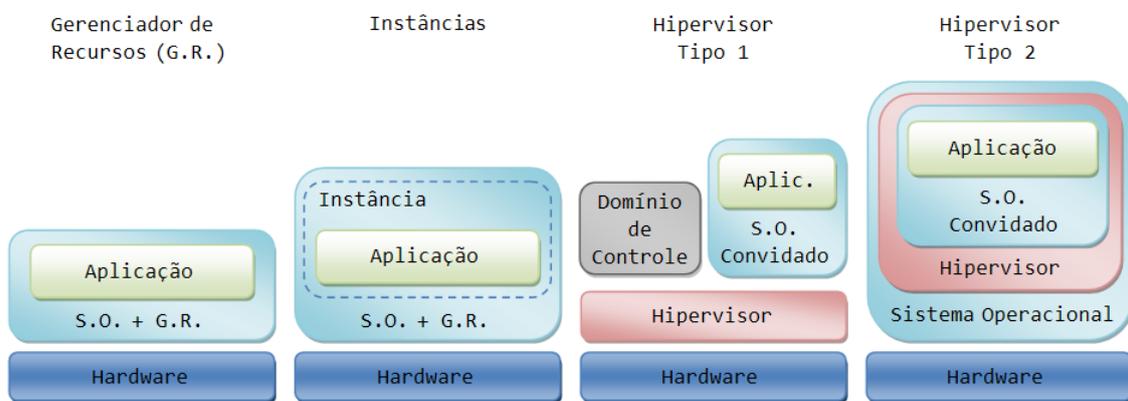


Figura 2.3: Tipos de virtualização. Adaptada de (SOLTER *et al.*, 2009)

A virtualização é a peça chave para a construção de um ambiente com tais funcionalidades de multi-arrendamento. O caráter “elástico” da nuvem, com a provisão de recursos virtualmente ilimitados, é obtido através da criação de novas instâncias de máquinas virtuais quando se faz necessário, oferecendo ao cliente *upgrades* de recursos instantâneos e automatizados.

2.3.2.2 *Web services*

O funcionamento da grande variedade de serviços disponíveis na internet, comumente designados como *web services*, segue uma série de protocolos que, em grande parte, empregam o padrão de linguagem *Extensible Markup Language* (XML) como elemento funcional básico. Adicionalmente, linguagens de descrição são utilizadas para a composição de interfaces através das quais clientes têm acesso aos serviços.

Os *web services* são oferecidos através da construção de sistemas que implementam Arquiteturas Orientadas a Serviços (SOA). O surgimento destas

arquiteturas está intrinsecamente relacionado a necessidades de mercado, quando um número crescente de empresas sentiu a necessidade de integração e interoperabilidade entre sistemas demasiadamente heterogêneos (ENDREI *et al.*, 2004).

A Computação em Nuvem torna possível a utilização de infraestrutura de hardware e software remotamente, oferecendo-os como serviços aos usuários finais. Isto é possível devido à utilização de interfaces de *web services*, através das quais requisições são traduzidas para o processamento por parte dos servidores de gerenciamento dos provedores, que administram o provimento de recursos de acordo com suas políticas internas de segurança e Acordos de Nível de Serviço (SLA) estabelecidos com seus clientes. Na Figura 2.4, este processo de requisição e resposta por serviços disponíveis na nuvem é representado de forma simplificada.

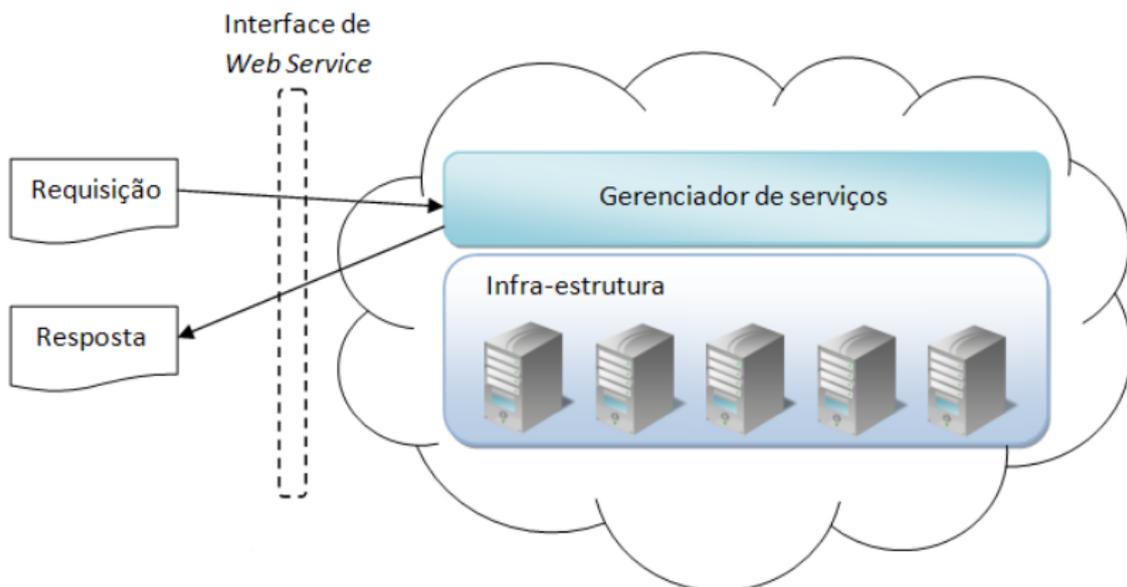


Figura 2.4: Comunicação entre cliente e nuvem através de uma interface de *web service*.

2.4 Modelos de serviço

O novo paradigma proposto pela Computação em Nuvem baseia-se na entrega de uma vasta variedade de recursos através de serviços disponíveis em escala mundial. Isto implica em um impacto profundo na forma como empresas e pessoas utilizam o *hardware* e o *software* de que necessitam, além de elevar ainda mais a importância das redes de comunicação através das quais esses recursos se tornam acessíveis.

A variedade de serviços disponíveis em uma nuvem faz com que sua classificação divirja sobremaneira. Nomenclaturas como Comunicação como Serviço (CaaS),

Gerenciamento como Serviço (MaaS), Dados como Serviço (DaaS), TI como Serviço (ITaaS) e até mesmo Tudo como Serviço (XaaS) são encontradas com relativa facilidade na literatura (PALLIS, 2010). Todavia, a estrutura da Computação em Nuvem é visivelmente formada por três camadas fundamentais (STANOEVSKA-SLABEVA *et al.*, 2010). Logicamente interconectados (Figura 2.5), os elementos que compõem esta visão triádica da nuvem são a Infraestrutura como Serviço (IaaS), a Plataforma como Serviço (PaaS) e o Software como Serviço (SaaS).

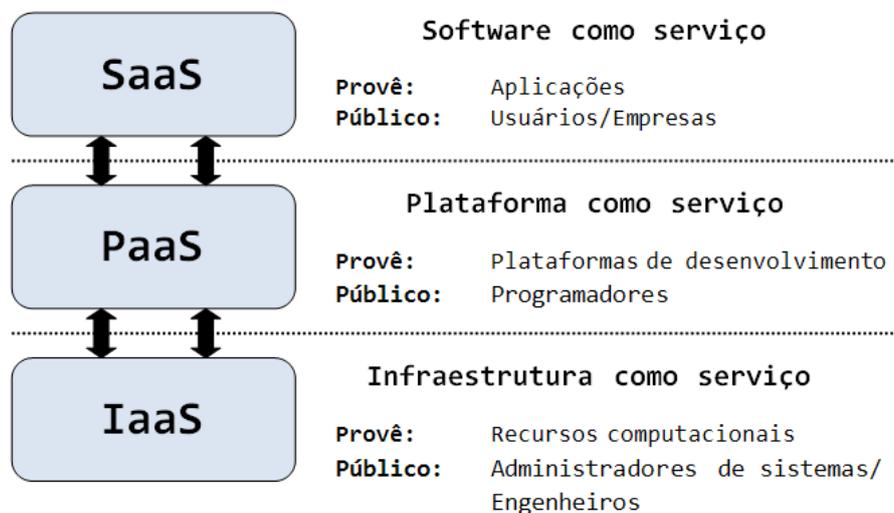


Figura 2.5: Visão em camadas dos modelos de serviço da Computação em Nuvem. Adaptada de (STANOEVSKA-SLABEVA *et al.*, 2010)

Atualmente, este tipo de classificação é amplamente aceito no mercado e na academia, pois engloba grande parte dos subtipos encontrados e define suas funcionalidades de forma coerente e concisa.

2.4.1 Infraestrutura como Serviço

Na era dos *mainframes*, grandes corporações vendiam tempo de processamento computacional a empresas que não possuíam capital suficiente para adquirir uma infraestrutura própria. O subsequente avanço dos sistemas de computação promoveu uma drástica mudança neste panorama, fazendo com que os computadores se fizessem presentes em organizações e lares de todo o mundo. A Computação em Nuvem surge para, mais uma vez, modificar a forma como empresas e pessoas adquirem e utilizam recursos computacionais.

Em contrapartida à venda direta de componentes de *hardware* e servidores completos, o modelo de Infraestrutura como Serviço (IaaS) oferece ao consumidor poder de processamento, armazenamento de dados, estrutura de rede e uma série de outros recursos básicos de forma virtualizada e seguindo os preceitos do *self-service* sob demanda, elasticidade e multi-arrendamento. Instituições que aderem a este tipo de serviço não têm controle direto sobre a infraestrutura física, mas podem utilizar os recursos virtualizados que lhes são oferecidos da forma que lhes convier, salvo restrições estabelecidas pelo provedor.

Ao eliminar a necessidade de posse de uma infraestrutura particular, o emprego de IaaS têm impacto direto em custos relacionados ao consumo de energia, espaço físico e manutenção de *hardware*. A taxação do serviço é realizada de acordo com o grau de utilização dos recursos, em um modelo frequentemente chamado de *pay-as-you-go*. Desta forma, o cliente paga apenas pelo que utiliza, o que põe fim a custos desnecessários provenientes da subutilização de servidores e *datacenters*.

Em conformidade com as cinco características principais da Computação em Nuvem, o modelo de IaaS fornece infraestrutura de forma dinâmica, automatizada e inerentemente escalável. Com isto, o cliente do serviço pode aumentar ou diminuir a quantidade de recursos alocados para si de acordo com sua necessidade. Esta flexibilidade permite, por exemplo, que pequenos empreendedores iniciem negócios com uma infraestrutura modesta e de baixo custo, alocando mais recursos de acordo com o crescimento de sua empresa.

Pioneira no mercado de IaaS e atualmente líder neste segmento, a empresa Amazon disponibiliza uma grande variedade de opções para a utilização de infraestrutura como serviço através do *Elastic Compute Cloud* (EC2). As configurações das instâncias oferecidas ao público pelo EC2 são mostradas na Tabela 2.1^{6,7}.

Ao abrir mão de uma infraestrutura própria através da adesão ao IaaS, uma organização vê muitos de seus dados ultrapassarem seus domínios, sendo armazenados na nuvem, um ambiente sobre o poder administrativo de terceiros. Por conseguinte, muitas questões relacionadas à segurança da informação são levantadas, sendo este um dos principais pontos de discussão do campo da Computação em

⁶Os valores de processamento são expressos em *EC2 Compute Units*. Um EC2 C.U. provê uma capacidade de processamento equivalente a um processador Opteron 2007 ou Xeon 2007 de 1,0-1,2 GHz.

⁷Elastic Block Store (EBS) é um serviço de armazenamento de dados oferecido pela Amazon.

Nuvem. Alguns dos principais aspectos relativos a este tema serão discutidos na Seção 2.6.

Tabela 2.1: Tipos de instâncias oferecidas pelo Amazon EC2 (AMAZON, 2010).

Tipo da instância	Memória	Processamento	Armazenamento	Arquitetura	Desempenho de E/S
Micro Instance	613 MB	Até 2 EC2 C.U.	Apenas EBS	32 ou 64 bits	Baixo
Small Instance	1,7 GB	1 EC2 C.U.	160 GB	32 bits	Moderado
Large Instance	7,5 GB	4 EC2 C.U.	850 GB	64 bits	Alto
Extra Large Instance	15 GB	8 EC2 C.U.	1.690 GB	64 bits	Alto
High-Memory Extra Large Instance	17,1 GB	6,5 EC2 C.U.	420 GB	64 bits	Moderado
High-Memory Double Extra Large Instance	34,2 GB	13 EC2 C.U.	850 GB	64 bits	Alto
High-Memory Quadruple Extra Large Instance	68,4 GB	26 EC2 C.U.	1.690 GB	64 bits	Alto
High-CPU Medium Instance	1,7 GB	5 EC2 C.U.	350 GB	32 bits	Moderado
High-CPU Extra Large Instance	7 GB	20 EC2 C.U.	1.690 GB	64 bits	Alto
Cluster Compute Quadruple Extra Large Instance	23 GB	33.5 EC2 C.U.	1.690 GB	64 bits	Muito Alto

2.4.2 Plataforma como Serviço

Enquanto a camada de IaaS oferece recursos computacionais e é voltada a administradores de sistemas e engenheiros que lidam diretamente com o gerenciamento de hardware e das redes de comunicação, a camada de Plataforma como Serviço (PaaS) é direcionada aos programadores, oferecendo-os um ambiente autoconfigurável através do qual é possível o desenvolvimento e o teste de aplicações sem a preocupação com a infraestrutura necessária para tal.

Por meio do modelo de PaaS, o programador pode criar e executar uma aplicação inteiramente na nuvem. As variações no uso da aplicação são então gerenciadas de forma integral pela plataforma. Caso seu nível de utilização cresça, um acréscimo na quantidade de recursos alocados é efetuado (RIGHTSCALE, 2010). Uma ação análoga é realizada nos casos de decréscimo do nível de utilização. Ademais, o ambiente oferecido pela Plataforma como Serviço pode cobrir todas as etapas de desenvolvimento de *software* ou especializar-se em uma área específica, como o gerenciamento de conteúdo (SUN, 2010).

Percebe-se, neste ponto, a comunicação vertical existente entre as camadas que compõem a estrutura da Computação em Nuvem. Enquanto utiliza interfaces de provedores para o consumo de recursos no modelo IaaS, a Plataforma como Serviço, por sua vez, oferece interfaces próprias para a comunicação com aplicações desenvolvidas pelos programadores que a utilizam. Trata-se, portanto, de uma camada de abstração entre a Infraestrutura como Serviço e o *Software* como Serviço (STANOEVSKA-SLABEVA *et al.*, 2010).

2.4.3 *Software* como Serviço

A forma tradicional de utilização de um programa se faz através da sua instalação e execução diretamente no computador do usuário. Empresas comercializam pacotes de *software* através da venda de licenças, que podem restringir seu uso por tempo de utilização e número máximo de instalações permitidas. Em um cenário corporativo, o uso de *software* neste modelo está atrelado à existência de uma infraestrutura de *hardware*, como servidores e *datacenters* próprios. Os custos gerados pela manutenção do ambiente de *software* e por essa infraestrutura muitas vezes ultrapassam a soma dos valores de todas as licenças adquiridas pela empresa. Apesar de nitidamente dispendiosa, esta é a forma como muitas instituições gerenciam a utilização de *software* até hoje.

O modelo de *Software* como Serviço (SaaS) consiste no provimento via rede de programas executados em sistemas e infraestruturas de terceiros, através de uma técnica de cobrança baseada na taxa de utilização do usuário. Comumente chamada de *pay-per-use*, essa técnica permite que clientes eliminem grande parte dos custos atrelados ao uso de um determinado *software*, excetuando-se os relativos à utilização das redes de comunicação. Não obstante, o emprego de SaaS gera uma grande dependência do usuário em relação a recursos que se encontram além de seu domínio administrativo, o que levanta muitos questionamentos a respeito da viabilidade de uma adesão integral ao modelo.

