

ANO
2018

ANDERSON SCHWEDE RAUGUST | ALOCAÇÃO DE INFRAESTRUTURAS VIRTUAIS
CONFIÁVEIS EM MÚLTIPLOS PROVEDORES IAAS



UDESC

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS GRADUAÇÃO EM COMPUTAÇÃO APLICADA

O paradigma da computação em nuvem consolidou o provisionamento de recursos virtuais sob demanda. Contudo, a diversidade de serviços, preços, Data Centers (DCs) e localizações geográficas tornou o ambiente de nuvem complexo e heterogêneo. Há muitos provedores de Infraestrutura como Serviço (IaaS), com diferentes custos de provisionamento e Qualidade de Serviço, dificultando do ponto de vista do cliente e escolha do provedor mais adequado às suas necessidades.

Sobretudo, a ocorrência de falhas em DCs de nuvens IaaS pode ocasionar a indisponibilidade das aplicações hospedadas. Neste contexto, esse trabalho apresenta um corretor de nuvem que visa melhorar dois aspectos: confiabilidade e capacidade de sobrevivência de uma Infraestrutura Virtual considerando o uso de réplicas e o espalhamento dos recursos virtuais sobre provedores, regiões e zonas.

Orientador: Guilherme Piegas Koslovski

Joinville, 2018

DISSERTAÇÃO DE MESTRADO

ALOCAÇÃO DE INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS EM MÚLTIPLOS PROVEDORES IAAS

ANDERSON SCHWEDE RAUGUST

JOINVILLE, 2018

UNIVERSIDADE DO ESTADO DE SANTA CATARINA - UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS - CCT
MESTRADO EM COMPUTAÇÃO APLICADA

ANDERSON SCHWEDE RAUGUST

ALOCÇÃO DE INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS EM
MÚLTIPLOS PROVEDORES IAAS

JOINVILLE

2018

ANDERSON SCHWEDE RAUGUST

**ALOCÇÃO DE INFRAESTRUTURAS VIRTUAIS CONFIÁVEIS EM
MÚLTIPLOS PROVEDORES IAAS**

Dissertação submetida ao Programa de Pós-Graduação em Computação Aplicada do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, para a obtenção do grau de Mestre em Computação Aplicada.

Orientador: Dr. Guilherme Piegas Koslovski

JOINVILLE

2018

Schwede Raugust, Anderson
Alocação de Infraestruturas Virtuais Confiáveis
em Múltiplos Provedores IaaS / Anderson Schwede
Raugust. - Joinville , 2018.
63 p.

Orientador: Guilherme Piegas Koslovski
Dissertação (Mestrado) - Universidade do Estado de
Santa Catarina, Centro de Ciências Tecnológicas,
Programa de Pós-Graduação em Computação Aplicada,
Joinville, 2018.

1. IaaS. 2. Infraestruturas Virtuais. 3.
Alocação. 4. Disponibilidade. 5. Confiabilidade. I.
Piegas Koslovski, Guilherme. II. Universidade do
Estado de Santa Catarina. Programa de Pós-Graduação.
III. Título.

Alocação de Infraestruturas Virtuais Confiáveis em Múltiplos Provedores IaaS

por

Anderson Schwede Raugust


Esta dissertação foi julgada adequada para obtenção do título de

Mestre em Computação Aplicada

Área de concentração em “Ciência da Computação”,
e aprovada em sua forma final pelo

**CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA
DO CENTRO DE CIÊNCIAS TECNOLÓGICAS DA
UNIVERSIDADE DO ESTADO DE SANTA CATARINA.**

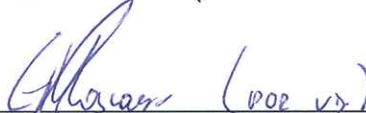
Banca Examinadora:



Prof. Dr. Guilherme Piêgas Koslovski
CCT/UDESC (Orientador/Presidente)



Prof. Dr. Mauricio Aronne Pillon
CCT/UDESC



Prof. Dr. João Vicente Ferreira Lima
UFSM

Joinville, SC, 25 de junho de 2018.

Dedico este trabalho aos meus familiares, amigos, colegas e professores que me acompanharam e me deram forças nessa magnífica trajetória.

AGRADECIMENTOS

A Deus, pela possibilidade. Por me acompanhar durante a jornada. Pelas pessoas que colocou cuidadosamente nesse caminho. Pela vida.

A família, pelo apoio de sempre. Seja material quando possível, com uma palavra ou com uma oração.

A minha namorada, Emanuele, pela paciência e companheirismo durante toda a caminhada. Por sempre ter a palavra certa. Por não poupar esforços para estar presente.

Aos professores do PPGCA e aos colegas do LabP2D, em especial professores Maurício e Charles pela imensa contribuição no desenvolvimento do trabalho e no meu desenvolvimento como pesquisador, nunca economizando sugestões e críticas.

Por fim mas não menos importante, ao orientador que às vezes assumiu papel de amigo, às vezes papel de irmão, às vezes papel de pai. Agradeço ao professor Guilherme pela oportunidade, pela paciência, pelo comprometimento, pela cobrança necessária e pelo enorme conhecimento transmitido. Com toda a certeza é um exemplo de profissional e ser humano que levarei comigo durante toda a vida.

“E não vos conformeis com este mundo, mas transformai-vos pela renovação do vosso entendimento, para que experimenteis qual seja a boa, agradável e perfeita vontade de Deus.”

(Bíblia Sagrada, Romanos 12:2)

RESUMO

O paradigma da computação em nuvem consolidou o provisionamento de recursos virtuais sob demanda. Contudo, a diversidade de serviços, preços, *Data Centers* (DCs) e localizações geográficas tornou o ambiente de nuvem complexo e heterogêneo. Há muitos provedores de Infraestrutura como Serviço (IaaS), com diferentes custos de provisionamento e Qualidade de Serviço (QoS), dificultando do ponto de vista do cliente a escolha do provedor mais adequado às suas necessidades. Sobretudo, a ocorrência de falhas em DCs de nuvens IaaS pode ocasionar a indisponibilidade temporária, parcial ou total, das aplicações hospedadas. Embora acordos de nível de serviço possam ser estabelecidos entre clientes e provedores, as aplicações ainda estão suscetíveis às eventuais falhas do DC. Neste contexto, esse trabalho apresenta um corretor de nuvem que visa melhorar dois aspectos: confiabilidade e capacidade de sobrevivência (*survivability*) de uma Infraestrutura Virtual (VI) considerando o uso de réplicas e o espalhamento dos recursos virtuais sobre provedores, regiões e zonas. Para atingir a confiabilidade especificada pelo cliente, utilizam-se réplicas, e para minimizar a probabilidade de falhas críticas, o espalhamento. A alocação de VI, gerenciado pelo corretor de nuvem proposto, considera os requisitos de confiabilidade e capacidade de sobrevivência como um modelo de Programação Linear Inteira Mista (MIP). Algumas restrições do MIP são relaxadas para obter um Programa Linear (LP) e utilizadas como entrada para a execução da heurística de recozimento simulado (*Simulated Annealing*). Os resultados demonstram que o corretor proposto incrementa a capacidade de sobrevivência e a confiabilidade de VIs sem um aumento significativo de custos. Sobretudo, a execução da heurística indicou a aplicabilidade da solução quando analisados os provedores IaaS existentes.

Palavras-chaves: IaaS, Infraestruturas Virtuais, Confiabilidade, Alocação, Disponibilidade e Capacidade de Sobrevivência.

ABSTRACT

The cloud computing paradigm consolidated the on-demand provisioning of virtual resources. However, the diversity of services, prices, data centers, and geographical footprints, have turned the clouds into a complex and heterogeneous environment. There are several Infrastructure as a Service (IaaS) providers differentiated by the provisioning costs and availability figures. Due to management complexity, the survivability and reliability aspects are often disregarded by tenants, eventually resulting in heavy losses due to unavailability of services that are hosted on Virtual Infrastructures (VIs). We present an alternative to improve VIs survivability and reliability, which considers the use of replicas and the spreading of virtual resources atop providers, regions, and zones. Replicas are used to achieve a user-defined reliability level while the controlled spreading of VI components decrease the probability of full outages. In addition, our proposal performs a cost-effective allocation. We formulate the VI allocation with survivability and reliability requirements as a Mixed Integer Program (MIP). Following, binary constraints are relaxed to obtain a Linear Program (LP). The LP solution is given as input for the simulated annealing technique. Simulation results with different reliability requests indicate an increasing in survivability without inflating costs.

Key-words: IaaS, Virtual Infrastructures, Reliability, Allocation, Availability and Survivability.