Embora seja a camada mais visível ao usuário final, o funcionamento do SaaS está intimamente ligado ao emprego de plataformas e infraestrutura como serviço. A comunicação vertical entre os elementos estruturais da Computação em Nuvem representada na Figura 2.5 é realizada em sua plenitude através do uso de SaaS, pois as aplicações oferecidas são executadas em ambientes baseados no modelo de PaaS, que, por sua vez, estão configurados sobre hardware virtualizado por meio de IaaS. A atuação conjunta destes elementos forma um sistema computacional complexo e inovador, cujo potencial ainda está sendo explorado tanto pelo mercado quanto pela academia.

O fato de toda a infraestrutura empregada estar inteiramente na nuvem cria a possibilidade de utilização dos mais diversos tipos de aplicação por meio de dispositivos com sérias limitações de recursos. Desta forma, dispositivos móveis podem ter acesso via *web services* a programas sofisticados e que necessitam de grande capacidade de processamento para sua execução. Observa-se, portanto, que a antiga topologia de rede onde os nós centrais consistiam em grandes *mainframes* que serviam a nós periféricos de recursos limitados é retomada, em contraste com a topologia atualmente em voga, onde os nós periféricos consistem em máquinas cada vez mais robustas.

O interesse atual pelo *Software* como Serviço torna-se evidente em ao se analisar dados divulgados por alguns dos mais importantes institutos de pesquisa econômica do mundo. Segundo o *International Data Corporation* (IDC), estima-se que o crescimento da receita provinda do SaaS em 2009 tenha sido próximo de 31%, o que representa mais de quatro vezes o crescimento de todo o mercado de software (IDC, 2008). Já de acordo com o instituto de pesquisa e consultoria Gartner, é esperado que os lucros gerados pelo SaaS dobrem até 2012, atingindo a marca de

14,5 bilhões de dólares (GARTNER, 2008).

2.5 Modos de implantação

A taxonomia utilizada para os modos de implantação de ambientes computacionais em nuvem varia consideravelmente, sendo muitos dos termos utilizados virtualmente equivalentes. Contudo, a classificação proposta pelo *National Institute of Standards and Technology* (NIST) desponta com grande aceitação e é encontrada com frequência na literatura (STANOEVSKA-SLABEVA *et al.*, 2010; REESE, 2009; SCHUBERT, 2010). De uma forma geral, os principais tipos de nuvem podem ser classificados como: privada, pública, híbrida e comunitária.

2.5.1 Nuvem pública

A infraestrutura e os serviços relacionados são disponibilizados ao público geral ou grandes conglomerados. Empresas como Amazon, Google, Microsoft e Salesforce oferecem todos os modelos de serviço (a saber: IaaS, PaaS e SaaS) através deste modo de implantação. Muitos expoentes da área têm as nuvens públicas como representantes puras do modelo da Computação em Nuvem (STANOEVSKA-SLABEVA *et al.*, 2010), visto que o mesmo se propõe a oferecer recursos como bens utilitários, o que necessariamente envolve uma relação entre provedor e cliente.

2.5.2 Nuvem privada

Neste modo de implantação, a infraestrutura e os serviços são dedicados exclusivamente a uma organização. Entretanto, os recursos físicos podem estar presentes em dependências de terceiros, e seu gerenciamento não é feito necessariamente *in loco*. Muitas empresas utilizam técnicas avançadas de virtualização para aumentar a eficiência em seus servidores. Tendo em vista que a taxa média de utilização situa-se em torno de 5 a 15% (IBM, 2008), e que, segundo (LOHR, 2009; MCKINSEY, 2009), este valor pode ser facilmente elevado à casa dos 18%, o emprego de uma nuvem privada mostra-se bastante atrativo, pois pode elevar esta taxa ao patamar de 35% (LOHR, 2009; MCKINSEY, 2009).

2.5.3 Nuvem comunitária

As nuvens comunitárias formam ambientes compartilhados entre várias organizações com interesses em comum. Em geral, atendem a nichos específicos, como, por exemplo, à comunidade acadêmica através do suporte a projetos científicos. Assim como nas nuvens privadas, os recursos físicos envolvidos podem estar localizados em dependências de terceiros e seu gerenciamento não é necessariamente realizado por uma das organizações que os utilizam.

2.5.4 Nuvem híbrida

As nuvens híbridas são uma combinação de duas ou mais nuvens distintas. Apesar de manterem-se como entidades únicas, as nuvens se comunicam através de interfaces padronizadas, criando-se a possibilidade de compartilhamento de recursos e aplicações. Muitas organizações empregam este modo de implantação para tirar proveito dos benefícios trazidos pelas nuvens públicas, enquanto mantêm algumas aplicações e dados em seu domínio administrativo através de nuvens privadas, evitando, desta forma, problemas relacionados à segurança de suas informações (SUN, 2010).

2.6 Segurança

Ao abrir mão de uma infraestrutura própria e aderir a serviços como IaaS, PaaS ou SaaS, uma organização vê muitos de seus dados ultrapassarem seus domínios administrativos e serem alocados e gerenciados em servidores de terceiros. Todos esses serviços são oferecidos por provedores através de técnicas de multi-arrendamento de recursos, o que significa que dados e aplicações de várias organizações distintas podem fazer uso de um mesmo servidor simultaneamente (SUN, 2010). Por conseguinte, muitas questões relativas à segurança são levantadas e exercem forte influência na decisão de clientes pela utilização desses serviços. A tabela 2.2 expõe os principais obstáculos para o crescimento e adoção da Computação em Nuvem.

Um dos aspectos fundamentais na terceirização de *datacenters* é a perda de controle da localização física dos dados. Embora acessíveis, não é possível saber com exatidão em que servidor as informações estão armazenadas. Alguns dos principais problemas práticos advindos desta conjuntura são uma eventual falência do provedor

Tabela 2.2: Obstáculos para o crescimento e a adoção da Computação em Nuvem (STANOEVSKA-SLABEVA *et al.*, 2010)

Obstáculo	Fonte
Disponibilidade	(IDC, 2008; ARMBRUST <i>et al.</i> , 2009)
Segurança	(IDC, 2008)
Desempenho	(IDC, 2008; ARMBRUST <i>et al.</i> , 2009)
Retenção de dados	(ARMBRUST <i>et al.</i> , 2009)
Confidencialidade e auditoria de dados	(ARMBRUST <i>et al.</i> , 2009)
Gargalos em transferências de dados	(ARMBRUST <i>et al.</i> , 2009)
Integração com sistemas internos	(IDC, 2008)
Falta de personalização	(IDC, 2008)

de serviços ou uma falha geral em seus sistemas ou infraestrutura. Há, ainda, a possibilidade de uma terceira parte obter acesso aos dados do provedor por meio de ações judiciais (REESE, 2009). Entretanto, soluções relativamente simples como criptografia e *backups* em servidores alternativos podem dirimir muitos dos riscos associados ao gerenciamento de dados na nuvem.

A infraestrutura oferecida por um provedor de IaaS é virtualizada. Os clientes, portanto, não possuem servidores físicos de uso exclusivo, mas instâncias de máquinas virtuais. Ademais, um mesmo cliente pode dispor de múltiplas instâncias, que, por sua vez, devem comunicar-se por meio de redes privadas. É de vital importância que o provedor assegure aos clientes que o tráfego dessas redes esteja protegido de qualquer forma de acesso indevido. Entretanto, esta é uma responsabilidade compartilhada por ambas as partes, e muitas das técnicas clássicas de prevenção a ataques podem ser aplicadas às instâncias de máquinas virtuais tal qual são utilizadas em servidores tradicionais.

Por fim, deve-se ter em mente que falhas de segurança são igualmente comuns em infraestruturas virtualizadas e físicas. Há de se ressaltar, porém, a capacidade de recuperação inerente aos ambientes em nuvem. Em servidores tradicionais, o processo de reativação de um sistema consome tempo e envolve paralisações de componentes críticos, acarretando prejuízos a quem os utiliza. As máquinas virtuais,

por outro lado, podem ser reativadas rápida e automaticamente, decrementando drasticamente o tempo médio para recuperação (MTTR). Além disso, falhas de segurança podem ser corrigidas e aplicadas sem qualquer influência na experiência do usuário final. Percebe-se, portanto, que a Computação em Nuvem oferece algumas vantagens quanto à segurança dos sistemas, embora seus pontos negativos sejam comumente evidenciados.

Bucando explorar os principais aspectos relativos à segurança em ambientes computacionais em nuvem, organizações como a *Cloud Security Alliance* realizam pesquisas na área e propõem um conjunto de boas práticas com o objetivo de dirimir muitos dos problemas enfrentados por profissionais e pesquisadores.

2.7 *Middlewares e framework* de simulação

As atividades de pesquisa no campo da Computação em Nuvem são realizadas, em grande parte, por meio da análise do comportamento de ambientes reais, denominados *testbeds*. Para tal, são empregadas várias modalidades de *middlewares*, que possibilitam a construção de nuvens em todos os seus modos de implantação. Ademais, a existência de *middlewares* de código aberto representa um ponto chave para os trabalhos de pesquisa, pois possibilita uma análise profunda e detalhada do funcionamento desses ambientes.

Por outro lado, os trabalhos que tratam da simulação dos ambientes computacionais em nuvem encontram-se em estado incipiente. Atualmente, a ferramenta de maior destaque nesta área consiste no *framework* de simulação CloudSim, que, embora ofereça um grande número de funcionalidades, ainda apresenta uma série de limitações.

2.7.1 CloudSim

O *framework* CloudSim, desenvolvido pelo *Cloud Computing and Distributed Systems Laboratory* (CLOUDS) da Universidade de Melbourne, visa oferecer os recursos necessários para a simulação de ambientes computacionais em nuvem. Desenvolvido inteiramente em Java e licenciado pela *General Public License* (GPL), o *framework* é extensível, facilmente adaptável e permite a criação de simulações em grande escala com alto grau de customização (CALHEIROS *et al.*, 2009).

A Figura 2.6 ilustra as camadas que compõem a arquitetura do CloudSim. No nível mais inferior, observa-se o motor de simulação, responsável pelas operações

de criação, gerenciamento e exclusão das entidades simuladas. A camada seguinte ilustra as principais classes que compõem o *framework*. No módulo de rede, são realizados o mapeamento de enlaces entre *datacenters* e clientes e o cálculo de atraso das mensagens trocadas entre os mesmos. O módulo de recursos da nuvem realiza a manipulação e coordenação dos eventos da simulação, além de gerenciar os dados relativos à infraestrutura oferecida por meio dos *datacenters* simulados. Em seguida, o módulo de serviços da nuvem ilustra as ações de provimento de máquinas virtuais e alocação de recursos como memória de sistema, processamento, armazenamento de dados e largura de banda. Como sua função consiste no gerenciamento desses recursos, ressalta-se sua grande integração com o módulo anterior. Observa-se então o módulo de serviços das máquinas virtuais, onde são realizadas a gerência das mesmas e a execução das tarefas enviadas pelos clientes, denominadas *cloudlets*. Por fim, a comunicação das entidades que compõem a nuvem com os clientes que utilizam seus recursos é feita por meio do módulo de interface, no qual máquinas virtuais e *cloudlets* podem ser manipuladas.

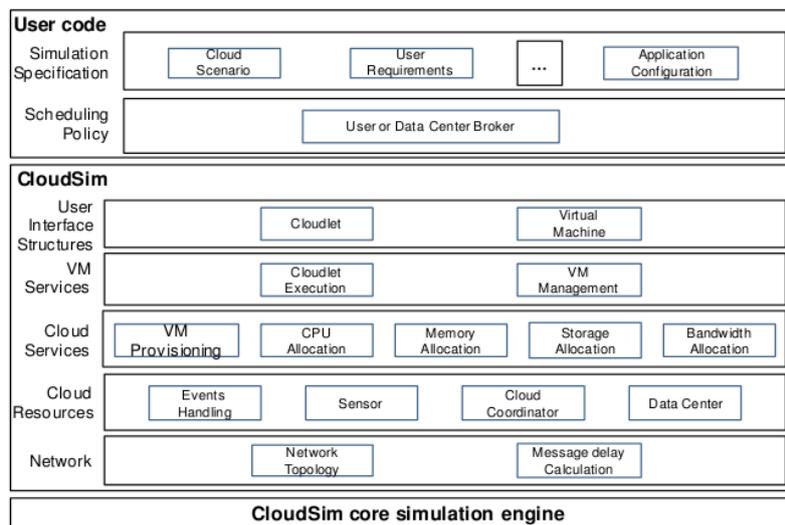


Figura 2.6: Arquitetura em camadas do CloudSim. (CALHEIROS *et al.*, 2010b)

A última camada da Figura 2.6 representa o código que o usuário do *framework* deve implementar para a criação dos ambientes de simulação. O módulo de política de escalonamento indica a criação das políticas de decisão e escalonadores que nortearão os processos de simulação. Além disso, o ambiente a ser simulado pode ser detalhadamente modelado, especificando-se os recursos que compõem a nuvem e os perfis de utilização referentes a cada um dos clientes.

Além dos mecanismos de decisão denominados políticas de *broker* (Seção 3.6),

o CloudSim permite a implementação de políticas de alocação de máquinas virtuais entre os *hosts* de um mesmo *datacenter*, escalonadores de máquinas virtuais em *hosts* e escalonadores de *cloudlets* em máquinas virtuais. A Figura 2.7 ilustra o efeito que a escolha desses escalonadores exerce na execução das *cloudlets*. Em (a), máquinas virtuais e *cloudlets* são escalonadas com compartilhamento de recursos no domínio do espaço. Desta forma, as máquinas virtuais são alocadas em momentos distintos, e as *cloudlets* executadas com dedicação exclusiva por cada um dos núcleos de processamento. Em (b), altera-se o escalonamento de *cloudlets* para o compartilhamento de recursos no domínio do tempo. Isto faz com que as mesmas sejam executadas de forma concorrente em um dado núcleo de processamento. Em (c), observa-se o inverso. As máquinas virtuais são escalonadas com compartilhamento de recursos no domínio do tempo, e as *cloudlets* no domínio do espaço. Assim, enquanto as primeiras são executadas concomitantemente, as últimas são executadas de forma exclusiva por um dos núcleos de processamento. Já em (d), os recursos são compartilhados no domínio do tempo por máquinas virtuais e *cloudlets*, o que gera um processamento concorrente de todas as entidades.