LISTA DE ILUSTRAÇÕES

Figura 1 – Infraestrutura Virtual	23
Figura 2 – Requisição de VI com confiabilidade alvo $c = 99,995\%$	35
Figura 3 – Número de réplicas para suportar $c = 99,995\%$ variando p	38
Figura 4 – Alocação de VI com confiabilidade alvo $c = 99,995\%$ e 3 grupos de falha (provedores, regiões e zonas). Quanto mais clara a cor, menor o custo.	40
Figura 5 – Fluxograma de execução das etapas	45
Figura 6 – Resultados da alocação com MIP.	53
Figura 7 – Resultados da alocação com MIP e RS-VIA.	55
Figura 8 – Resultados da alocação com RS-VIA.	56

LISTA DE TABELAS

Tabela 1 – Principais contribuições dos trabalhos relacionados.	30
Tabela 2 – Notação para representar requisições de VIs, provedores IaaS, e detalhes fundamentais do modelo.	36
Tabela 3 – Publicações diretas.	59
Tabela 4 – Publicações em colaboração.	59

LISTA DE ABREVIATURAS E SIGLAS

CC	<i>Cloud Computing</i>
DC	<i>Data Center</i>
EC2	<i>Elastic Cloud Computing</i>
HA	<i>High-Availability</i>
IaaS	<i>Infrastructure-as-a-Service</i>
LP	<i>Linear Programming</i>
MIP	<i>Mixed Integer Programming</i>
MTBF	<i>Mean Time Between Failures</i>
MTBO	<i>Mean Time Between Outages</i>
MTTF	<i>Mean Time To Fail</i>
MTTR	<i>Mean Time To Repair</i>
NIST	<i>National Institute of Standards and Technology</i>
ORP	<i>Opportunistic Redundancy Pooling</i>
PaaS	<i>Platform-as-a-Service</i>
QoS	<i>Quality of Service</i>
RS-VIA	<i>Reliable and Survivable Virtual Infrastructure Allocation</i>
SA	<i>Simulated Annealing</i>
SaaS	<i>Software-as-a-Service</i>
SLA	<i>Service Level Agreement</i>
S3	<i>Simple Storage Service</i>
VI	<i>Virtual Infrastructure</i>
VL	<i>Virtual Link</i>
VM	<i>Virtual Machine</i>
VN	<i>Virtual Network</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	OBJETIVOS	17
1.2	METODOLOGIA E CONTRIBUIÇÕES	17
1.3	ORGANIZAÇÃO DO TEXTO	18
2	REVISÃO DA LITERATURA	20
2.1	COMPUTAÇÃO EM NUVEM	20
2.1.1	Nuvens IaaS	21
2.1.2	Infraestrutura Virtual	22
2.1.3	Provisionamento de VIs	24
2.2	FALHAS EM NUVENS IAAS E VIS	25
2.2.1	Conceitos e Definições	25
2.2.1.1	Cadeia de Falhas	26
2.2.1.2	Tipos de Falhas	26
2.2.2	Métricas	26
2.2.2.1	Disponibilidade	27
2.2.2.2	Confiabilidade	28
2.2.2.3	Capacidade de Sobrevivência	28
2.2.3	Replicação	28
2.3	DEFINIÇÃO DO PROBLEMA	29
2.3.1	Trabalhos Relacionados	30
2.3.1.1	Alocação de Recursos Físicos para Hospedar VI	31
2.3.1.2	Técnicas para Provisionamento de VIs com Capacidade de Sobrevivência	32
2.4	CONSIDERAÇÕES PARCIAIS	32
3	MODELO EXATO PARA ALOCAÇÃO CONFIÁVEL DE VIS EM MÚLTIPLOS PROVEDORES IAAS	34
3.1	FORMULAÇÃO DO PROBLEMA	34
3.1.1	Requisições de VIs e Provedores IaaS	34
3.1.2	Probabilidade de Falha	35
3.1.3	Definição de Réplicas para VMs Críticas	37
3.1.4	Alocação de Provedores IaaS para Hospedar VIs	38
3.2	MODELO EXATO PARA ALOCAÇÃO DE VIS EM MÚLTIPLOS PROVEDORES IAAS	39

3.2.1	Variáveis e Objetivos	39
3.2.1.1	Custo de Alocação da VI	41
3.2.1.2	Impacto de Falhas em Provedores IaaS	42
3.2.1.3	Função Objetivo	43
3.2.2	Restrições para Alocação	43
3.2.2.1	Restrições de VMs	43
3.2.2.2	Restrições de Enlaces	44
3.3	<i>RELIABLE AND SURVIVABLE VIRTUAL INFRASTRUCTURE ALLO-</i> <i>CATION (RS-VIA)</i>	44
3.3.1	RELAXANDO AS VARIÁVEIS E ADICIONANDO RESTRIÇÕES	45
3.3.2	HEURÍSTICA DE DECISÃO	46
3.4	CONSIDERAÇÕES PARCIAIS	47
4	SIMULAÇÃO E DISCUSSÃO DOS RESULTADOS	49
4.1	MÉTRICAS	49
4.2	PARÂMETROS	50
4.3	DEFINIÇÃO DE CENÁRIOS DE SIMULAÇÃO	50
4.3.1	MIP	51
4.3.2	Alocação Exata e RS-VIA	51
4.3.3	RS-VIA em Larga Escala	51
4.4	RESULTADOS	52
4.4.1	Execução do MIP	52
4.4.2	MIP Exato com RS-VIA	54
4.4.3	RS-VIA	54
4.5	CONSIDERAÇÕES PARCIAIS	55
5	CONCLUSÃO	58
5.1	TRABALHOS FUTUROS	59
5.2	PUBLICAÇÕES	59
5.2.1	Diretas	59
5.2.2	Trabalhos em Colaboração	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

A contratação de provedores de Infraestrutura como Serviço (IaaS) se tornou popular entre diversas empresas (ROSENBERG; MATEOS, 2010; STALLINGS, 2015). Sobretudo, provedores de nuvens IaaS oferecem uma estrutura robusta e precificam seguindo o modelo *pay-as-you-go*, no qual o usuário é tarifado pelos recursos que compõem a Infraestrutura Virtual (*Virtual Infrastructure* – VI) reservada (MELL; GRANCE, 2011). Em resumo, VIs são um conjunto de recursos computacionais virtualizados (*e.g.*, máquinas virtuais, contêineres) e redes virtuais (*e.g.*, roteadores, *switches*) (SOTOMAYOR *et al.*, 2009).

Atualmente, associado ao modelo IaaS, existem serviços que são hospedados e oferecidos em VIs e, portanto, suscetíveis a eventuais indisponibilidades dos provedores. Mesmo que os provedores cumpram rígidas medidas administrativas nos seus *Data Centers* (DCs) para que a VI oferecida seja confiável, elas podem não ser suficientes para sistemas que necessitam estar ininterruptamente operacionais (NOORMOHAMMADPOUR; RAGHAVENDRA, 2017). Os índices de disponibilidade fornecidos por provedores são divulgados e estabelecidos em contratos, através do Acordo de Nível de Serviço (*Service Level Agreement* – SLA) (*e.g.*, 99,95%).

Porém, no caso de aplicações críticas, o grau de disponibilidade fornecido nem sempre atinge o nível desejado, podendo causar prejuízos financeiros. Por exemplo, a indisponibilidade de 20 horas do provedor Amazon EC2 causou transtorno a milhões de usuários dos serviços da *Netflix*, *Instagram* e *Pinterest* (AVRAM, 2011). Recentemente, os serviços do *GitHub*, *Trello*, *Giphy*, *Medium* e *Slack* foram afetados pelo mesmo problema e ficaram 4 horas inacessíveis (AMAZON EC2, 2017). Tal indisponibilidade causou um prejuízo estimado em 150 milhões de dólares (HERSHER, 2017). Nestes casos, a política normalmente aplicada é a recompensa das horas de indisponibilidade em créditos para locação e reinício das VIs. Para serviços críticos, além do alto grau de disponibilidade, ainda é necessário a certeza de que os dados processados estão corretos. Entretanto, de modo geral o SLA considera somente a medida do tempo disponível.

Neste tipo de serviço, a confiabilidade e a capacidade de sobrevivência (do inglês *survivability*) são características essenciais pois as VIs são dependentes de DCs e quedas não planejadas são comuns nestas infraestruturas (GOVINDAN *et al.*, 2016; NOORMOHAMMADPOUR; RAGHAVENDRA, 2017). Por um lado, a aplicação crítica precisa de confiabilidade, a qual leva em consideração a probabilidade de uma VI estar operando corretamente. Por outro, precisa da capacidade de sobrevivência,

que indica a habilidade da VI permanecer operacional na ocorrência de uma falha no DC hospedeiro. Assim, os serviços disponíveis em um provedor classificado como de Alta Disponibilidade (*High Availability* – HA), além de terem um custo elevado, não tornam uma VI necessariamente confiável. Isso se deve aos diferentes tipos de falhas que podem ocorrer no DC, principalmente nos planos de controle e de gerenciamento dos recursos (GOVINDAN *et al.*, 2016).

O nível de confiabilidade de uma VI pode ser melhorado através da utilização de réplicas (YEOW; WESTPHAL; KOZAT, 2011). A réplica consiste na disponibilização excedente de recursos para serviços que já encontram-se em operação. Uma réplica passiva não substitui o serviço existente, mas fica pronta para entrar em operação em caso de falha ou lentidão do serviço principal. Por sua vez, uma réplica ativa recebe requisições e atua diretamente no processamento de respostas (SILVA; OBELHEIRO; KOSLOVSKI, 2017).

Dispor de réplicas sempre implica em aumento de custo, o que nem sempre é compatível com o orçamento dos clientes. No entanto, o cliente tem conhecimento da aplicação e é capaz de quantificar quais são os seus serviços críticos. Buscando conciliar o aumento de confiabilidade do serviço e a minimização do custo, pode-se restringir a associação de réplicas a apenas instâncias críticas (KOSLOVSKI *et al.*, 2010). Assim, tornando os serviços críticos mais confiáveis, obtém-se um aumento no grau de confiabilidade da aplicação reduzindo o impacto financeiro das réplicas.

Contudo, replicando cada máquina virtual (*Virtual Machine* – VM) de uma VI dobraria, pelo menos, o custo de provisionamento, o que é impraticável para a maioria dos clientes. Consequentemente, uma indisponibilidade temporária das VMs não críticas é tolerável. Assim, o serviço passa a ser resiliente à falha de instâncias críticas, mas permanece vulnerável à indisponibilidade de provedores.

Dentre as tarefas administrativas essenciais para o provisionamento de VIs (JUNIOR; MIERS; KOSLOVSKI, 2017; ANHALT; KOSLOVSKI; PRIMET, 2010), o presente trabalho foca na alocação e seleção de provedores. A alocação de uma VI designa o processo de alocação de recursos físicos para hospedar os recursos virtuais que compõe a mesma. Esse mapeamento e alocação de recursos configura um problema NP-Difícil, diante do elevado volume de combinações possíveis entre hospedeiros e recursos virtuais (CHOWDHURY; BOUTABA, 2010). A VI pode ser alocada em recursos de um ou múltiplos DCs.

A importância de alocar uma VI em múltiplos provedores vem das diferentes necessidades de clientes de nuvem. Um exemplo é a localização do DC. O cliente pode querer reduzir a latência de seus usuários alocando sua VI em um provedor que possua um DC mais próximo deles. Agora, se esse cliente receber uma grande de-

manda de usuários de uma nova região, pode ser conveniente alocar parte de sua VI em um novo provedor, que possua um DC mais próximo da região da nova demanda. Outro fator importante é diretamente relacionado à capacidade de sobrevivência, pois considerar o uso de múltiplos provedores aumenta a possibilidade de um maior espalhamento da VI. Quanto mais domínios de falha diferentes usados para alocar a VI, menor a probabilidade de uma falha total (BODÍK *et al.*, 2012; CAVALCANTI; OBE-LHEIRO; KOSLOVSKI, 2014).

Considerando o exposto até aqui, fica evidente que há um problema de instabilidade nos provedores de nuvem (NOORMOHAMMADPOUR; RAGHAVENDRA, 2017; GOVINDAN *et al.*, 2016). Os clientes não podem confiar apenas no serviço básico de um provedor IaaS, enquanto os serviços que garantem uma maior confiabilidade são caros e inviáveis para empresas com menor poderio financeiro. Portanto, clientes carecem de uma solução que aumente a confiabilidade e capacidade de sobrevivência de sua VI sem aumentar drasticamente o custo de alocação.

1.1 OBJETIVOS

O objetivo geral do trabalho é a definição de uma solução para alocação de uma infraestrutura virtual confiável em múltiplos provedores IaaS. Para tanto, os objetivos específicos para alcançar o objetivo geral são:

- Definir o número de réplicas de acordo com a probabilidade de falhas de provedores e da especificação de requisitos do usuário;
- Modelar a confiabilidade de uma VI em múltiplos provedores;
- Identificar o custo mínimo para obter o nível de confiabilidade especificado;
- Modelar os componentes como programação inteira mista;
- Definir uma heurística de aproximação escalável;
- Analisar a aplicabilidade da solução com simulações controladas.

1.2 METODOLOGIA E CONTRIBUIÇÕES

Inicialmente, a revisão da literatura foi realizada para definir os conceitos e explorar o problema. Após, foi realizada uma pesquisa de trabalhos relacionados na literatura especializada, utilizando a metodologia de revisão narrativa. Esta metodologia seleciona trabalhos que tratam o mesmo domínio e que possuem um objetivo relacionado ao objetivo do presente trabalho. Como resultado da revisão foi constatado que o tema possui uma grande lacuna de pesquisa.

Portanto, foi proposto um sistema autônomo que atua como corretor de serviço na nuvem (MELL; GRANCE, 2011). Através dos dados coletados dos provedores de nuvem pública, ele atua de forma independente para determinar a alocação que melhor atenda aos objetivos do cliente quanto à confiabilidade, capacidade de sobrevivência e custo.

Por se tratar de um problema de otimização, um modelo de Programação Inteira Mista (Mixed Integer Programming – MIP) foi implementado (HILLIER; LIEBERMAN, 2013). Como resolver um MIP em larga escala é computacionalmente intratável, uma heurística de aproximação baseada em Recozimento Simulado (*Simulated Annealing* – SA) também foi implementada (GLOVER; KOCHENBERGER, 2006).

Os resultados indicam que o MIP pode encontrar uma alocação intermediária que é benéfica para o cliente, provendo confiabilidade e capacidade de sobrevivência sem aumentar drasticamente o custo de alocação. A aplicação da heurística atingiu resultados próximos dos ótimos nos cenários comparados, mostrando-se uma boa alternativa para aplicação em cenários onde o MIP não pode ser aplicado devido ao limitante de processamento.

O presente trabalho destaca três principais contribuições:

- A formulação da alocação de VI em múltiplos provedores como um MIP, considerando capacidade de sobrevivência, confiabilidade e custo. Ainda, o MIP considera VMs críticas e regulares, além de requisitos de comunicação entre VMs;
- As restrições do MIP são relaxadas para obter um modelo de Programação Linear (Linear Programming – LP), no qual a técnica SA é aplicada para encontrar uma solução aceitável em tempo tratável computacionalmente;
- Os resultados das simulações baseados em dados de provedores de nuvem pública são analisados, demonstrando a aplicabilidade e eficiência da proposta.

1.3 ORGANIZAÇÃO DO TEXTO

O Capítulo 1 apresentou a motivação, o problema, os objetivos e perspectivas do presente trabalho. Na sequência, o Capítulo 2 apresenta a revisão da literatura, explorando os conceitos utilizados, especificando o problema e apresentando os resultados da busca por trabalhos relacionados na literatura especializada. Já o Capítulo 3 apresenta a definição formal do problema e detalha a formulação do MIP proposto, com objetivos e restrições. Também, discute a heurística de aproximação baseada em SA, denominada RS-VIA. Os resultados das simulações com o MIP e a heurística com dados reais são apresentados e discutidos no Capítulo 4. Por fim, o Capítulo 5 apre-

sentam as considerações finais sobre o trabalho e as perspectivas e sugestões para trabalhos futuros.

2 REVISÃO DA LITERATURA

O presente capítulo apresenta uma discussão sobre conceitos-chaves para o entendimento do trabalho. A Seção 2.1 discute os conceitos de nuvens IaaS, incluindo a sua classificação e os atores envolvidos. Também apresenta definições e exemplos de VIs, abordando o processo e etapas para o seu provisionamento. A Seção 2.2 aborda falhas em provedores IaaS, ressaltando as definições de falhas, réplicas, métricas para mensurar a tolerância a falha de uma VI em um provedor e exemplos reais. Na Seção 2.3 o leitor encontra a definição do problema explorado, ressaltando a lacuna de pesquisa e posicionando perante os trabalhos relacionados. Por fim, a Seção 2.4 apresenta uma revisão do capítulo, destacando os principais pontos a serem considerados no restante da leitura. A partir do entendimento do presente capítulo, a discussão de um modelo exato para alocação de VIs ocorre no Capítulo 3.

2.1 COMPUTAÇÃO EM NUVEM

O conceito de nuvem usualmente aceito pela comunidade acadêmica é proveniente do *National Institute of Standards and Technology* (NIST), que define como um modelo para permitir um acesso ubíquo, conveniente e sob demanda a um conjunto compartilhado de recursos computacionais configuráveis (*e.g.*, redes, servidores, armazenamento, aplicações, e serviços). Tais recursos podem ser rapidamente provisionados e liberados com um esforço mínimo de gerenciamento ou interação com o provedor de serviço (MELL; GRANCE, 2011).