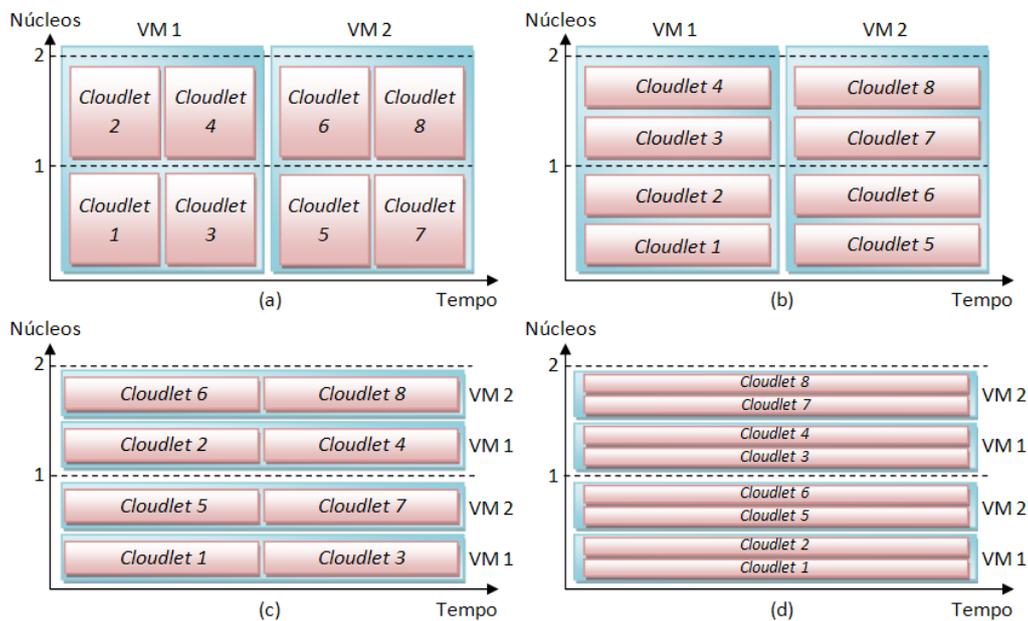


Figura 2.7: Escalonamento de máquinas virtuais e *cloudlets* de acordo com o tipo de compartilhamento de recursos: (a) Domínio do espaço para máquinas virtuais e *cloudlets*, (b) Domínio do espaço para máquinas virtuais e domínio do tempo para *cloudlets*, (c) Domínio do tempo para máquinas virtuais e domínio do espaço para *cloudlets* e (d) Domínio do tempo para máquinas virtuais e *cloudlets*. Adaptada de (CALHEIROS *et al.*, 2009)

A Figura 2.8 ilustra o fluxo de comunicação entre as principais entidades de simulação do CloudSim. Inicialmente, todos os *datacenters* criados são registrados no *Cloud Information Service* (CIS), entidade responsável pelo provimento de informações relativas ao ambiente simulado. Posteriormente, as requisições por recursos dos clientes, realizadas por meio de seus *brokers*, são respondidas com uma lista dos *datacenters* disponíveis. As características de cada um dos *datacenters* são então analisadas para se verificar a viabilidade de alocação das máquinas virtuais do cliente, que, após esta análise, são enviadas para criação. Em seguida, o cliente envia tarefas para execução, recebendo os resultados como resposta. A comunicação se encerra com a desalocação das máquinas virtuais.

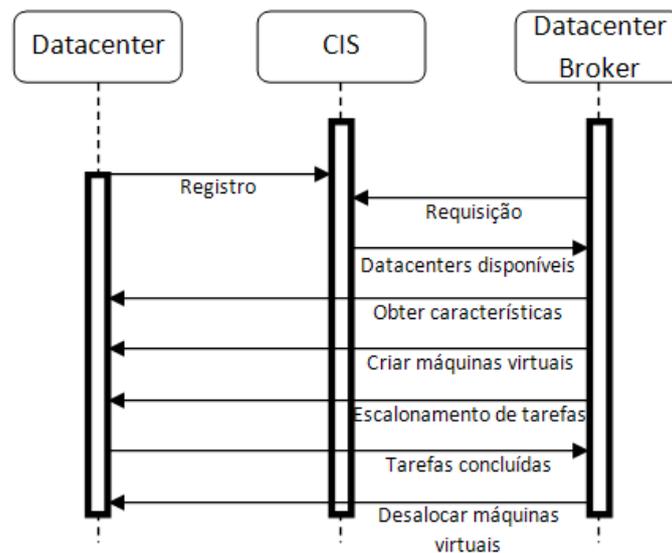


Figura 2.8: Fluxo de comunicação entre entidades do CloudSim. Adaptada de (CALHEIROS *et al.*, 2010b)

2.8 Perspectivas futuras

O campo da Computação em Nuvem, embora recente, já apresenta um grande volume de trabalhos acadêmicos relacionados. Trata-se de uma área dinâmica e em contínua evolução. Embora ainda se encontre muitos pontos de divergência quanto a conceitos e taxonomia, sua crescente popularidade tende a dirimi-los, criando-se gradualmente um consenso entre mercado e academia.

Atualmente, questões como segurança e padronização representam pontos-chave para o crescimento e amadurecimento da área. A Amazon, tanto pelo pioneirismo quanto pela grande participação no setor de IaaS, desponta como detentora das

interfaces mais utilizadas para o provimento de infraestrutura como serviço, o que pode significar a criação de um padrão em torno das mesmas. De acordo com (IDC, 2008; GARTNER, 2008), há ainda a previsão de grande crescimento nos próximos anos para os setores de PaaS e SaaS, pois muitas organizações procurarão redução de custos e aumento de produtividade através dos serviços oferecidos pela Computação em Nuvem.

2.9 Resumo

Este capítulo apresentou os fundamentos teóricos básicos para a compreensão dos principais aspectos relacionados ao campo da Computação em Nuvem. Após uma breve definição do termo, foi realizada uma contextualização histórica, onde a evolução dos sistemas de computação foi evidenciada. Em seguida, os modelos computacionais relacionados e as principais tecnologias envolvidas foram discutidos. Fez-se então uma descrição dos três principais modelos de serviço e dos modos de implantação de nuvens existentes. O capítulo encerrou-se com uma discussão sobre aspectos de segurança, ferramentas de pesquisa existentes e perspectivas futuras para área.

Capítulo 3

CloudReports

Neste capítulo, são apresentadas as ferramentas empregadas para a criação da aplicação que constitui o objetivo principal deste trabalho. A metodologia adotada é explorada através de análises descritivas e estruturais do simulador, além de uma descrição dos ambientes a serem simulados. A seção final discute os algoritmos construídos para os mecanismos de decisão utilizados nas simulações.

3.1 Material

A execução deste trabalho foi realizada unicamente através da utilização de ferramentas de *software* de código aberto. Esta característica dá ao programador total liberdade para a modificação e adaptação de seus componentes, caso os mesmos não atendam integralmente as necessidades do projeto. Entre as licenças utilizadas, estão a *Oracle Corporation Binary Code License* (OCBCL) para a linguagem de programação Java, a *Common Development and Distribution License* (CDDL) para o ambiente de desenvolvimento NetBeans e a *General Public License* (GPL) para os *frameworks* CloudSim e JFreeChart.

3.1.1 Ferramentas de programação

A codificação da aplicação criada por meio deste trabalho foi realizada através do emprego da plataforma de desenvolvimento Java. Esta escolha proporcionou à aplicação vantagens como a independência de plataforma, o que possibilita sua execução em diferentes sistemas operacionais sem a necessidade de recompilação. A linguagem Java possui ainda um sistema de gerenciamento de memória automático

comumente denominado *garbage collector*, que isenta o programador de atividades complexas, diminuindo assim a possibilidade de falhas no sistema advindas de erros de programação (GONG *et al.*, 1999).

Adicionalmente, fez-se uso do NetBeans, um Ambiente de Desenvolvimento Integrado (IDE) desenvolvido e mantido pela Sun Microsystems em conjunto com uma comunidade de desenvolvedores através de projetos de código aberto. A IDE possui compatibilidade com um grande número de linguagens, como C, C++, FORTRAN, Ada e COBOL, mas seu principal foco sempre foi a integração com a plataforma Java. A escolha para sua utilização na realização deste trabalho deveu-se em grande parte à sua excelente ferramenta para construção de interfaces gráficas através da API Swing, o que proporciona um aumento significativo de produtividade ao programador. Ademais, são oferecidos recursos comuns como coloração de sintaxe, inspeção de componentes, navegação de objetos, *debugger* e adição de *plugins*.

3.1.2 CloudSim

A aplicação desenvolvida visa a realização de simulações de ambientes de computação distribuída onde a virtualização desempenha um papel determinante. Estes são os ambientes característicos da Computação em Nuvem, onde máquinas virtuais são executadas em grandes *datacenters* e atendem a usuários em escala global.

Para a realização das simulações desses ambientes, foram utilizados os recursos oferecidos pelo *framework* CloudSim. Esta ferramenta representa o motor de simulação da aplicação, sendo responsável pela criação e gerenciamento de todos os elementos do ambiente simulado. Os principais aspectos relativos ao funcionamento do *framework* foram discutidos na Seção 2.7.1.

3.1.3 JFreeChart

O JFreeChart trata-se de uma API Java de código aberto para a construção de gráficos de alta qualidade de forma simples e facilmente integrável em diversos tipos de programas. A aplicação aqui descrita tem como objetivo, além da realização de simulações, a geração de relatórios detalhados sobre as mesmas. Esses relatórios são compostos em parte por gráficos ilustrativos que descrevem muitos aspectos do comportamento das entidades simuladas, facilitando sobremaneira a compreensão

dos dados a serem analisados. O JFreeChart disponibiliza um grande número de opções de gráficos e integra a eles automaticamente recursos de *zoom* e exportação para arquivos de imagem. Sua escolha deveu-se especialmente pela riqueza de funcionalidades que oferece e por sua implementação integral em linguagem Java.

3.2 Análise descritiva

O objetivo principal deste trabalho concentra-se na construção de uma extensão gráfica para o *framework* CloudSim, visando a concepção de uma ferramenta de simulação para sistemas distribuídos cujas arquiteturas seguem o modelo hoje denominado Computação em Nuvem. Embora sua aplicação possua ampla disseminação comercial e o estudo de suas tecnologias gere um interesse crescente do meio acadêmico, esse novo paradigma de provisão e utilização de recursos computacionais ainda carece de alternativas para a criação de ambientes em âmbito de simulação, e as atividades de pesquisa e desenvolvimento da área ainda se restringem à utilização de *testbeds* reais, o que naturalmente limita o poder de escala dos experimentos e os tornam de difícil reprodução (CLOUDSIM, 2010).

Empregando-se a plataforma de desenvolvimento Java e o *framework* de simulação CloudSim, além das ferramentas adicionais, criou-se o CloudReports, um simulador gráfico que busca tornar a criação de simulações de ambientes computacionais em nuvem uma atividade prática e objetiva, embora mantendo características fundamentais como escalabilidade, alto grau de customização e extensibilidade.

O CloudReports permite que usuários não familiarizados com a linguagem Java realizem simulações através de uma interface gráfica simples e intuitiva (Figura 3.1). Os ambientes construídos podem ser salvos e armazenados em disco para a reprodução posterior dos resultados obtidos, e detalhes das simulações são registrados através de um recurso de geração automática de *logs*.

A interface gráfica oferece flexibilidade para a criação de *datacenters* com quantidade variável de recursos, incluindo, ainda, a customização individual dos hosts que o compõem. Esse aspecto flexível também é aplicado à criação de clientes, dado que os mesmos podem dispor de máquinas virtuais com configurações distintas e inteiramente customizáveis.

A produção de relatórios consiste em uma funcionalidade chave da aplicação.

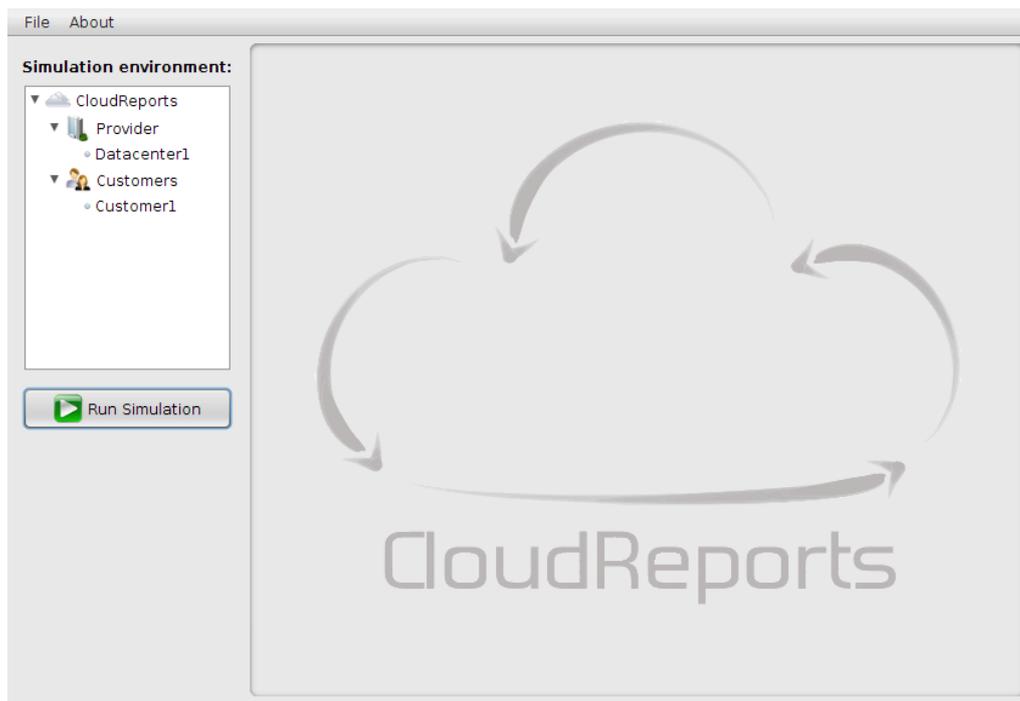


Figura 3.1: Tela inicial do CloudReports.

Após o término da simulação, é realizada uma compilação de informações importantes para a compreensão dos dados gerados. Grande parte dessas informações é exibida na forma de gráficos explanatórios, que podem ser manipulados pelo usuário e armazenados em disco no formato de imagens.

3.3 Ambiente de simulação

O ambiente de simulação do CloudReports reproduz a interação entre um provedor de Infraestrutura como Serviço (IaaS) e seus clientes. O provedor possui de um a muitos *datacenters*, que podem ser modelados de acordo com seu sistema operacional, arquitetura dos processadores, hipervisor empregado, banda disponível, custos de utilização, política de alocação de máquinas virtuais e opções de consumo de energia. Há ainda atributos relativos às redes de comunicação que interligam as entidades simuladas. Os *hosts* que compõem um *datacenter* podem ser configurados atribuindo-se valores de memória de sistema, banda, armazenamento, poder de processamento e opções de escalonadores de máquinas virtuais e modelos de utilização energética. A criação de *datacenters* heterogêneos torna-se simples por meio da adição de diferentes perfis de configuração de *hosts*.

Os clientes, por sua vez, são modelados por meio de um perfil de utilização de

recursos e configurações relativas às máquinas virtuais que serão executadas através da infraestrutura oferecida pelo provedor. O perfil de utilização consiste em uma descrição das tarefas, denominadas *cloudlets*, que serão processadas pelas máquinas virtuais e pela indicação de uma política de escolha de *datacenters* representada pela entidade *broker*. As *cloudlets* são descritas através de características como quantidade de núcleos necessária para o processamento, comprimento em milhões de instruções por segundo (MIPS) e tamanho dos arquivos de entrada e saída que serão transferidos entre clientes e *datacenters*. Há ainda modelos distintos de utilização de CPU, memória e banda de comunicação. Por fim, o tipo de *broker* determina de que forma serão escolhidos os *datacenters* que alocarão as máquinas virtuais e, por conseguinte, executarão as *cloudlets*. O CloudReports oferece quatro tipos distintos de brokers, que selecionam *datacenters* de acordo com critérios de custo, poder de processamento e latência de rede. A quarta opção consiste na ausência desses critérios, empregando-se um algoritmo *round robin* para a seleção dos *datacenters*.

Após a criação de um ambiente seguindo os modelos supracitados, o usuário dá início à simulação através do botão “*Run Simulation*”. Primeiramente, o módulo de simulação do CloudReports obtém todas as informações criadas pelo usuário através da interface gráfica e as traduz para a criação das entidades de simulação do CloudSim. Em seguida, todos os clientes buscam alocar suas máquinas virtuais utilizando os recursos disponíveis nos *datacenters* do provedor e seguindo as políticas de seus respectivos *brokers*. A fase de alocação pode perdurar durante todo o período de simulação através da prática de migração de máquinas virtuais. Os clientes então enviam suas *cloudlets* para execução e aguardam o retorno das mesmas. Por fim, os dados gerados durante todo o processo são compilados pelo módulo de gerenciamento de relatórios do CloudReports e exibidos para o usuário. O modelo típico de um ambiente simulado pela aplicação é ilustrado na Figura 3.2.

3.4 Arquitetura

O CloudReports é composto por cinco módulos distintos. Seu funcionamento advém da comunicação entre os elementos que o constituem e os recursos oferecidos pelas ferramentas citadas na Seção 3.1, formando a estrutura observada na Figura 3.3. Os módulos mostrados em azul foram criados por meio deste trabalho e são partes integrantes do CloudReports, enquanto os mostrados em vermelho representam componentes de terceiros utilizados pela aplicação.