Segundo Mell e Grance (2011), existem cinco características essenciais providas pela computação em nuvem: auto-atendimento sob demanda, permitindo ao consumidor provisionar capacidades computacionais automaticamente conforme a necessidade, sem interação humana; amplo acesso a rede, disponibilizando serviços pela Internet para diferentes dispositivos (*e.g.*, celulares, computadores, *workstations*); permitir que recursos dos provedores sejam agrupados para servir diferentes consumidores, fazendo com que o cliente dificilmente saiba a localização exata dos recursos que está utilizando, mas ainda autorizando que ele especifique o local em alto nível (país, estado, etc); elasticidade rápida, alocando e liberando recursos elasticamente para ajustar à demanda do serviço, de modo que o cliente tenha a impressão que os recursos são ilimitados e que podem se ajustar a qualquer demanda; por fim, a computação em nuvem provê serviços mensuráveis, proporcionando transparência e a oportunidade do provedor otimizar a utilização de seu DC. Apenas com a combinação de todas as características apresentadas é possível garantir plenamente a computa-

ção em nuvem.

Além das características essenciais, a computação em nuvem apresenta quatro modelos de implementação (MELL; GRANCE, 2011). A Nuvem Privada é provisionada para o uso exclusivo de uma organização. Já a Nuvem Comunitária é compartilhada entre diferentes organizações, geralmente com objetivos em comum. A Nuvem Pública é o modelo mais difundido, destinado ao público em geral. Tal modelo é provido por provedores como Amazon, Google, Rackspace, entre outros. Combinando dois ou mais dos modelos apresentados caracteriza-se uma Nuvem Híbrida. Cabe ressaltar que embora o presente trabalho utilize apenas dados de nuvens públicas em exemplos e simulações (conforme detalhado no Capítulo 4) ele pode ser aplicado em qualquer modelo de implementação.

A computação em nuvem pode ser provida com diferentes modelos de serviço. *Software* como um serviço (*Software-as-a-Service* – SaaS) oferece ao consumidor o acesso às aplicações que estejam executando em uma infraestrutura de nuvem, sem que ele gerencie essa infraestrutura. Plataforma como um serviço (*Platform-as-a-Service* – PaaS) é um modelo que permite ao consumidor implantar suas aplicações sobre uma infraestrutura de nuvem, sem a possibilidade de gerenciar essa infraestrutura. A aplicação precisa ser criada com componentes (linguagem de programação, bibliotecas, etc) suportados pelo provedor. Alguns exemplos de serviços PaaS são IBM Bluemix, Windows Azure e Jelastic. Além desses, encontra-se a Infraestrutura como um serviço (*Infrastructure-as-a-Service* – IaaS), foco do presente trabalho. Segundo Miers e Koslovski (2014) e Mell e Grance (2011), é o modelo mais básico de serviços que podem ser oferecidos pela nuvem. Como o nome sugere, o modelo de infraestrutura como um serviço consiste em provisionar processamento, armazenamento, rede, e todos os recursos computacionais básicos necessários.

2.1.1 Nuvens IaaS

Os provedores IaaS tem como base um conjunto de DCs. Os DCs são compostos por equipamentos (servidores, roteadores, switches e enlaces) gerenciados por uma instituição, organizados em uma arquitetura definida pela mesma (BENSON; AKELLA; MALTZ, 2010). Com a popularização do modelo IaaS, muitos clientes optam por terceirizar seu DC. Essa terceirização evita o alto custo de montagem e de manutenção, passando todas as preocupações e gastos de infraestrutura para a contratada. Assim, o cliente tem a total liberdade para focar no objetivo do seu negócio.

Para a compreensão do funcionamento de uma nuvem IaaS, é necessário entender quem são os atores e como eles interagem entre si. Em Unbehagen *et al.* (2015), Miers e Koslovski (2014) e Mell e Grance (2011) são definidos os cinco atores principais:

- **Cliente/Consumidor:** Quem utiliza o serviço dos Fornecedores/Provedores;
- **Fornecedor/Provedor:** Responsável por entregar o serviço às partes interessadas;
- **Auditor:** Avalia, independentemente, a qualidade dos serviços oferecidos, utilizando informações que indiquem o desempenho e a segurança do mesmo;
- **Corretor:** Gerencia o uso dos serviços do Fornecedor, conduzindo e ajustando a negociação entre o Cliente e o Fornecedor;
- **Portador:** Permite a conexão entre o Cliente e o Fornecedor, permitindo assim que o acesso do Cliente ao serviço contratado.

Entre os atores definidos, destacam-se as partes principais do processo que são o cliente e o fornecedor. O fornecedor visa o lucro de seu negócio, enquanto o cliente quer a qualidade do serviço que está contratando, como baixa latência, proximidade geográfica dos usuários finais e alta disponibilidade. Geralmente o cliente tem conhecimento do seu negócio, sabendo o que é necessário para mantê-lo operante. Muitas vezes falta ao cliente o conhecimento técnico necessário para conseguir gerenciar a sua VI da melhor forma sobre o provedor, Isso se dá tanto pela dificuldade do processo quanto pela abstração de dados de falhas do provedor, que acabam permanecendo em sigilo. Assim, surge uma lacuna na necessidade de um intermediador nessa relação.

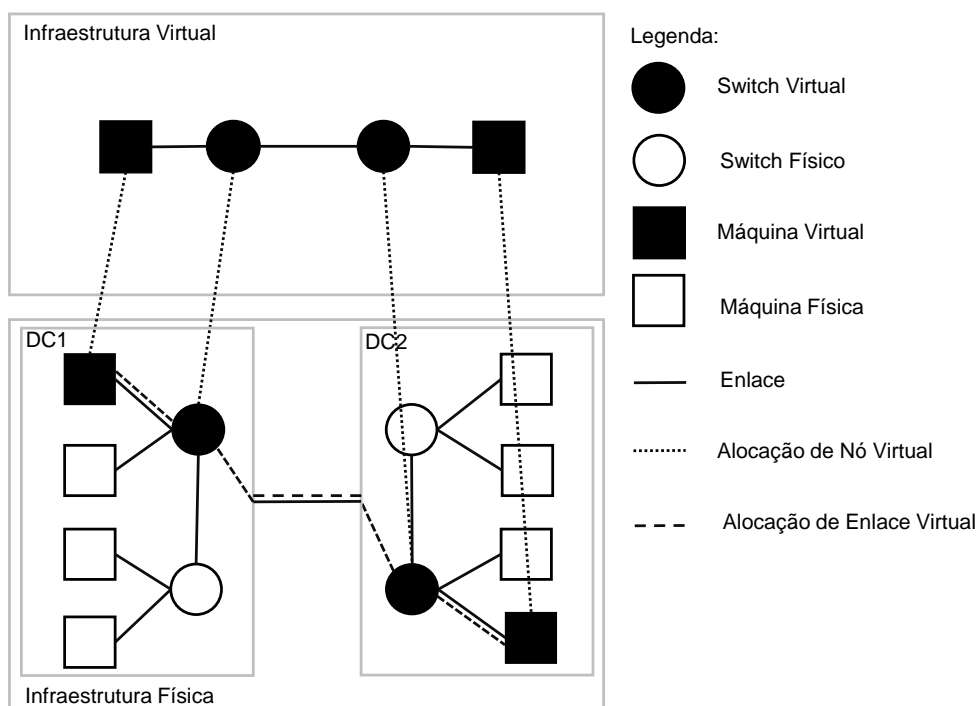
O presente trabalho se posiciona no papel do corretor. Nele pode ocorrer o gerenciamento da VI do cliente utilizando dados dos provedores para mediar a relação entre tais atores. Assim é possível atender às necessidades do cliente, preenchendo a lacuna de conhecimento técnico relacionada com confiabilidade e custo.

2.1.2 Infraestrutura Virtual

Utilizando uma nuvem IaaS, o cliente deixa de se preocupar com uma infraestrutura física para acessar uma Infraestrutura Virtual (VI). Uma VI é a combinação de redes virtuais com recursos computacionais virtualizados (e.g., máquinas virtuais, container), podendo ser definida como uma união e evolução de tais conceitos (SOTOMAYOR *et al.*, 2009). Com tal evolução, a infraestrutura física é totalmente abstraída. Assim, o serviço virtual é reservado e provisionado dinamicamente, conforme a necessidade do cliente. Especificamente, redes virtuais (*Virtual Network* – VN) são um conjunto de recursos virtuais mapeados sobre a mesma infraestrutura física, permitindo que múltiplas redes virtuais personalizadas coabitem os mesmos recursos físicos (YU *et al.*, 2008). São compostas de nós virtuais, como roteadores e *switches*, interligados por enlaces virtuais (*Virtual Link* – VL).