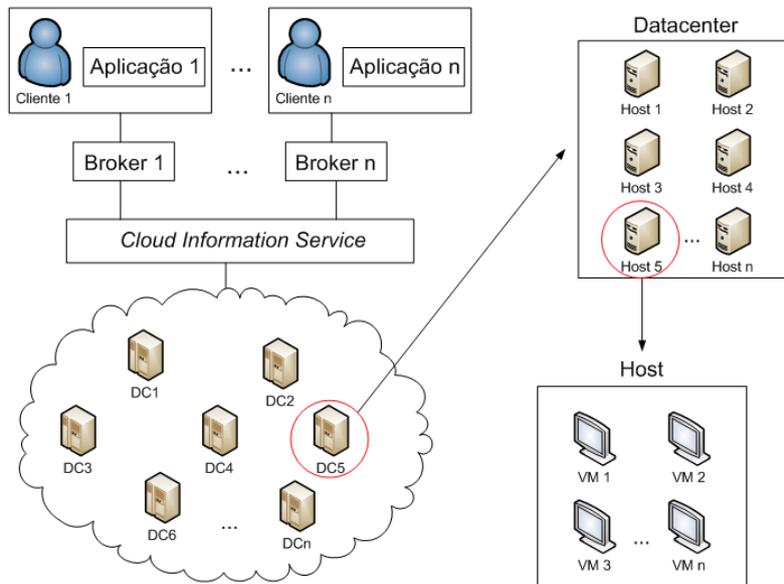


Figura 3.2: Ambiente de simulação típico do CloudReports.

Na base dessa arquitetura, observa-se a Máquina Virtual Java (JVM), responsável pela interpretação dos *bytecodes* e pelo gerenciamento dos recursos utilizados junto ao sistema operacional. O CloudSim, *framework* utilizado como motor de simulação, encontra-se entre a JVM e os módulos do CloudReports responsáveis pela tradução do ambiente criado através da interface gráfica para geração das entidades a serem simuladas.

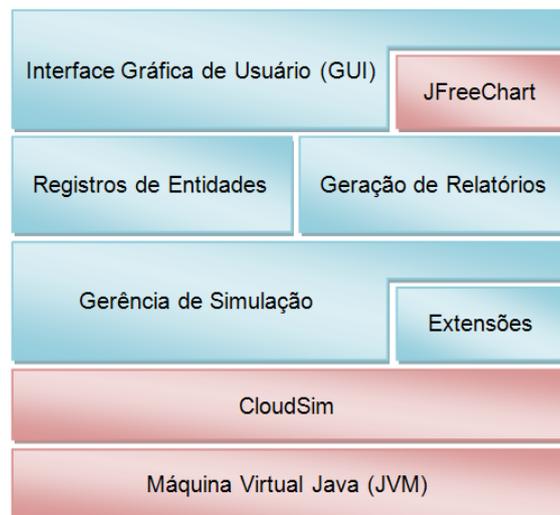


Figura 3.3: Estrutura em módulos do CloudReports.

3.4.1 Extensões

O módulo de extensões é composto por classes que herdam características diretamente de elementos do CloudSim. Os *brokers* constituem o principal exemplo, pois são classes filhas do componente *DatacenterBroker* e buscam tornar seu comportamento mais específico, atendendo diretamente às necessidades da aplicação.

Para que as políticas de decisão discutidas na Seção 3.6 fossem codificadas, criou-se quatro implementações da classe abstrata *Broker*, a saber: *CheapestDatacenterBroker*, *FastestDatacenterBroker*, *LowestLatencyDatacenterBroker* e *RoundRobinDatacenterBroker*. Esta relação de dependência é ilustrada do diagrama da Figura 3.4. A classe *Broker*, por sua vez, é uma extensão da classe *DatacenterBroker*, pertencente ao *framework* CloudSim e responsável pela escolha de *datacenters* para a alocação de máquinas virtuais.

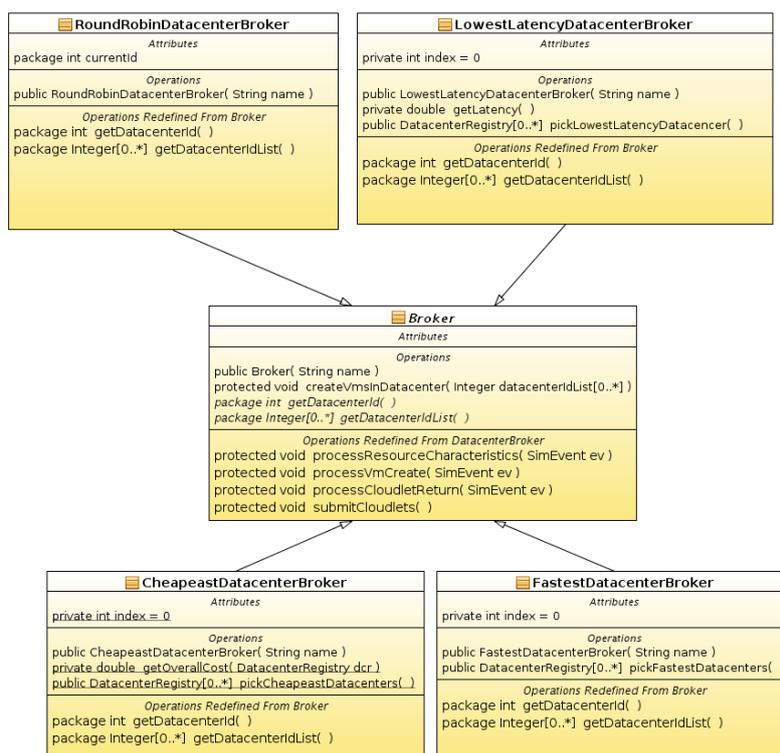


Figura 3.4: Diagrama UML das classes que implementam políticas de *broker*.

Também foram implementadas extensões para as classes *PowerDatacenter* e *DatacenterCharacteristics*. Nestes casos, os objetivos consistiam em melhorar as formas como *cloudlets* eram executadas nos *datacenters* e facilitar a extração de

dados para a composição dos relatórios finais.

3.4.2 Gerência de simulação

O módulo de gerência de simulação é responsável pela obtenção de dados junto aos registros de entidades e pela tradução e transmissão dos mesmos à camada inferior. Após o termino da simulação, este módulo também é encarregado de enviar os dados resultantes para a camada de geração de relatórios, que, por sua vez, realiza o tratamento de todas as informações e as apresenta à interface gráfica. Grande parte desse tratamento é realizado com a utilização de recursos oferecidos pelo *framework* JFreeChart para a geração automática de gráficos.

3.4.3 Registro de entidades

No módulo de registro de entidades, são armazenadas informações sobre a modelagem do cenário de simulação que é realizada através da interface gráfica. Posteriormente, essas informações são traduzidas pelo módulo de gerência de simulação para a criação de entidades do CloudSim. As principais classes pertencentes a esse módulo são a *DatacenterRegistry* e a *CustomerRegistry*, que armazenam informações de *datacenters* e clientes respectivamente.

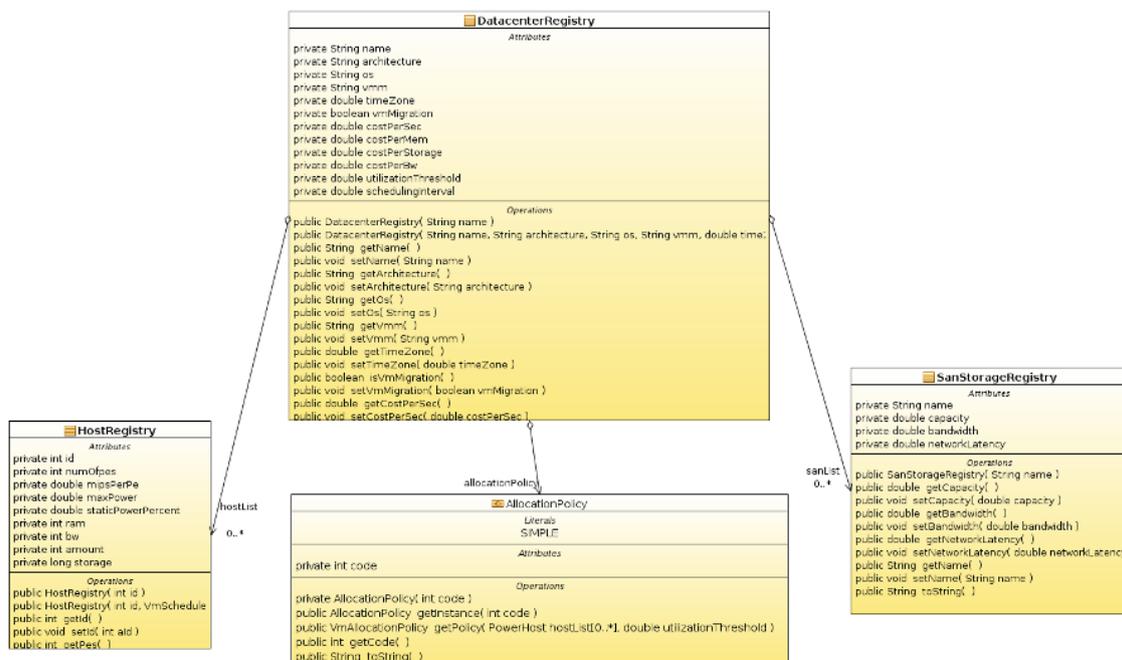


Figura 3.5: Diagrama UML das classes que compõem um registro de *datacenter*.

O diagrama mostrado na Figura 3.5 ilustra as relações de dependência entre as classes *DatacenterRegistry*, *HostRegistry*, *SanStorageRegistry* e a enumeração *AllocationPolicy*. Observa-se que um registro de *datacenter* possui uma lista de registros de hosts e áreas de armazenamento em rede, além de uma política de alocação de máquinas virtuais.

No diagrama da Figura 3.6, são representadas as relações de dependência entre as classes *CustomerRegistry*, *UtilizationProfile* e *VirtualMachineRegistry*. O diagrama ilustra claramente que os clientes simulados possuem um número arbitrário de máquinas virtuais e um perfil de utilização de recursos, que armazena informações sobre as *cloudlets* e está vinculado a uma política de broker.

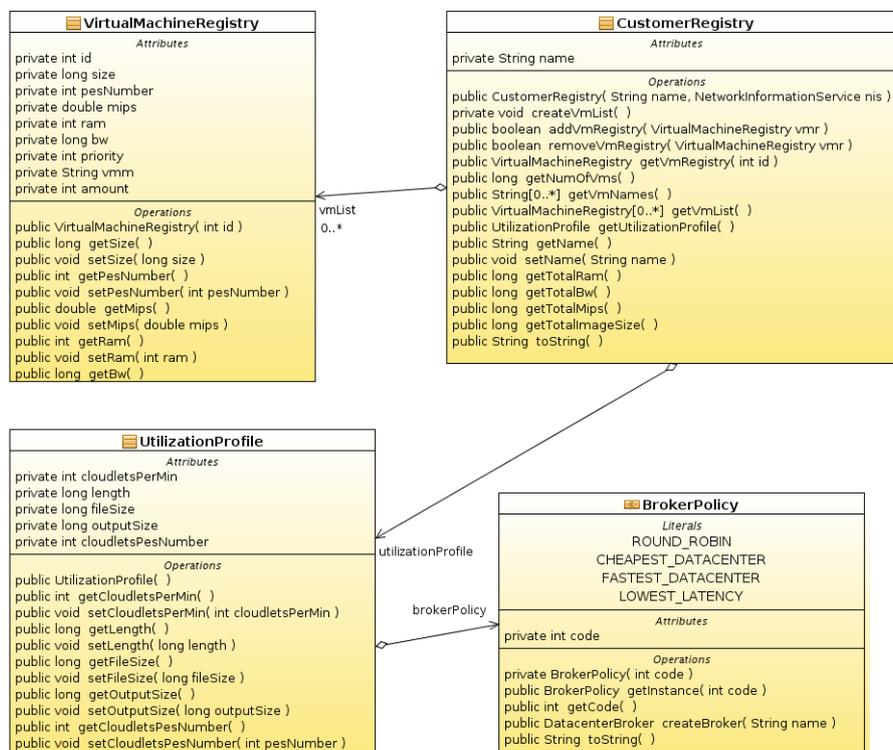


Figura 3.6: Diagrama UML das classes que compõem um registro de cliente.

3.4.4 Geração de relatórios

O módulo de geração de relatórios comunica-se diretamente com a gerência de simulação para a obtenção dos dados que compõem os relatórios finais. As informações são obtidas tanto em tempo de simulação quanto após o término da mesma. Em seguida, os resultados são processados e transmitidos à camada superior, onde serão exibidos por meio da interface gráfica.

3.4.5 Interface Gráfica de Usuário

A camada superior da estrutura consiste na interface gráfica (GUI). O usuário a utiliza diretamente para criar ambientes, dar início às simulações e visualizar os relatórios finais. Por ser responsável pelo armazenamento das características que definem os ambientes, o módulo de registros de entidades comunica-se constantemente com a camada superior e também transmite informações de forma direta para a geração de relatórios.

3.5 Entidades funcionais

As simulações realizadas por meio do CloudReports possuem dois tipos de entidades funcionais: os registros e os serviços de informação. Os primeiros representam elementos que compõem um ambiente real, como *datacenters* e clientes. São eles: *CustomerRegistry*, *DatacenterRegistry*, *HostRegistry*, *SanStorageRegistry* e *VirtualMachineRegistry*. Instâncias dessas entidades são geradas automaticamente à medida que o usuário modela um ambiente de simulação através da interface gráfica da aplicação.

Os serviços de informação são entidades através das quais diferentes módulos da aplicação têm acesso aos registros existentes. O *DatacenterInformationService* fornece informações sobre *datacenters* e, por conseguinte, seus *hosts* e áreas de armazenamento em rede (SAN). Já através do *CustomerInformationService*, pode-se obter dados relativos aos clientes e suas máquinas virtuais. Há ainda o *NetworkInformationService*, que disponibiliza informações sobre os *links* de comunicação entre registros de *datacenters* e clientes.

Os serviços de informação são a peça chave para a realização da tradução dos dados contidos nos registros do CloudReports para entidades análogas que são utilizadas pelo motor de simulação do CloudSim.

3.6 Algoritmos de decisão

Uma das características do perfil de utilização de recursos relativo a um cliente é a forma como os *datacenters* são escolhidos no momento de alocação das máquinas virtuais. O CloudReports oferece quatro mecanismos de escolha distintos, a saber: *Cheapest Datacenter*, *Fastest Datacenter*, *Lowest Latency* e *Round Robin*.

3.6.1 *Cheapest Datacenter*

Neste modelo, o cliente optará sempre pelo *datacenter* cuja soma dos custos de utilização é a mais baixa dentre todos os disponíveis. Caso haja mais de uma opção com o mesmo valor, a alocação das máquinas virtuais será realizada em estilo *round robin* entre os *datacenters* escolhidos. O Algoritmo 1 mostra como este mecanismo de decisão é implementado.

```
Entrada: Lista de datacenters disponíveis  
Saída : Lista de datacenters com menor custo de utilização  
1 lista ← obterDatacentersDisponíveis();  
2 menorCusto ← obterCustoDeUtilização(lista.primeiroItem);  
3 retorno ← criarLista(lista.primeiroItem);  
4 for item ← lista.segundoItem to lista.últimoItem do  
5     custo ← obterCustoDeUtilização(item);  
6     if custo igual a menorCusto then  
7         retorno.adicionar(item);  
8     end  
9     else if custo menor que menorCusto then  
10        menorCusto ← custo;  
11        retorno.limparLista();  
12        retorno.adicionar(item);  
13    end  
14 end  
15 return selecionarDatacentersAptos(retorno);
```

Algoritmo 1: Política de decisão do *broker Cheapest Datacenter*

Após obter a lista de todos os *datacenters* disponíveis, o valor de custo de utilização relativo ao primeiro elemento é atribuído à variável *menorCusto*. Na linha 4, inicia-se um laço onde o custo de utilização de cada um dos *datacenters* é verificado. Caso se identifique um valor igual ao armazenado na variável *menorCusto*, o *datacenter* é adicionado à lista de retorno. Se um valor inferior for identificado, todos os elementos da lista de retorno são removidos, e, em seguida, apenas o *datacenter* em questão é adicionado. Valores de custo de utilização maiores ao armazenados na variável *menorCusto* são desconsiderados. Por fim, filtra-se a lista de retorno, selecionando-se apenas os *datacenters* aptos a alocarem as máquinas virtuais do cliente.

3.6.2 *Fastest Datacenter*

Com esta alternativa de *broker*, o cliente sempre optará pelo *datacenter* com maior poder de processamento. A escolha, descrita no Algoritmo 2, é feita de forma análoga à empregada pelo *broker Cheapeast Datacenter*, optando-se sempre pelo *datacenter* com o maior número de MIPS disponíveis. Assim como no modelo anterior, a existência de mais de uma opção leva à adoção de uma política de alocação em estilo *round robin* entre os *datacenters* escolhidos.

```
Entrada: Lista de datacenters disponíveis  
Saída : Lista de datacenters com maior capacidade de processamento  
1 lista ← obterDatacentersDisponíveis();  
2 maiorMips ← obterMips(lista.primeiroItem);  
3 retorno ← criarLista(lista.primeiroItem);  
4 for item ← lista.segundoItem to lista.últimoItem do  
5   mips ← obterMips(item);  
6   if mips igual a maiorMips then  
7     retorno.adicionar(item);  
8   end  
9   else if mips maior que maiorMips then  
10    maiorMips ← mips;  
11    retorno.limparLista();  
12    retorno.adicionar(item);  
13  end  
14 end  
15 return selecionarDatacentersAptos(retorno);
```

Algoritmo 2: Política de decisão do *broker Fastest Datacenter*

3.6.3 *Lowest Latency*

Esta abordagem, descrita no Algoritmo 3, determina que o cliente sempre aloque suas máquinas virtuais no *datacenter* com menor latência de comunicação. Este mecanismo de escolha também é análogo aos mostrados nos algoritmos anteriores. O processo é composto pela obtenção uma lista de *datacenters* disponíveis, seleção dos menores valores de latência e filtragem das opções onde a alocação das máquinas virtuais é viável. A existência de múltiplas opções também acarreta em uma política de alocação de máquinas virtuais em estilo *round robin* entre os *datacenters* escolhidos.

```
Entrada: Lista de datacenters disponíveis  
Saída : Lista de datacenters com menor latência de rede  
1 lista ← obterDatacentersDisponíveis();  
2 menorLatência ← obterLatência(lista.primeiroItem);  
3 retorno ← criarLista(lista.primeiroItem);  
4 for item ← lista.segundoItem to lista.últimoItem do  
5   | latência ← obterLatência(item);  
6   | if latência igual a menorLatência then  
7   |   | retorno.adicionar(item);  
8   | end  
9   | else if latência menor que menorLatência then  
10  |   | menorLatência ← latência;  
11  |   | retorno.limparLista();  
12  |   | retorno.adicionar(item);  
13  | end  
14 end  
15 return selecionarDatacentersAptos(retorno);
```

Algoritmo 3: Política de decisão do *broker Lowest Latency*

O broker cujo mecanismo de escolha chama-se *Round Robin* utiliza a técnica de mesmo nome para efetuar uma escolha dentre todos os *datacenters* disponíveis sem qualquer critério de seleção. Tal implementação é útil como instrumento de comparação, confrontando-se seus resultados com os obtidos através de outros mecanismos de escolha.

3.7 Resumo

Este capítulo apresentou aspectos gerais do simulador desenvolvido por meio deste trabalho. A primeira seção tratou das ferramentas utilizadas, discutindo-se o caráter de código aberto da aplicação e os *frameworks* selecionados. Em seguida, uma análise descritiva foi realizada e os ambientes de simulação foram descritos. Fez-se então uma abordagem da estrutura do simulador, com análises dedicadas a cada um de seus módulos. O capítulo encerrou-se com um estudo dos algoritmos representativos das quatro políticas de decisão utilizadas pelos *brokers* dos clientes simulados.

Capítulo 4

Resultados

Neste capítulo, um cenário de simulação é criado e simulado por meio da ferramenta CloudReports. A composição do cenário é descrita na Seção 4.1 de acordo com as características do provedor de Infraestrutura como Serviço (IaaS) e dos clientes que fazem uso dos recursos oferecidos. Em seguida, na Seção 4.2, os resultados obtidos são discutidos através de uma análise do relatório final gerado pelo simulador.

4.1 Cenário simulado

As simulações realizadas por meio do CloudReports são compostas por dois tipos principais de agentes: um provedor de IaaS, que possui de um a muitos *datacenters*, e um conjunto de clientes que dispõem desse serviço. Os clientes utilizam os recursos oferecidos pelo provedor para o envio e a alocação de máquinas virtuais que, por sua vez, executam um conjunto de tarefas, aqui denominadas *cloudlets*. Conforme descrito no Capítulo 3, a dinâmica de escolha de *datacenters* para o envio e alocação de máquinas virtuais e a posterior execução de *cloudlets* é ditada pelo perfil de utilização dos clientes e pelos recursos oferecidos pelo provedor.

O cenário aqui simulado é formado por um provedor que dispõe de quatro *datacenters* distribuídos em cidades distintas do país e quatro clientes com diferentes perfis de utilização. As características de cada um desses agentes são descritas a seguir.

4.1.1 Provedor

A infraestrutura do provedor a ser simulado é composta por quatro *datacenters* identificados pelos nomes das cidades onde se encontram, a saber: Fortaleza, São Paulo, Rio de Janeiro e Brasília. A Tabela 4.1 mostra detalhadamente os recursos disponíveis nesses *datacenters* e algumas de suas características principais. Os valores de custo de utilização de processamento, memória, armazenamento e banda são fornecidos em unidades monetárias.

A arquitetura de processamento, o sistema operacional, o hipervisor utilizado e os custos de utilização são atributos únicos de um *datacenter*, sendo automaticamente aplicados a todos os seus nós. Por outro lado, o poder de processamento, a quantidade de memória de sistema e a capacidade de armazenamento são características de *hosts*, que podem ser customizados individualmente, criando-se *datacenters* com recursos heterogêneos.

Há de se ressaltar, porém, que todos os *datacenters* aqui simulados são compostos por nós homogêneos, ou seja, todos os *hosts* possuem a mesma configuração. A Tabela 4.2 exibe a configuração individual de cada um desses *hosts*. Estes atributos exercem grande influência no fluxo de eventos da simulação, pois, como descrito na Seção 3.6, para que uma máquina virtual seja efetivamente alocada, é necessário que ao menos um *host* do provedor seja capaz de atender simultaneamente a seus requisitos de recursos e ao perfil de utilização do cliente.

A capacidade de armazenamento dos *datacenters* não coincide com a soma dos valores de capacidade de armazenamento individual dos *hosts* que os compõem devido à existência de Áreas de Armazenamento em Rede (SAN). Cada um dos *datacenters* simulados foi configurado com uma SAN de 10 TB, cujo custo de utilização coincide com o custo de armazenamento exposto na Tabela 4.1.

O processo de simulação também conta com uma reprodução simples dos enlaces de comunicação existentes entre *datacenters* e clientes. Essas conexões são representadas através de valores de largura de banda e latência de rede. As características dos enlaces cuja fonte consiste em um *datacenter* são mostradas nas Tabelas 4.3, 4.4, 4.5 e 4.6.

Tabela 4.1: Características dos *datacenters* simulados.

	Fortaleza	São Paulo	Rio de Janeiro	Brasília
Arquitetura	x86	x86	x86	x86
Sistema Operacional	Linux	Linux	Linux	Linux
Hipervisor	Xen	Xen	Xen	Xen
Número de <i>hosts</i>	20	60	40	25
Processadores	80	240	160	100
Capacidade de processamento (MIPS)	2M	24M	14.4M	2.5M
Memória de sistema	80 GB	1 TB	640 GB	200 GB
Capacidade de armazenamento	30 TB	190 TB	50 TB	35 TB
Custo de processamento (por segundo)	0,001 u.m.	0,012 u.m.	0,003 u.m.	0,002 u.m.
Custo de memória (por MB)	0,03 u.m.	0,04 u.m.	0,07 u.m.	0,06 u.m.
Custo de armazenamento (por MB)	0,001 u.m.	0,001 u.m.	0,002 u.m.	0,002 u.m.
Custo de banda (por bit)	0,03 u.m.	0,04 u.m.	0,05 u.m.	0,05 u.m.

Tabela 4.2: Características dos *hosts* simulados.

	Fortaleza	São Paulo	Rio de Janeiro	Brasília
Processadores	4	4	4	4
MIPS por processador	25K	100K	90K	25K
Memória de sistema	4 GB	16 GB	16 GB	8 GB
Capacidade de armazenamento	1 TB	3 TB	1 TB	1 TB
Largura de banda	10000 kbps	15000 kbps	15000 kbps	10000 kbps
Escalonador de máquinas virtuais	Compartilhado no tempo	Compartilhado no espaço	Compartilhado no tempo	Compartilhado no espaço

Tabela 4.3: Características dos enlaces cuja fonte é o *datacenter* localizado em Fortaleza.

Fonte	Destino	Largura de banda (kbps)	Latência (s)
Fortaleza	São Paulo	1000	0,2
Fortaleza	Rio de Janeiro	1000	0,2
Fortaleza	Brasília	1000	0,25
Fortaleza	Cliente 1	1000	0,3
Fortaleza	Cliente 2	1000	0,2
Fortaleza	Cliente 3	1000	0,25
Fortaleza	Cliente 4	1000	0,2

Tabela 4.4: Características dos enlaces cuja fonte é o *datacenter* localizado em Brasília.

Fonte	Destino	Largura de	
		banda (kbps)	Latência (s)
Brasília	São Paulo	1500	0,1
Brasília	Rio de Janeiro	1500	0,1
Brasília	Fortaleza	1500	0,2
Brasília	Cliente 1	1500	0,2
Brasília	Cliente 2	1500	0,2
Brasília	Cliente 3	1500	0,2
Brasília	Cliente 4	1500	0,2

Tabela 4.5: Características dos enlaces cuja fonte é o *datacenter* localizado no Rio de Janeiro.

Fonte	Destino	Largura de	
		banda (kbps)	Latência (s)
Rio de Janeiro	São Paulo	4000	0,1
Rio de Janeiro	Brasília	1500	0,1
Rio de Janeiro	Fortaleza	1000	0,25
Rio de Janeiro	Cliente 1	3000	0,1
Rio de Janeiro	Cliente 2	2000	0,1
Rio de Janeiro	Cliente 3	2000	0,2
Rio de Janeiro	Cliente 4	2000	0,15

Tabela 4.6: Características dos enlaces cuja fonte é o *datacenter* localizado em São Paulo.

Fonte	Destino	Largura de banda (kbps)	Latência (s)
São Paulo	Rio de Janeiro	4000	0,1
São Paulo	Brasília	4000	0,15
São Paulo	Fortaleza	4000	0,2
São Paulo	Cliente 1	3000	0,1
São Paulo	Cliente 2	3000	0,1
São Paulo	Cliente 3	3000	0,1
São Paulo	Cliente 4	4000	0,15

4.1.2 Clientes

A composição do cenário simulado é concluída por quatro clientes, aqui denominados Cliente 1, Cliente 2, Cliente 3 e Cliente 4. Suas máquinas virtuais devem ser enviadas e alocadas em *hosts* que atendam aos requisitos mínimos de recursos das mesmas. Ademais, os *hosts* escolhidos devem pertencer a um *datacenter* que siga o perfil de utilização do cliente. Posteriormente, os clientes enviam as *cloudlets* que serão executadas por essas máquinas virtuais e enviadas de volta após a execução.

A Tabela 4.7 mostra as características das máquinas virtuais de cada um dos clientes simulados. Assim como os atributos dos *hosts*, os recursos exigidos por cada máquina virtual desempenham um papel chave no fluxo de eventos da simulação, podendo inclusive determinar a inviabilidade da mesma. Isto ocorre quando os *datacenters* apontados pelo broker do cliente não possuem *hosts* capazes de atender à demanda de recursos de suas máquinas virtuais. Neste caso, a simulação não pode ser iniciada, e uma mensagem de erro é exibida pelo simulador.

A exibição de características no formato da Tabela 4.7 é possível devido ao fato de todos os clientes possuírem máquinas virtuais com configurações idênticas. Esta

abordagem foi adotada com o intuito de simplificar a análise dos dados. Contudo, ressalta-se que o CloudReports permite a criação de múltiplas configurações de máquinas virtuais para cada cliente simulado.

Os perfis de utilização de recursos são mostrados na Tabela 4.8. Cada cliente emprega uma das políticas de *broker* oferecidas pelo simulador para que se verifique a influência das mesmas nos resultados obtidos. É também especificado o número de *cloudlets* a ser enviado e suas características, que são equivalentes para um determinado cliente.

Por fim, os valores de largura de banda e latência dos *links* de comunicação cuja fonte consiste em um cliente são mostrados nas Tabelas 4.9, 4.10, 4.11 e 4.12. Nestes casos, como não há interação entre clientes durante o processo de simulação, apenas as conexões cliente-*datacenter* são reproduzidas pelo simulador.

Tabela 4.7: Configurações das máquinas virtuais dos clientes simulados.

	Cliente 1	Cliente 2	Cliente 3	Cliente 4
Quantidade	15	5	4	7
Hipervisor	Xen	Xen	Xen	Xen
Processadores	2	1	4	2
MIPS por processador	6000	4000	10000	6000
Memória de sistema	3 GB	1 GB	2 GB	2GB
Largura de banda	1000 kbps	1000 kbps	512 kbps	1000 kbps
Escalonador de <i>cloudlets</i>	Compartilhado no tempo	Compartilhado no espaço	Compartilhado no espaço	Compartilhado no tempo

Tabela 4.8: Perfis de utilização de recursos dos clientes simulados.

	Cliente 1	Cliente 2	Cliente 3	Cliente 4
Política do broker	<i>Round Robin</i>	<i>Cheapeast Datacenter</i>	<i>Fastest Datacenter</i>	<i>Lowest Latency</i>
Número de <i>cloudlets</i>	100	150	250	350
Mínimo de processadores para execução	2	1	2	1
Comprimento das <i>cloudlets</i> (MIPS)	45K	20K	50K	30K

Tabela 4.9: Características dos enlaces cuja fonte é o Cliente 1.

Fonte	Destino	Largura de banda (kbps)	Latência (s)
Cliente 1	São Paulo	512	0,2
Cliente 1	Brasília	512	0,2
Cliente 1	Fortaleza	512	0,25
Cliente 1	Rio de Janeiro	512	0,3

Tabela 4.10: Características dos enlaces cuja fonte é o Cliente 2.

Fonte	Destino	Largura de	
		banda (kbps)	Latência (s)
Cliente 2	São Paulo	1000	0,15
Cliente 2	Brasília	1000	0,2
Cliente 2	Fortaleza	1000	0,25
Cliente 2	Rio de Janeiro	1000	0,15

Tabela 4.11: Características dos enlaces cuja fonte é o Cliente 3.