A Figura 1 ilustra a composição de uma VI. Nela, os nós na cor preta indicam a VI alocada na infraestrutura física, com seus respectivos enlaces virtuais. A infraestrutura física, dividida na Figura 1 em 2 DCs, pode pertencer a dois provedores diferentes, que gerenciam a nuvem computacional e comercializam os serviços IaaS. A pioneira na comercialização desse tipo de serviço foi a Amazon, que desde 2006 oferece produtos como o *Elastic Cloud Computing* (EC2), que disponibiliza diferentes tipos de VMs para a composição de uma VI, *Simple Storage Service* (S3), para armazenamento simples de grandes volumes de dados, entre outros (AMAZON AWS, 2018). Similar aos serviços citados da Amazon, o Google apresenta o *Google Compute Engine* e *Google Cloud Storage*, disponibilizando os mesmos serviços de EC2 e S3, respectivamente (GOOGLE CLOUD PLATAFORM, 2018). A Microsoft também investiu nos serviços, agregando vários produtos ao Azure onde o cliente pode solicitar máquinas virtuais, armazenamento, redes virtuais e mais uma gama de serviços (MICROSOFT AZURE, 2018). Também na corrida, o Rackspace oferece o *Managed Hosting*, comercializando a utilização de seus recursos para a hospedagem de VIs (RACKSPACE, 2018). Além desses, segundo o serviço de monitoramento CloudHarmony¹, existem pelo menos quarenta provedores que oferecem serviços baseados em IaaS.

Figura 1 – Infraestrutura Virtual



Fonte: Elaboração própria, 2018.

Na Figura 1, a VI é especificada no quadro superior. Ela é composta de VMs ou contêineres, representados pelos quadrados preenchidos, e os *switches* vir-

¹ cloudharmony.com – acesso em 21/01/2018

tuais (PFAFF *et al.*, 2015), representados pelos círculos preenchidos. Fazendo a comunicação entre esses componentes, as arestas representam os enlaces virtuais. No quadro inferior, a infraestrutura física é representada. Ela é dividida em dois DCs, que representam zonas, regiões ou provedores diferentes. Nela as máquinas físicas e os *switches* físicos são retratados pelos quadrados vazios e círculos vazios, respectivamente, enquanto os enlaces físicos são representados pelas arestas do grafo. No exemplo da Figura 1, cada recurso virtual está mapeado para um recurso físico diferente com o propósito de facilitar a leitura. Na prática, os recursos físicos costumam hospedar múltiplos recursos virtuais, podendo estes fazer parte da mesma VI ou não.

Desta forma, a utilização de VIs possibilita a interconexão entre diferentes provedores, DCs e usuários. Pode-se concluir, portanto, que as VIs são um meio de se prover a computação em nuvem quando se trata de SaaS e PaaS, e o produto final quando se trata de IaaS (SCHOO *et al.*, 2010). Atualmente, os serviços e aplicações mais populares utilizam uma solução IaaS, seja pública, privada, comunitária ou híbrida. O principal motivo é a escalabilidade necessária ao grande volume de dados que recebem diariamente suas VIs. Entre os que utilizam nuvens públicas encontram-se Netflix, Instagram, Pinterest, GitHub, Trello, Giphy, Medium, Slack, Airbnb, Spotify e Nubank hospedados na Amazon EC2 (AVRAM, 2011; AMAZON EC2, 2017).

2.1.3 Provisionamento de VIs

O provisionamento de VIs compreende uma série de etapas a serem executadas, formando assim o chamado ciclo de vida da VI (ANHALT; KOSLOVSKI; PRIMET, 2010). As principais tarefas são especificação de requisição, alocação, configuração, execução, monitoração e término, na qual ocorre a liberação dos recursos virtuais. Na especificação a VI é descrita, detalhando VMs, roteadores, switches, rede e os requisitos de Qualidade de Serviço (*Quality of Service* – QoS) que deverão ser acatados pelo provedor. Já na etapa de alocação, o provedor averigua se a infraestrutura física pode receber a VI respeitando a especificação da etapa anterior. Após, os equipamentos são preparados e configurados para receber a VI na etapa de configuração. A fase de instanciação gera e disponibiliza a VI para o usuário final, pronta para o uso. Estando em uso pelo usuário, a VI entra na fase de monitoração. Nessa fase o provedor pode, respeitando os requisitos de QoS e outros estabelecidos no SLA, realizar ajustes na alocação da VI para garantir a elasticidade, maximizar a eficiência do uso de sua infraestrutura física, além de outros algoritmos que podem ser implementados pelo provedor (JUNIOR; MIERS; KOSLOVSKI, 2017). Por fim, com o fim do contrato ou solicitação de encerramento pelo usuário final, a etapa de término libera os recursos físicos, removendo permanentemente a VI.

Dentre todas as etapas descritas, o presente trabalho atua dentro da etapa de

alocação. A alocação reúne dois níveis de informações a serem analisadas: a seleção de provedores e a seleção de recursos físicos. Primeiro, a seleção dos provedores, em mais alto nível, é realizada pelo corretor (*broker*) analisando informações públicas dos provedores e características da VI solicitada, como o custo da VM e de comunicação, distância do DC para os usuários finais e dados públicos de falhas (JUNIOR; MIERS; KOSLOVSKI, 2017; ZHANG; CHENG; BOUTABA, 2010). Após os provedores serem selecionados e receberem a requisição da VI a ser alocada, a alocação passa para a etapa de seleção de recursos físicos para hospedar os recursos virtuais. Nessa etapa o provedor utiliza algoritmos internos, mantidos em sigilo, que visam atender ao seu interesse na seleção dos recursos físicos (respeitando o SLA). Os objetivos dessa alocação podem ser financeiros, como minimizar a fragmentação do DC, de QoS, como menor latência ou maior vazão, ou de resiliência, diminuindo a probabilidade de falhas (FISCHER *et al.*, 2013).

2.2 FALHAS EM NUVENS IAAS E VIS

As falhas em provedores laaS são muito mais comuns e devastadoras do que parecem, tanto para o próprio provedor quanto para os clientes hospedados. Por exemplo, a indisponibilidade de 20 horas da Amazon EC2 causou transtorno a milhões de usuários dos serviços da *Netflix*, *Instagram* e *Pinterest* (AVRAM, 2011). Essa indisponibilidade foi causada pelo algoritmo interno de escalonamento que alterou uma rota de comunicação para outra sem capacidade de suportar o volume de dados necessário para o DC, resultando na inoperância total das VIs ali hospedadas. Recentemente, os serviços do *GitHub*, *Trello*, *Giphy*, *Medium* e *Slack* também sofreram com indisponibilidade, ficando 4 horas inacessíveis (AMAZON EC2, 2017). Segundo o relatório disponibilizado por AMAZON EC2 (2017), um erro em um parâmetro para a exclusão de servidores de um subsistema eliminou um número muito maior do que o planejado inicialmente, interrompendo os serviços hospedados nesse DC. A *startup* Cyence Inc. estima um prejuízo de 150 milhões de dólares (HERSHER, 2017).

2.2.1 Conceitos e Definições

Se tratando de um provedor laaS, o histórico de ocorrência de falhas em seus DCs é um importante aliado para reduzir a probabilidade de que uma falha ocorra em seu sistema (GOVINDAN *et al.*, 2016). Analisando-os, é possível concluir que alguns provedores oferecem um serviço mais robusto do que outros. Primeiro, é necessário entender como uma falha acontece e se propaga. Assim, a discussão da presente seção é conduzida dentro do contexto de nuvens laaS.

2.2.1.1 Cadeia de Falhas

Há normalmente uma sequência de eventos no sistema que resulta em falha, chamada de cadeia fundamental ou de ameaça (DANTAS, 2005). De acordo com Ucla *et al.* (2001), essa cadeia é composta de defeito, erro e por fim a falha. O defeito é caracterizado por ser uma condição do sistema que, devido a um processo computacional, condições do ambiente ou defeito externo, produz um erro. Esse erro gerado inicia então o processo de propagação do erro. Quando um erro é propagado até a interface de serviço, fazendo-o entregar um serviço incorreto, ele se torna uma falha. A falha de um sistema pode causar um defeito em outros sistemas que interagem com ele, seguindo o processo de propagação. Tais falhas podem ocorrer tanto nas VMs como nos recursos de rede, como *switch* e roteador.

Essa cadeia também afeta grandes DCs, como o do Google. Govindan *et al.* (2016) cita que atualmente falhas são comuns dada a grande escala e heterogeneidade dos componentes. Também, a rede está em constante alteração dado o aumento da demanda de tráfego e a adição de novos serviços. A combinação de tais fatores gera um alto risco de ocorrência da cadeia e propagação para diferentes componentes, sendo um desafio para o gerenciamento da nuvem.