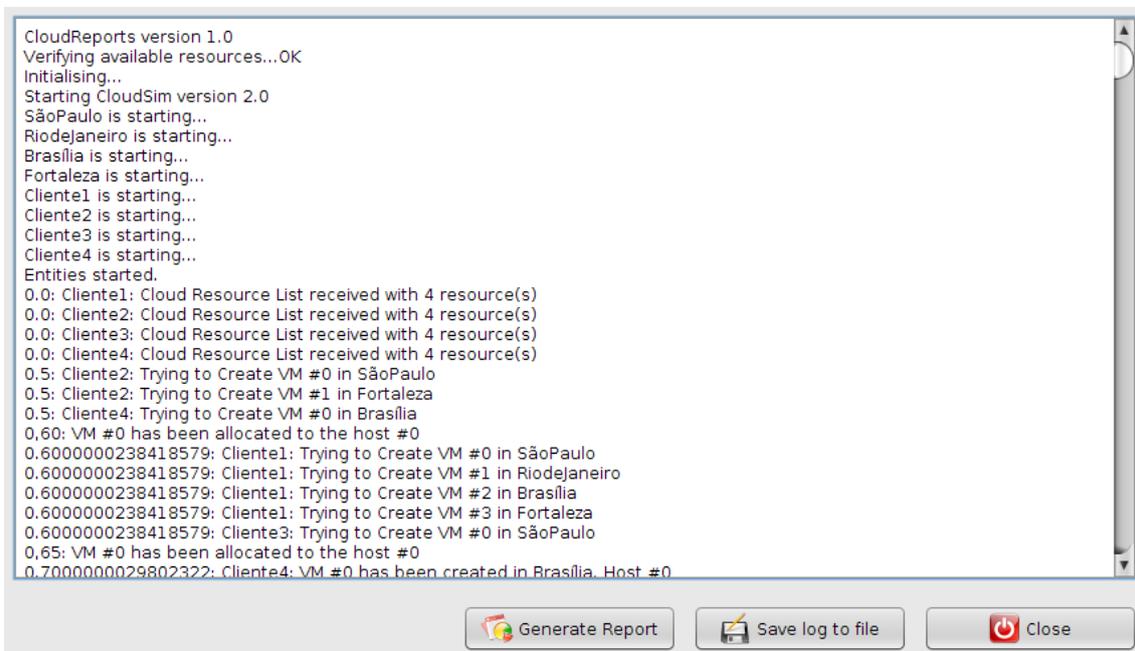
Fonte	Destino	Largura de	
		banda (kbps)	Latência (s)
Cliente 3	São Paulo	512	0,15
Cliente 3	Brasília	512	0,2
Cliente 3	Fortaleza	512	0,3
Cliente 3	Rio de Janeiro	512	0,25

Tabela 4.12: Características dos enlaces cuja fonte é o Cliente 4.

Fonte	Destino	Largura de	
		banda (kbps)	Latência (s)
Cliente 4	São Paulo	1000	0,15
Cliente 4	Brasília	1000	0,1
Cliente 4	Fortaleza	1000	0,25
Cliente 4	Rio de Janeiro	1000	0,15

4.2 Resultados obtidos

Após a criação do cenário descrito nas seções anteriores e a realização do processo de simulação, um relatório final é produzido pelo simulador. Os dados obtidos são representados por meio várias modalidades de gráficos, visando uma melhor compreensão das informações geradas e expondo aspectos relevantes referentes ao comportamento de todas as entidades simuladas. Adicionalmente, um *log* é produzido (Figura 4.1) com descrições detalhadas do fluxo de eventos da simulação. Nas seções subsequentes, são analisados cada um dos gráficos que compõem o relatório produzido pelo CloudReports.



```
CloudReports version 1.0
Verifying available resources...OK
Initialising...
Starting CloudSim version 2.0
SãoPaulo is starting...
Riodejaneiro is starting...
Brasília is starting...
Fortaleza is starting...
Cliente1 is starting...
Cliente2 is starting...
Cliente3 is starting...
Cliente4 is starting...
Entities started.
0.0: Cliente1: Cloud Resource List received with 4 resource(s)
0.0: Cliente2: Cloud Resource List received with 4 resource(s)
0.0: Cliente3: Cloud Resource List received with 4 resource(s)
0.0: Cliente4: Cloud Resource List received with 4 resource(s)
0.5: Cliente2: Trying to Create VM #0 in SãoPaulo
0.5: Cliente2: Trying to Create VM #1 in Fortaleza
0.5: Cliente4: Trying to Create VM #0 in Brasília
0.60: VM #0 has been allocated to the host #0
0.6000000238418579: Cliente1: Trying to Create VM #0 in SãoPaulo
0.6000000238418579: Cliente1: Trying to Create VM #1 in Riodejaneiro
0.6000000238418579: Cliente1: Trying to Create VM #2 in Brasília
0.6000000238418579: Cliente1: Trying to Create VM #3 in Fortaleza
0.6000000238418579: Cliente3: Trying to Create VM #0 in SãoPaulo
0.65: VM #0 has been allocated to the host #0
0.7000000029802322: Cliente4: VM #0 has been created in Brasília. Host #0
```

Figura 4.1: Tela de exibição do *log* de simulação.

4.2.1 Máquinas virtuais alocadas por *datacenter*

O gráfico de máquinas virtuais alocadas por *datacenter* é mostrado na Figura 4.2. O eixo vertical ilustra cada um dos *datacenters* simulados, enquanto o eixo horizontal exhibe a quantidade de máquinas virtuais alocadas. As cores das barras identificam qual cliente é responsável pela utilização dos recursos.

Por conta da política de *broker Round Robin* (vide Tabela 4.8), o Cliente 1 executa parte de suas máquinas virtuais em cada um dos *datacenters* disponíveis. O Cliente 2, por sua vez, emprega a política *Cheapest Datacenter*, que seleciona

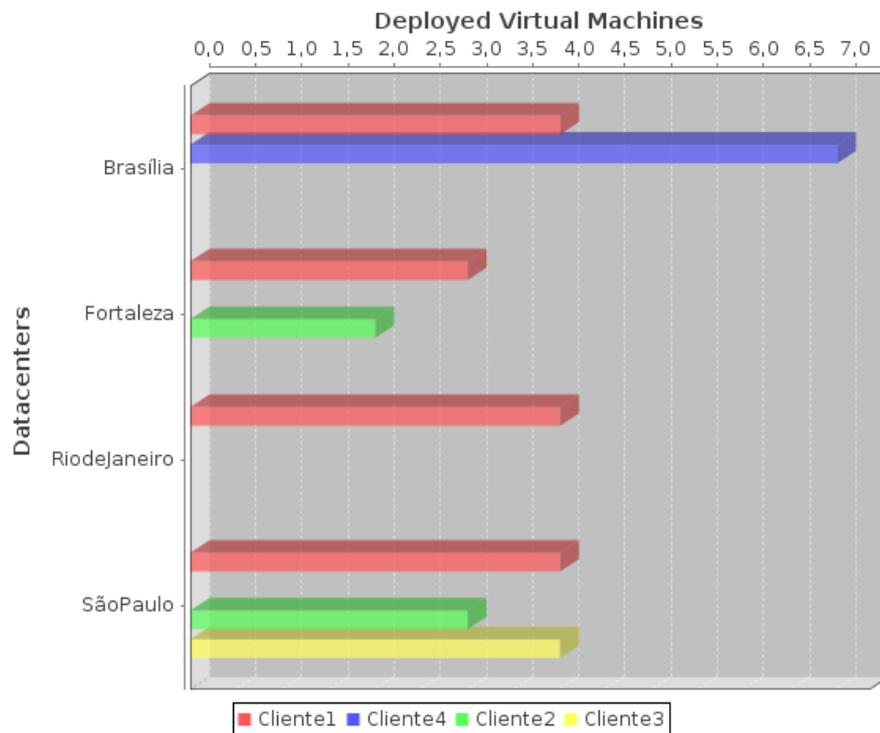
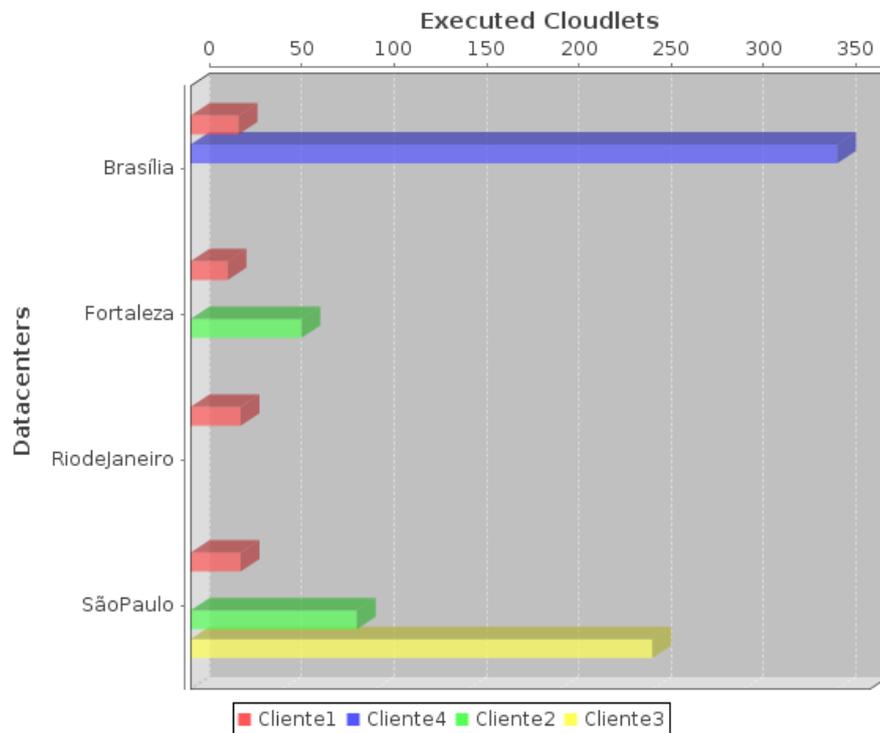


Figura 4.2: Máquinas virtuais alocadas por *datacenter*.

os menores custos de utilização. Como os *datacenters* localizados em São Paulo e Fortaleza possuem os menores preços, ambos alocam as máquinas virtuais do Cliente 2. A política *Fastest Datacenter* faz com que o Cliente 3 opte por alocar todas suas máquinas virtuais no *datacenter* de São Paulo, pois este possui a maior capacidade de processamento (vide Tabela 4.1). Por fim, o *datacenter* com menor latência de comunicação com o Cliente 4, que emprega a política *Lowest Latency*, está localizado em Brasília e aloca todas as sete máquinas virtuais desse cliente.

4.2.2 *Cloudlets* executadas por *datacenter*

O gráfico de *cloudlets* executadas por *datacenter* tem um formato semelhante ao da Figura 4.2, visto que as *cloudlets* enviadas por cada cliente são executadas por suas respectivas máquinas virtuais. Dessa forma, o número de máquinas virtuais alocadas por cada *datacenter* é aproximadamente proporcional ao número de *cloudlets* executadas pelos mesmos, como mostrado na Figura 4.3.

Figura 4.3: *Cloudlets* executadas por *datacenter*.

4.2.3 Custos por *datacenter*

Os custos produzidos por cada cliente em um determinado *datacenter* levam em consideração seu preço de utilização e a quantidade de recursos alocada para as máquinas virtuais. A Figura 4.4 mostra os custos gerados pelos quatro clientes simulados. Devido a sua política de *broker Round Robin*, o Cliente 1 gera custos em todos os *datacenters*. A política *Cheapest Datacenter* garante, como esperado, os menores custos de utilização ao Cliente 2. Já os clientes 3 e 4 geram custos apenas nos *datacenters* selecionados por suas políticas de *broker* para a alocação de suas máquinas virtuais.

4.2.4 Recursos utilizados por *datacenter*

A quantidade de recursos utilizados em cada um dos *datacenters* simulados é representada na Figura 4.5. São mostrados os valores relativos à utilização de CPU (em MIPS), memória de sistema e banda de comunicação durante todo o período de simulação.

O método de distribuição de recursos utilizado nos *datacenters* estabelece que, uma vez alocada, uma máquina virtual sempre utiliza todos os recursos referentes

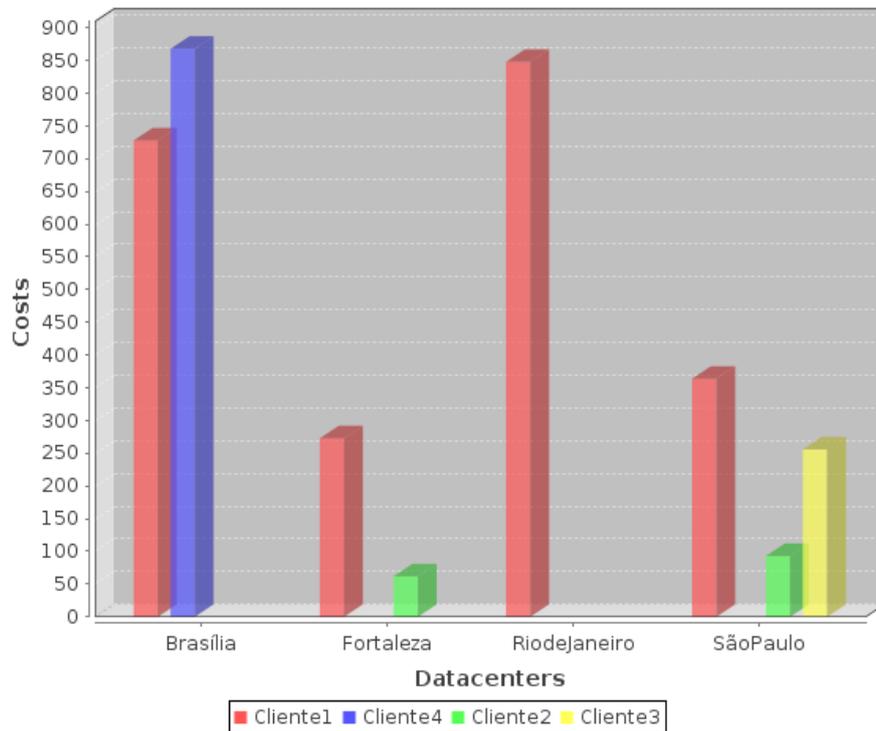


Figura 4.4: Custos produzidos nos *datacenters* simulados.

à sua configuração. Em conformidade com esta característica, os gráficos da Figura 4.5 apresentam curvas semelhantes para os três tipos de recursos representados.

4.2.5 *Cloudlets* executadas por máquina virtual

Os gráficos da Figura 4.6 mostram a quantidade de *cloudlets* que cada máquina virtual de um cliente executou durante o processo de simulação. Em todos os casos, observa-se que a quantidade de *cloudlets* é distribuída de forma aproximadamente uniforme entre as máquinas virtuais disponíveis. Isto se deve ao fato de o simulador utilizar um algoritmo *round robin* para atribuir números identificadores que vinculam uma *cloudlet* à máquina virtual que a executará.

A política de *broker* do Cliente 1 faz com que máquinas virtuais presentes em todos os *datacenters* simulados executem suas *cloudlets*, como já havia sido observado na Figura 4.3. O Cliente 2 tem duas máquinas virtuais alocadas em Fortaleza e três alocadas em São Paulo, cada uma delas executando precisamente trinta *cloudlets*. No caso do Cliente 3, de acordo com a política de *broker Fastest Datacenter*, quatro máquinas virtuais são alocadas em São Paulo e executam pouco mais de sessenta *cloudlets*. O Cliente 4 tem trezentas e cinquenta *cloudlets*

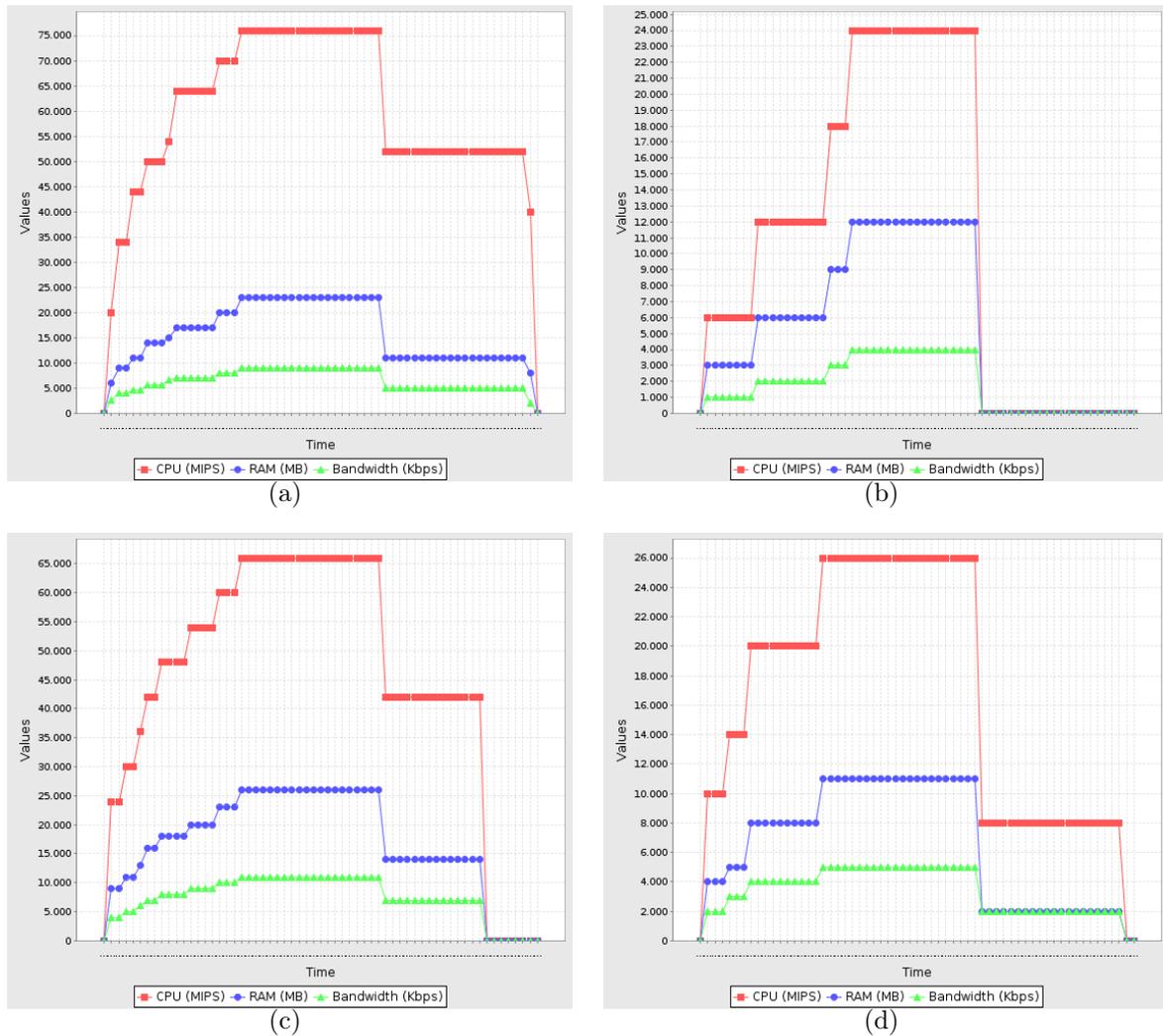


Figura 4.5: Recursos utilizados nos *datacenters* de (a) São Paulo, (b) Rio de Janeiro, (c) Brasília e (d) Fortaleza.

distribuídas igualmente entre suas sete máquinas virtuais alocadas no *datacenter* de Brasília.

4.2.6 Recursos utilizados por máquina virtual

De acordo com o que foi descrito na Seção 4.2.4, as máquinas virtuais simuladas sempre utilizam a quantidade total de recursos especificada em suas configurações. Desta forma, uma vez alocada, a máquina virtual mantém sua taxa de utilização de recursos constante, assim como mostrado nos gráficos da Figura 4.7 para o Cliente 3. Como todas as máquinas virtuais possuem a mesma configuração, percebe-se que a diferença entre os gráficos gerados reside apenas no momento em que cada uma

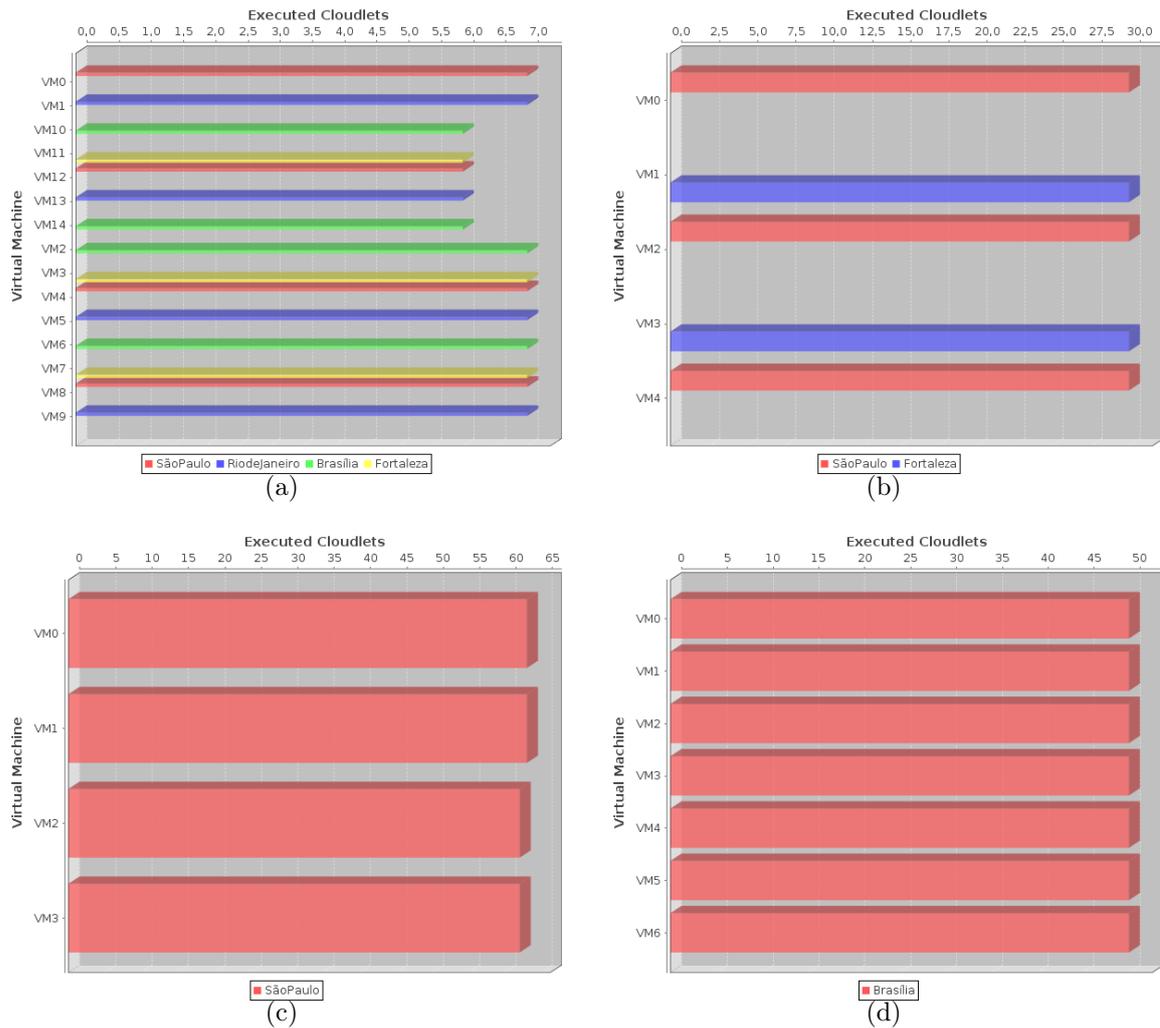


Figura 4.6: *Cloudlets* executadas pelas máquinas virtuais do (a) Cliente 1, (b) Cliente 2, (c) Cliente 3 e (d) Cliente 4.

delas é alocada. Devido à semelhança entre os gráficos e visando a simplificação na exibição dos dados, as taxas de utilização de recursos das máquinas virtuais dos demais clientes serão omitidas.

4.2.7 Tempo de execução por *cloudlet*

Esta modalidade de gráfico mostra o tempo de partida e o tempo de chegada de todas as *cloudlets* de um dado cliente, resultando em uma representação visual do tempo de execução das mesmas. Ademais, duas linhas horizontais exibem as médias aritméticas desses valores, e a distância que as separa representa o tempo médio de execução das *cloudlets*.

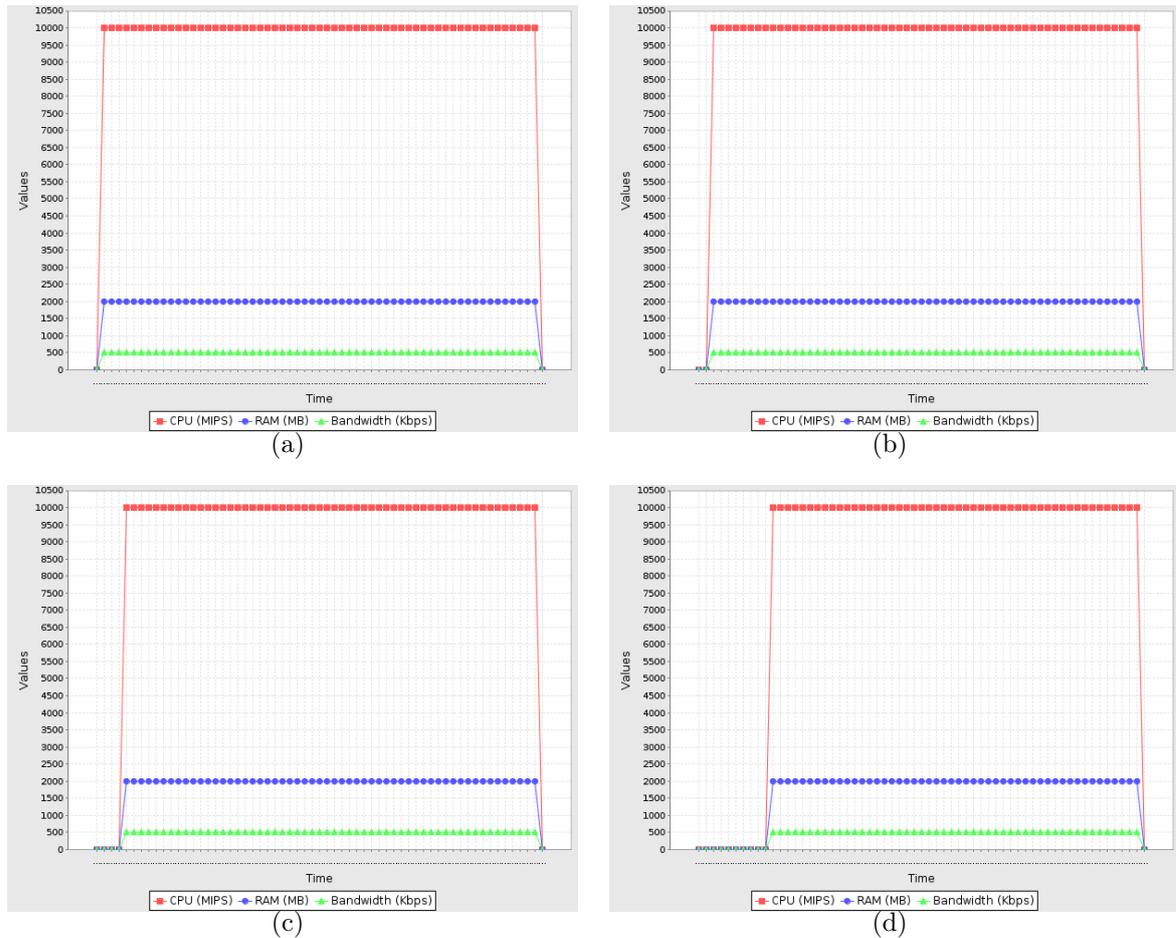


Figura 4.7: Recursos utilizados pelas máquinas virtuais (a) VM0, (b) VM1, (c) VM2 e (d) VM3 do Cliente 3.

De acordo com o algoritmo *round robin* empregado pelo simulador, o vínculo entre N *cloudlets* e K máquinas virtuais de um dado cliente é estabelecido de acordo com a expressão $VM_k \leftrightarrow C_{(i+k)}$, com $0 \leq n < N$ e $0 \leq k < K$. Esta relação explica o efeito “rampa” observado na Figura 4.8 para os tempos de partida das *cloudlets* do Cliente 1. Além disso, o fato do escalonador das máquinas virtuais deste cliente compartilhar a execução de *cloudlets* no tempo é parcialmente responsável pelos “vales” existentes na curva dos tempos de chegada. Como algumas máquinas virtuais recebem um menor número de *cloudlets*, seu tempo total de execução é reduzido, o que dá origem ao fenômeno.

As curvas de tempo de execução para os Clientes 2 e 3 se assemelham devido aos escalonadores de suas máquinas virtuais compartilharem a execução de *cloudlets* no espaço, obtendo-se assim os gráficos no formato de “escada” mostrados na Figura 4.9. Contudo, as linhas que compõem o gráfico do Cliente 2 têm um aspecto mais

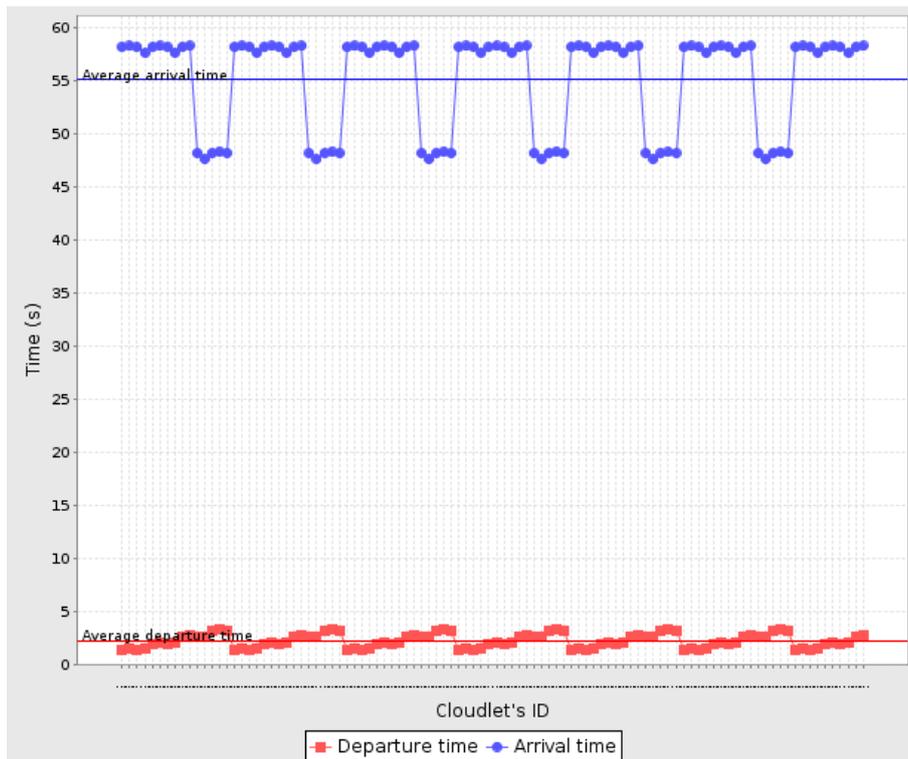


Figura 4.8: Tempos de execução de *cloudlets* do Cliente 1.

irregular devido ao fato de suas *cloudlets* serem executadas em dois *datacenters* distintos. A variação dos valores de latência de rede influencia diretamente o tempo de chegada das *cloudlets*, gerando a irregularidade observada.