2.2.1.2 Tipos de Falhas

As falhas são classificadas de acordo com a maneira que o sistema pode falhar. Segundo Avizienis *et al.* (2004), existem três categorias. As falhas por valor fazem com que o sistema forneça uma resposta incorreta. Falhas de temporização ou de desempenho fazem com que o sistema não responda no intervalo de tempo especificado. Essa categoria pode ser dividida em falhas de omissão, quando o sistema deixa de responder as requisições, e em falhas de parada (*crash*), em que o sistema deixa de operar ignorando as mensagens até o reestabelecimento da operação. Por fim, existem as falhas bizantinas ou arbitrárias, quando o sistema fornece respostas incorretas a conjuntos de requisições.

A presente proposta atua na redução do impacto de falhas de parada. Desse modo, atua com objetivo que a inoperância de um componente da VI não afete o seu funcionamento como um todo.

2.2.2 Métricas

Existem algumas medidas que ajudam a mensurar a qualidade do serviço fornecido por um provedor IaaS. O MTBF (*Mean Time Between Failures* ou tempo médio entre falhas, Equação 2.1) diz respeito a média de tempo decorrido entre uma falha e outra, sendo a razão entre o tempo total de serviço correto no período observado

pelo número de falhas ocorridas nesse período. Por outro lado, o MTTR (*Mean Time To Repair* ou tempo médio para recuperar, Equação 2.2) é dado como a razão entre o tempo indisponível no período observado pelo número de falhas ocorridas.

$$MTBF = \frac{\text{tempo disponível}}{\text{número de falhas}} \quad (2.1)$$

$$MTTR = \frac{\text{tempo indisponível}}{\text{número de falhas}} \quad (2.2)$$

Baseando-se no MTBF, o MTBO (*Mean Time Between Outages* ou tempo médio entre quedas) se refere às falhas de *hardware*, especificamente. Para isso, é preciso identificar a fração das falhas ocasionadas por problemas de *hardware* (FFAS). Essa fração é subtraída de 1 e, por fim, se torna o divisor do MTBF na Equação 2.3, resultando no MTBO.

$$MTBO = \frac{MTBF}{1 - FFAS} \quad (2.3)$$

Utilizando essas medidas, é possível chegar a valores que dão uma melhor noção do impacto das falhas nos sistemas. É o caso de disponibilidade e confiabilidade.

2.2.2.1 Disponibilidade

A disponibilidade do serviço ofertado por um provedor está diretamente ligada ao tempo em que ele está respondendo (*uptime*). Segundo Ucla *et al.* (2001), disponibilidade representa a probabilidade de um sistema apresentar o serviço correto em um determinado instante. Essa medida simples ajuda a ter uma visão geral da instabilidade dos serviços em um período de tempo observado. Assim, ela é expressa como a porcentagem de tempo em que esteve disponível em relação ao total de tempo do período analisado. Por exemplo, analisando um período de trinta dias (720 horas) em que um provedor teve uma indisponibilidade total de 5 horas temos, seguindo a Equação 2.4, que a disponibilidade é igual a 99,3%. Dantas (2005) indica que a disponibilidade também pode ser encontrada através da Equação 2.5.

$$\frac{720 - 5}{720} \times 100 \quad (2.4)$$

$$\frac{MTTF}{MTTF + MTTR} \quad (2.5)$$

Para exemplificar a aplicação com dados reais, para o mês de abril de 2017 o provedor *Amazon EC2*, na região *ap-northeast-2*, apresentou uma disponibilidade de 99,997% enquanto o provedor *ElasticHosts*, na região *ams-e* apresentou 99,809%².

2.2.2.2 Confiabilidade

A confiabilidade é uma medida robusta para mensurar a probabilidade da ocorrência de falhas em um determinado provedor, definida por Ucla *et al.* (2001) como a medida da entrega contínua do serviço correto, com relação ao tempo médio para ocorrer uma falha. Em outras palavras, ao contrário da disponibilidade que mensura apenas se o serviço está respondendo, a confiabilidade indica também se a resposta está correta. Para encontrar o valor são necessárias informações provenientes do provedor, como MTTF e MTTR, das quais o cliente não tem conhecimento. A confiabilidade é expressa como $1 - p$, sendo p a probabilidade de falha encontrada a partir do MTBF. Por fim, a medida é expressa como uma porcentagem (99,975%, por exemplo).

Seguindo o exemplo utilizado na Seção 2.2.2.1 com dados extraídos da plataforma *CloudHarmony*, a confiabilidade apresenta uma variação maior visto que o provedor *ElasticHosts* teve mais *outages* na região analisada. Assim, para esse a confiabilidade é de 97,495%, enquanto para a *Amazon* é de 99,861%. Portanto, é possível concluir que a confiabilidade é mais precisa por considerar o número de quedas e o tempo entre as ocorrências.

2.2.2.3 Capacidade de Sobrevivência

A capacidade de sobrevivência é brevemente descrita por Dantas (2005) como uma propriedade que o sistema possui de continuar operante sob determinadas condições hostis de operação. No presente trabalho, a essa capacidade é discutida e aplicada com o espalhamento dos recursos virtuais sobre múltiplos recursos físicos, diminuindo assim a probabilidade de falha total (BODÍK *et al.*, 2012; CAVALCANTI; OBELHEIRO; KOSLOVSKI, 2014).

2.2.3 Replicação

Réplicas consistem na alocação excedente de recursos para serviços que já encontram-se em operação. A replicação pode ter duas diferentes finalidades (RIGHI, 2013). Uma delas é melhorar o desempenho, diminuindo a sobrecarga de uma máquina e melhorando a escalabilidade. Outra é melhorar a disponibilidade de um serviço, assumindo a operação em caso de falha de um nó. De acordo com Silva, Obelheiro e Koslovski (2017) as técnicas de replicação variam de acordo com o grau de sincronismo e o modelo de implementação.

² Retirado de: <<https://cloudharmony.com/>>

Uma das técnicas é a replicação ativa, implementada de forma que todas as réplicas recebem, processam e respondem a requisição paralelamente. Com ela, o cliente não percebe a ocorrência de uma falha de parada devido à redundância no processamento interno. É importante ressaltar que o cliente recebe apenas uma resposta.

Na técnica de replicação passiva, apenas uma réplica faz todo o trabalho de receber, processar e responder. As demais são atualizadas periodicamente através de *checkpoints* da réplica em uso, geralmente sendo atualizados na ordem de milissegundos (SILVA; OBELHEIRO; KOSLOVSKI, 2017). Quando uma falha de parada ocorre na réplica em uso, outra assume o seu posto a partir do último *checkpoint*. Nesse sistema, o cliente pode receber a ocorrência da falha de parada dado o tempo da operação de troca da réplica em uso. A presente proposta é agnóstica à técnica de replicação, podendo ser aplicada com todos os modelos citados.

2.3 DEFINIÇÃO DO PROBLEMA

A partir dos conceitos expostos anteriormente, fica evidente a possível instabilidade dos provedores de nuvem IaaS. Os clientes ficam a mercê do SLA (*Service Level Agreement* ou acordo de nível de serviço) para ter a qualidade do serviço, sem conhecimento técnico para melhorar a confiabilidade de sua VI. O SLA estabelece as diretrizes do serviço oferecido pelo provedor. Nele o cliente especifica, dentre as opções fornecidas pelo provedor, os requisitos da VI. Tais requisitos podem ser tanto de QoS quanto a própria disponibilidade.

Porém, por mais que o provedor implemente mecanismos internos, o SLA não garante que as diretrizes serão cumpridas, principalmente em relação à disponibilidade. Caso o SLA seja quebrado, o cliente geralmente recebe créditos para utilizar em novos serviços do provedor.

Utilizando informações públicas disponíveis sobre os provedores, a implementação de um corretor de nuvem pode ajudar a preencher a lacuna de conhecimento do cliente. Dessa forma, o corretor pode prover o espalhamento controlado dos recursos virtuais sobre múltiplos provedores, regiões ou zonas para aumentar a capacidade de sobrevivência. Combinando isso à replicação de recursos críticos para garantir a confiabilidade solicitada, tem-se uma solução em potencial para preencher a lacuna identificada. Para posicionar o presente trabalho, a Seção 2.3.1 apresenta um levantamento do estado da arte na literatura especializada.

2.3.1 Trabalhos Relacionados

Para compor o grupo de trabalhos relacionados a partir da literatura especializada, foi utilizada a metodologia de revisão narrativa. Nela foram escolhidos trabalhos que tratam o mesmo domínio e que possuem um objetivo relacionado ao objetivo do presente trabalho. Para tanto, foi elaborada a Tabela 1, que apresenta a visão geral dos trabalhos em ordem crescente de publicação. Ela destaca a função objetivo, o mecanismo de solução que os autores propuseram, os recursos virtuais utilizados, e se os autores utilizam cenários com múltiplos provedores e réplicas.