Por fim, as máquinas virtuais do Cliente 4 são executadas em um mesmo *datacenter*, recebem quantidades iguais de *cloudlets* e possuem escalonadores que compartilham a execução das mesmas no tempo. Neste caso, todas as *cloudlets* são enviadas simultaneamente aos *datacenters*, compartilham o tempo de CPU das máquinas virtuais durante sua execução e, ao fim do processo, são enviadas de volta ao cliente também simultaneamente, gerando o gráfico exibido na Figura 4.10.

4.3 Resumo

Este capítulo discutiu os resultados obtidos por meio da utilização do simulador CloudReports. O cenário simulado foi descrito detalhadamente na Seção 4.1, onde as características do provedor de IaaS e de seus clientes foram fornecidas. A Seção 4.2 exibiu os resultados, analisando cada um dos gráficos gerados pela aplicação.

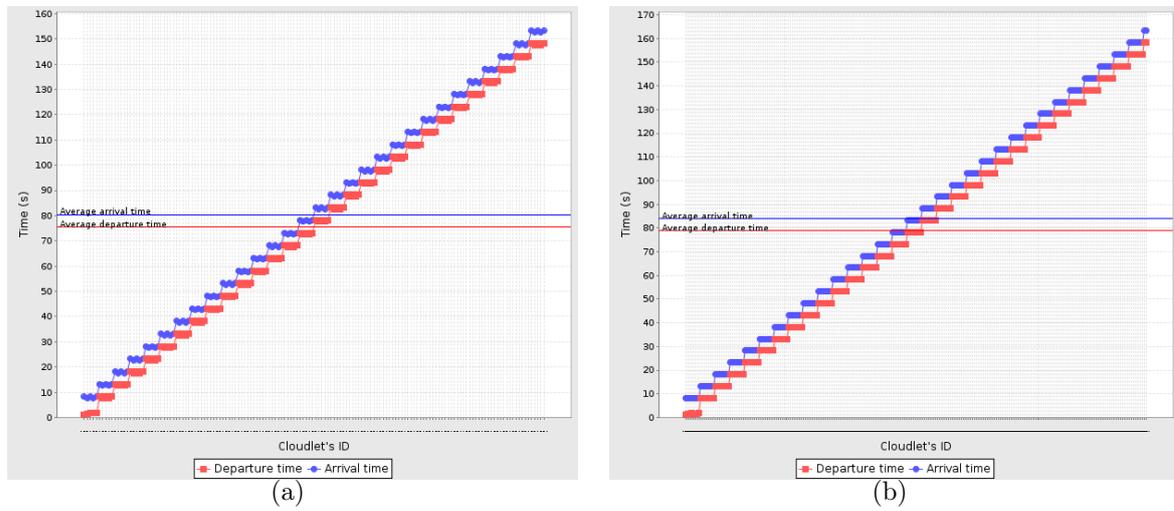


Figura 4.9: Tempos de execução de *cloudlets* do (a) Cliente 2 e (b) Cliente 3.

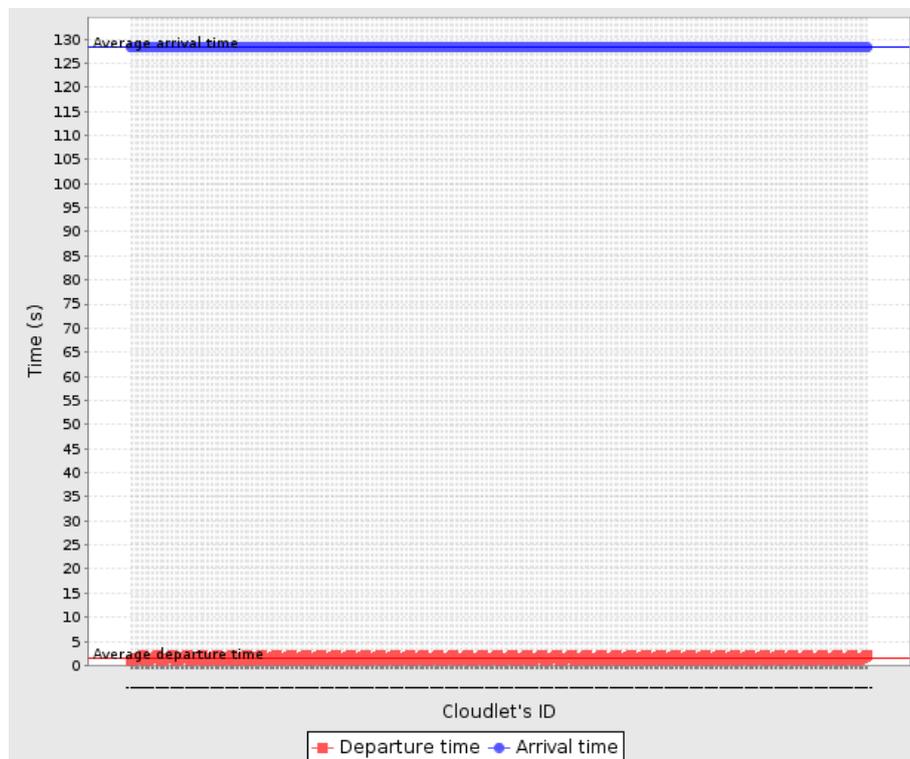


Figura 4.10: Tempos de execução de *cloudlets* do Cliente 4.

Capítulo 5

Conclusões

Neste capítulo, são feitas considerações finais acerca do trabalho realizado. A Seção 5.1 lista as principais contribuições provindas do mesmo, enquanto a Seção 5.2 trata das limitações ainda existentes e suas implicações. A Seção 5.3 aborda as melhorias e correções programadas para o CloudReports, além de propor o desenvolvimento de novas funcionalidades e discutir o futuro planejado para o projeto.

5.1 Contribuições

A principal contribuição deste trabalho consiste na criação de uma ferramenta de *software*, intitulada CloudReports, cuja função é o provimento de uma interface gráfica e funcionalidades para geração de relatórios que, em conjunto com os recursos oferecidos pelo *framework* CloudSim, constituem um meio prático e flexível para a simulação de sistemas distribuídos baseados no modelo proposto pelos princípios da Computação em Nuvem. Essa ferramenta dispõe de recursos que possibilitam a configuração de uma quantidade virtualmente ilimitada de cenários de simulação compostos pela representação de um provedor de Infraestrutura como Serviço (IaaS) e de seus clientes, que lançam mão desse serviço para a alocação de máquinas virtuais. Ademais, o simulador oferece ao usuário uma interface gráfica de fácil utilização e que, atrelada ao recurso de geração de relatórios, exibe uma compilação dos resultados das simulações por meio de gráficos e listagens de características do ambiente.

Uma importante etapa do desenvolvimento deste trabalho tratou da integração

necessária entre a ferramenta criada (CloudReports) e o *framework* de simulação utilizado (CloudSim). Durante este processo, algumas falhas que influenciavam diretamente os resultados obtidos nas simulações foram encontradas e reportadas aos criadores do CloudSim por meio da lista de discussão do projeto. É esperado que, em versões futuras do *framework*, essas falhas tenham sido sanadas. Além disso, algumas melhorias para o mesmo também foram propostas, como o uso da classe BigDecimal para o cálculo dos custos de utilização de recursos, o que diminui a possibilidade de erros relacionados à representação de valores monetários, e a mudança na abordagem empregada para o cálculo do custo de processamento, que, na versão atual, desconsidera a existência de *datacenters* com recursos heterogêneos.

Complementarmente à identificação de falhas e à proposição de melhorias, algumas funcionalidades ausentes no *framework* de simulação foram implementadas. A principal delas trata da verificação da viabilidade de simulação dos ambientes. Através deste recurso, uma análise das entidades a serem simuladas é realizada previamente ao início da simulação. Caso seja constatada sua inviabilidade, a simulação é abortada e uma mensagem é exibida ao usuário, o que previne falhas na aplicação. Embora funcionais, estes recursos não foram incorporados na versão oficial do *framework*, permanecendo como características próprias do simulador desenvolvido.

Dados os resultados satisfatórios obtidos e a até então inexistência de um simulador correlato, acredita-se ter dado uma importante contribuição para o campo de pesquisa e desenvolvimento na área da Computação em Nuvem.

5.2 Limitações

Dentre as limitações apresentadas pelo simulador, pode-se destacar a forma de representação das *cloudlets* atreladas ao perfil de utilização de recursos dos clientes. Suas configurações são realizadas de forma estática, sem qualquer modelo que simule variações de suas características, o que implica na existência de apenas um tipo de *cloudlet* por cliente. Ressalta-se ainda o número restrito de implementações de escalonadores e modelos provimento e utilização de recursos.

No que concerne às limitações impostas pelo CloudSim, as falhas apresentadas pelos mecanismos de migração de máquinas virtuais tornam-se evidentes. Isto se deve em grande parte ao estado inicial de desenvolvimento de um modelo de consumo energético dos *datacenters*, que representa um dos critérios de decisão para as políticas de migração. Adicionalmente, o modo de representação dos enlaces de

comunicação ainda é excessivamente simplificado, e sua influência nos resultados finais ainda é questionável.

5.3 Perspectivas para trabalhos futuros

Tendo como base as limitações expostas na seção anterior, algumas melhorias e correções estão programadas para tornar o CloudReports um simulador mais robusto e com resultados ainda mais precisos e consistentes.

A integração do módulo de registro de entidades com um modelo de banco de dados certamente tornará a utilização de memória de sistema menos dispendiosa. A criação de políticas de *broker* e escalonadores mais sofisticados com a aplicação de técnicas de inteligência computacional representará um significativo acréscimo de qualidade nos resultados observados, e a implementação de novos modelos de provimento e utilização de recursos os aproximarão aos obtidos em um cenário real. A sofisticação do modelo de representação dos enlaces de comunicação consiste em uma melhoria crucial para o simulador, o que elevará sobremaneira a consistência do processo de simulação.

É prevista ainda uma participação ativa no desenvolvimento do *framework* de simulação CloudSim através do projeto criado no portal *Google Code*. As correções nos mecanismos de migração de máquinas virtuais e a criação de um modelo robusto que represente o consumo energético dos recursos computacionais simulados criam novas possibilidades de pesquisa, inclusive em campos como o *Green Computing*.

Este trabalho encerra-se com a perspectiva de criação de um projeto de código aberto onde a participação de um número maior desenvolvedores será incentivada. É também considerada a possibilidade de sua integração com o projeto oficial do *framework* CloudSim.

Referências Bibliográficas

AMAZON. *Amazon EC2 Instance Types*. 2010. Disponível em <http://aws.amazon.com/ec2/instance-types>. Último acesso em 22/09/2010.

ARMBRUST, M. *et al.* Above the clouds: A berkeley view of cloud computing. 2009. Disponível em <http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html>. Último acesso em 25/10/2010.

BELOGLAZOV, A.; BUYYA, R. Energy efficient allocation of virtual machines in cloud data centers. *Proceedings of the 10th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2010.

BREY, B. B. *Applying PIC18 Microcontrollers. Architecture, Programming, and Interfacing using C and Assembly*. 2008. Pearson Prentice Hall.

BUNKER, D. T. G. *Delivering Utility Computing - Business-driven IT Optimization*. 2006. John Wiley Sons.

BUYYA, R. Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility. *International Symposium on Cluster Computing and the Grid*, 2009.

BUYYA, R. *et al.* *The Grid Economy*. 2005.

BUYYA, R.; BUDENDORFER, K. *Market-Oriented Grid and Utility Computing*. 2010. John Wiley and Sons.

BUY YA, R.; RANJAN, R.; CALHEIROS, R. N. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. 2009.

BUY YA, R.; YEO, C. S.; VENUGOPAL, S. *Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities*. 2008a.

BUY YA, R. *et al.* *Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility*. 2008b.

CALHEIROS, R. N.; BUY YA, R.; ROSE, C. A. F. Building an automated and self-configurable emulation testbed for grid applications. 2010a.

CALHEIROS, R. N. *et al.* Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services. *International Conference on Parallel Processing*, 2009.

CALHEIROS, R. N. *et al.* Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *International Journal of Software: Practice and Experience*, 2010b.

CLOUDSIM. *CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services*. 2010. Disponível em <http://www.cloudbus.org/cloudsim/>.

COULOURIS, G. *et al.* *Distributed Systems: Concepts and Design*. 2005. 4th edition. Pearson Addison Wesley.

ENDREI, M. *et al.* *Patterns: Service-Oriented Architecture and Web Services*. 2004. RedBooks.

FOSTER, C. K. I. *Computational Grids*. 1998.

FOSTER, I. *et al.* The anatomy of the grid: Enabling scalable virtual organization. *International Journal of High Performance Computing Applications*, 2001.

GARTNER. *Gartner Says Worldwide SaaS Revenue in the Enterprise Application Markets Will Grow 27 Per Cent in 2008*. 2008. Disponível em <http://www.gartner.com/it/page.jsp?id=783212>. Último acesso em 12/10/2010.

- GONG, L. *et al.* *Inse Java 2 Platform Security: Architecture, API Design and Implementation*. 1999. 2nd Edition. Sun Microsystems Series.
- IBM. *IBM Server Optimization and Integration Services – VMware server virtualization*. 2008. IBM White Paper.
- IDC. *IDC Predictions 2009: An Economic Pressure Cooker Will Accelerate the IT Industry Transformation*. 2008.
- KIM, K. H.; BELOGLAZOV, A.; BUYYA, R. Power-aware provisioning of cloud resources for real-time services. *Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science*, 2009.
- LEGRAND, A.; MARCHAL, L.; CASANOVA, H. Scheduling distributed applications: the simgrid simulation framework. *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2003.
- LOHR, S. *When Cloud Computing Doesn't Make Sense*. 2009. The New York Times Bits Blog. Disponível em <http://bits.blogs.nytimes.com/2009/04/15/when-cloud-computing-doesnt-make-sense>. Último acesso em 23/10/2010.
- MALIS, A. Routing over large clouds (rolc) charter. *Part of the 32 IETF meeting minutes*, 1993. Disponível em <http://www.ietf.org/proceedings/32/charters/rolc-charter.html>.
- MCKINSEY. *Clearing the Air on Cloud Computing*. 2009. McKinsey and Company White Paper.
- MUSEUM, C. H. *The Babbage Engine*. 2010. Disponível em <http://www.computerhistory.org/babbage>. Último acesso em 20/10/2010.
- PALLIS, G. Cloud computing: The new frontier of internet computing. *View from the cloud*, 2010.
- REESE, G. *Cloud Application Architectures*. 2009. O'Reilly.
- RIGHTSCALE. *Define Cloud Computing*. 2010. RightScale Blog. Disponível em <http://blog.rightscale.com/2008/05/26/define-cloud-computing>. Último acesso em 10/10/2010.

RITTINGHOUSE, J. W.; RANSOME, J. F. *Cloud Computing. Implementation, Management and Security*. 2010. CRC Press.

SCHUBERT, L. *The Future of Cloud Computing - Opportunities for European Cloud Computing Beyond 2010*. 2010. European Commission - Information Society and Media.

SOLTER, N. A. *et al. OpenSolaris Bible*. 2009. Wiley Publishing.

STANOEVSKA-SLABEVA, K. *et al. Grid and Cloud Computing - A Business Perspective on Technology and Applications*. 2010. Springer.

SULISTIO, A. *et al. A toolkit for modelling and simulating data grids: an extension to gridsim*. *Wiley InterScience*, 2008.

SUN. *A Guide to Getting Started with Cloud Computing*. 2010. Sun white paper. Disponível em https://www.sun.com/offers/docs/cloud_computing_primer.pdf. Acesso em 10/10/2010.

TANENBAUM, A. S. *Modern Operating Systems*. 2001. 2nd edition. Prentice Hall.

TANENBAUM, M. V. S. A. S. *Distributed Systems. Principles and Paradigms*. 2006. 2nd edition. Pearson Prentice Hall.

WEISHAUPL, T. *et al. Business in the grid: The big project*. *Grid Economics and Business Models*, 2005.