Tabela 1 – Principais contribuições dos trabalhos relacionados.

Autor	Função objetivo	Mecanismo de solução	Recurso virtual	MP ³	Rép.
Koslovski <i>et al.</i> (2010)	Prover a confiabilidade desejada a cada tarefa na VI	<i>Best fit</i> para reduzir a fragmentação	Switch, VM, enlace	Não	Sim
Yeow, Westphal e Kozat (2011)	Minimizar a quantidade de recursos utilizada para uma VInf.	Programação Linear Inteira Mista	Switch, enlace	Não	Sim
Houidi <i>et al.</i> (2011)	Minimizar a soma dos custos de todas as requisições em redes virtuais	Programação Linear Inteira Mista + Múltiplas heurísticas	Switch, enlace	Sim	Não
Bodik <i>et al.</i> (2012)	Aumentar tolerância a falhas e diminuir o uso da largura de banda	Programação Linear Inteira Mista + <i>Minimum k-way cut</i>	Switch, VM, enlace	Não	Não
Cavalcanti, Obelheiro e Koslovski (2014)	Minimizar as métricas de falhas e a fragmentação	Programação Linear Inteira Mista (ótima)	Switch, VM, enlace	Não	Não
Vieira, Bittencourt e Madeira (2014)	Minimizar o custo de agendamento	Programação Inteira Mista + Heurística <i>Utilization Degree Based</i>	VM	Não	Sim
Houidi, Louati e Zeghlache (2015)	Minimizar o custo de alocação das RVs e o consumo de energia do <i>data center</i>	Programação Linear Inteira Mista	Switch, VM, enlace	Não	Não
Bays <i>et al.</i> (2016)	Minimizar a largura de banda consumida por redes virtuais, considerando o aumento do custo dado pela segurança	Programação Linear Inteira Mista + Heurística <i>annealing-based</i>	Switch, enlace	Não	Não
Oliveira e Koslovski (2017)	Diminuir a fragmentação do <i>data center</i>	Heurísticas baseadas em árvores	Switch, VM, enlace	Não	Não
Caron e Assunção (2016)	Otimizar a utilização de recursos	Heurísticas baseadas em árvores	VM	Não	Não
Souza <i>et al.</i> (2017)	Alocar uma VI entregando os requisitos de QoS, diminuir o custo de alocação e aumentar o lucro do provedor	Programação Linear Inteira Mista + Heurística <i>Deterministic rounding-based</i>	Switch, VM, enlace	Não	Não
Ficco <i>et al.</i> (2018)	Minimizar o número de servidores utilizados, otimizar o balanceamento de recursos e otimizar o cumprimento do SLA de cada requisição	Meta-heurística baseada no paradigma de otimização de recifes de corais para alocação + teoria dos jogos para realocação	VM	Não	Não

Para uma análise mais profunda, os trabalhos relacionados foram divididos em dois grandes grupos discutidos a seguir. O primeiro, na Seção 2.3.1.1, descreve os trabalhos que tratam de alocação de recursos físicos para a hospedagem de VIs. O segundo, na Seção 2.3.1.2, descreve os que propõe técnicas que melhorem a capacidade de sobrevivência e confiabilidade de recursos virtuais.

2.3.1.1 Alocação de Recursos Físicos para Hospedar VI

Houidi *et al.* (2011) propuseram um MIP e um conjunto de heurísticas para resolver o problema de alocação de redes virtuais, focando na redução do custo e aumento da taxa de aceitação. Os autores inovaram propondo a alocação sobre múltiplos provedores de redes virtuais. Neste artigo é compartilhada uma abordagem similar considerando nuvens IaaS, contudo, os detalhes dos provedores não são necessários, nem mecanismos de interoperabilidade. Uma perspectiva diferente é analisada por Caron e Assunção (2016), ao invés de considerar múltiplos provedores, a proposta visa a alocação simultânea sobre um aglomerado privado e um provedor de nuvem pública. Uma alocação ótima em relação a múltiplos critérios de alocação foi proposta. A solução pode ser aplicada em conjunto com nossa abordagem para melhorar a seleção de provedores.

A respeito da alocação de VIs em um único DC, algumas técnicas para minimizar o consumo de largura de banda combinado com suporte a privacidade foi proposto por Bays *et al.* (2016). Em seguida, Oliveira e Koslovski (2017) propõem uma heurística baseada em árvores para diminuir o tempo de alocação da VI. Contudo, a heurística tende a agrupar recursos virtuais, aumentando assim o impacto de uma eventual falha. Apesar de não ter como objetivo uma alocação que preze pela capacidade de sobrevivência, um espalhamento controlado de recursos virtuais sobre um DC foi aplicada por Souza *et al.* (2017), entretanto total conhecimento e controle no DC são necessários.

Também necessitando de total conhecimento e controle no DC, Ficco *et al.* (2018) propôs uma meta-heurística para gerenciar a realocação de recursos elasticamente na infraestrutura da nuvem. O objetivo é realizar uma alocação balanceada entre os interesses do cliente e do provedor, respeitando o SLA durante os processos de gerenciamento. Se tratando de um algoritmo interno, também pode ser utilizado em conjunto com a nossa abordagem, compondo um cenário global de utilização.

Resumindo, a literatura sobre alocação de VIs em DC, ou cenários similares, compreende múltiplas propostas com objetivos distintos (FISCHER *et al.*, 2013). A respeito das abordagens com múltiplos provedores, as técnicas propostas anteriormente dependem de dados sobre a interoperabilidade e/ou o compartilhamento de dados privados dos provedores, enquanto a presente proposta é baseada em informações públicas e pode ser aplicada para qualquer provedor IaaS. Ademais, a presente proposta é agnóstica a mecanismos de alocação privados e pode ser futuramente combinada com qualquer algoritmo.

2.3.1.2 Técnicas para Provisionamento de VIs com Capacidade de Sobrevivência

O provisionamento de VIs com capacidade de sobrevivência foi inicialmente proposto em (KOSLOVSKI *et al.*, 2010). Similarmente ao presente trabalho, o mecanismo utilizou o ORP (*Opportunistic Redundancy Pooling* ou alocação oportunista de redundâncias) para a definição do número de réplicas e subsequentemente espalhou-as em diferentes servidores. Contudo, a alocação foi realizada em um DC controlado com o qual o mecanismo possuía total conhecimento da probabilidade de falha e MTBF. Nesse caso, a alocação foi feita em dois passos, primeiro definindo o número de réplicas e depois aplicando uma heurística de alocação. A abordagem em duas etapas pode levar a uma solução sub-ótima. Grupos de falhas não foram considerados, nem alocação com menor custo. O presente trabalho inova por definir juntamente réplicas e o espalhamento de VMs em múltiplos provedores, regiões e zonas.

A técnica ORP é também aplicada para alocação de redes virtuais (YEOW; WESTPHAL; KOZAT, 2011). Em resumo, um pequeno conjunto de réplicas foi definido para suportar múltiplos clientes. Nós compartilhamos uma visão diferente, considerando um cenário não cooperativo, normalmente observado em provedores públicos. Na verdade, o SLA é individualmente firmado com cada cliente definindo uma confiabilidade alvo.

Além do mais, Bodik *et al.* (2012) melhorou a tolerância a falhas na comunicação de um DC sem aumentar o uso da largura de banda, enquanto Cavalcanti *et al.* (2014) investigou o *tradeoff* entre a fragmentação de um DC e o provisionamento com capacidade de sobrevivência. Vieira, Bittencourt e Madeira (2014) objetivou uma alocação visando diminuir o custo de agendamento, analisando a disponibilidade com um mecanismo *QoS-aware* utilizando réplicas. Cabe ressaltar que a presente proposta combinou o menor custo com uma alocação prezando a capacidade de sobrevivência e a confiabilidade em múltiplos provedores de nuvem, preenchendo uma lacuna de pesquisa no que diz respeito à perspectiva do cliente.

2.4 CONSIDERAÇÕES PARCIAIS

A utilização do serviço IaaS é uma solução de crescente apelo dada a economia que pode proporcionar ao cliente, que não mais precisa manter o seu próprio DC. Além da economia, a escalabilidade e a mensurabilidade dos serviços são outros atrativos. Entretanto, apesar da crescente utilização, estando os maiores serviços do mundo hospedados em provedores de nuvem, a ocorrência de falhas continua comum. Isso impacta negativamente na experiência do usuário final e causa grande prejuízo aos cliente de nuvem.

Assim, há uma lacuna de pesquisa no que diz respeito a confiabilidade e capacidade de sobrevivência do serviço do ponto de vista do cliente. Os provedores não divulgam suas informações internas de falhas, tendo o cliente que confiar cegamente no serviço fornecido.

O problema se agrava quando, a partir dos conceitos e discussões apresentados, é possível afirmar que há um problema de instabilidade da nuvem. Embora os provedores IaaS se preocupem com a estabilidade do seu serviço, eles não podem garantir a não ocorrência de falhas de parada. Portanto, há a necessidade de um serviço intermediário que possa prover uma maior confiabilidade aos clientes.

Com isso, o presente capítulo apresentou os principais conceitos e definições para o entendimento do trabalho. A seguir, o Capítulo 3 explora o problema da alocação de VIs, trazendo a definição formal do problema e apresentando um modelo de programação linear inteira mista (MIP) como proposta de solução.

3 MODELO EXATO PARA ALOCAÇÃO CONFIÁVEL DE VIS EM MÚLTIPLOS PROVEDORES IAAS

O presente capítulo apresenta uma discussão sobre o problema da alocação de VIs e uma proposta de um modelo para a alocação confiável e com capacidade de sobrevivência para VIs em múltiplos provedores IaaS. A Seção 3.1 detalha o problema e apresenta uma definição formal do mesmo através da definição de uma notação que é utilizada no restante do trabalho. Após, a Seção 3.2 apresenta um modelo de Programação Linear Inteira Mista (MIP) para a alocação de VIs em múltiplos provedores, com requisitos que visam garantir a confiabilidade e aprimorar a capacidade de sobrevivência. Ainda, a Seção 3.3 traz a descrição da heurística RS-VIA. Por fim, a Seção 3.4 apresenta um resumo do capítulo elencando as principais conclusões e aprendizados a serem considerados.

3.1 FORMULAÇÃO DO PROBLEMA

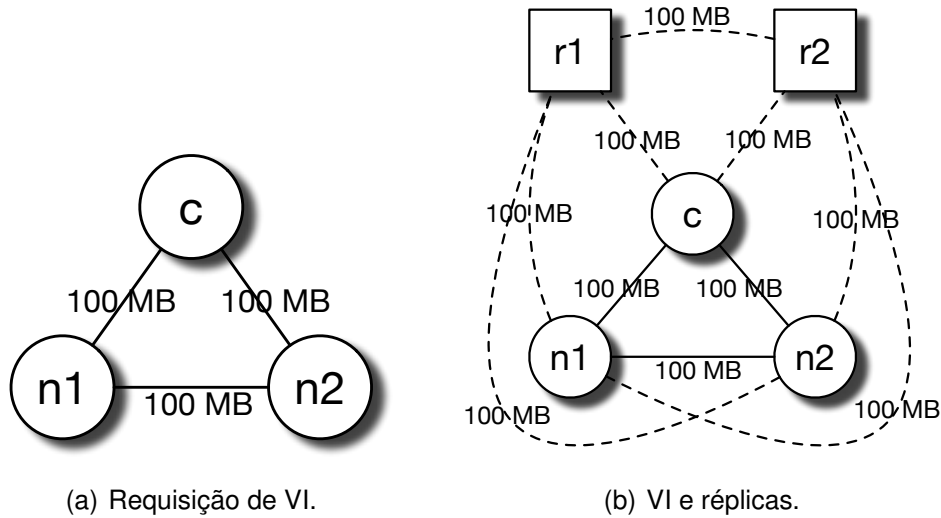
A apresentação da formulação do problema é dividida em quatro etapas. A Seção 3.1.1 especifica a notação utilizada para requisições de VIs e mostra alguns exemplos de requisição. Já a Seção 3.1.2 mostra como a probabilidade de falhas é extraída e utilizada. A Seção 3.1.3 detalha o uso de réplicas na solução proposta. Por fim, a alocação é apresentada na Seção 3.1.4.

3.1.1 Requisições de VIs e Provedores IaaS

A alocação de VI, com o auxílio do corretor proposto, exige que o cliente identifique os componentes críticos e o grau de confiabilidade alvo (YEOW; WESTPHAL; KOZAT, 2011; KOSLOVSKI *et al.*, 2010). Uma VI passa a ser constituída de VMs regulares e críticas. O cliente tem ciência que somente as VMs críticas possuem réplicas. Formalmente, uma requisição de VI é representada por $VI(N, D, V, c)$, sendo N o conjunto de VMs, $D \subset N$ representa as VMs críticas, e V representa os enlaces virtuais entre VMs (cada enlace com uma requisição de transferência de dados, denotada em v_{nm}). A confiabilidade alvo é dada por c (99, 995%, por exemplo).

Para exemplificar, a Figura 2 apresenta duas requisições de VI. Primeiramente, uma VI sem réplicas é representada na Figura 2(a). Para esta mesma VI, o cliente deseja que o nível de confiabilidade atinja $c = 99, 995\%$. Portanto, réplicas das VMs críticas e novos enlaces virtuais são acrescentados, transformando a requisição inicial na representada pela Figura 2(b). Neste exemplo, as requisições de transferência de dados são definidas como 100 MB.

Figura 2 – Requisição de VI com confiabilidade alvo $c = 99,995\%$.



Fonte: Elaboração própria, 2018.

A requisição da VI pode ser alocada sobre um ou múltiplos provedores de nuvem. Cada provedor IaaS é representado por $P(R, Z)$, nos quais os servidores são organizados em regiões (R) e zonas (Z). A Tabela 2 resume a notação utilizada no trabalho.

3.1.2 Probabilidade de Falha

Limitado pela barreira da confidencialidade sobre os números de MTBF, MTTR e MTBO, um cliente deve contar com aproximações para melhorar a configuração de sua VI, especialmente para identificar o número de réplicas necessárias. Quando não disponível, a probabilidade de falhas e os números de confiabilidade podem ser inferidos com base em quedas anteriores. Utilizando o conhecimento sobre as quedas, o MTBF pode ser aproximadamente deduzido. Por exemplo, para os últimos 30 dias (período constantemente utilizado por serviços de monitoramento), tem-se o cálculo representado na Equação 3.1, sendo as durações dadas em horas, conforme normalmente utilizado por provedores IaaS públicos. Seguindo o raciocínio, a probabilidade de falhas (p) é dada pela Equação 3.2, e finalmente, a confiabilidade é definida por $1 - p$.

$$\text{MTBF} = \frac{720 - \sum \text{duração da queda}}{\text{número de quedas}} \quad (3.1)$$

$$p = \frac{1}{\text{MTBF}} \quad (3.2)$$

Tabela 2 – Notação para representar requisições de VIs, provedores IaaS, e detalhes fundamentais do modelo.

Notação	Descrição
Provedores IaaS	
$P(R, Z)$	Um provedor IaaS composto de um conjunto de regiões (R) e zonas (Z).
$j \in R_i$	Uma região j de um provedor i .
$k \in Z_{ij}$	Uma zona k da região j e provedor i .
Requisições de VI	
$VI(N, D, V, c)$	Uma VI composta de N VMs, $D \subset N$ VMs críticas, uma matriz de tráfego (V) e o nível de confiabilidade alvo (c).
$n \in N$	Uma VM regular n .
$m \in D \subset N$	Uma VM crítica m .
$l_{nm} \in V$	Enlace virtual entre VM n e m . Cada enlace solicita um volume de dados a ser transferido v_{nm} .
Detalhes fundamentais do sistema	
B	Conjunto de réplicas para o cenário de falha do pior caso.
$M(i, j, k, c, s)$	Número de réplicas para suportar o nível de confiabilidade c com s VMs críticas no provedor i , região j e zona k .
$C(i, j, n)$	Custo de alocação da VM n no provedor i , região j .
$C_v(z, k)$	Custo de transferência de dados entre as zonas z e k , contabilizado mesmo para diferentes provedores.
x_{nijk}	VM n mapeada no provedor i , região j e zona k (binário).
b_{nijk}	Réplica b mapeada no provedor i , região j e zona k (binário).
xl_{nmzk}	Matriz de mapeamento de enlaces virtuais (nm) para zonas (z and k) (binário).
bl_{nmzk}	Matriz de mapeamento de enlaces de réplicas (nm) para zonas (z and k) (binário).
y_i^p	Número de VMs hospedadas pelo provedor i (inteiro).
y_{ij}^r	Número de VMs hospedadas pelo provedor i e região j (inteiro).
y_{ijk}^z	Número de VMs hospedadas pelo provedor i , região j e zona k (inteiro).

Dados disponíveis publicamente sobre a disponibilidade e falhas (em um período de 30 dias) sobre provedores de nuvem IaaS são monitorados por serviços externos, como o CloudHarmony¹. Por exemplo, em Abril/2017, o CloudHarmony identificou uma disponibilidade de 99,997% para a região *ap-northeast-2* do provedor Amazon EC2, e 99,809% para a região *ams-e* do provedor ElasticHosts. O último teve um

¹ CloudHamony: <<https://cloudharmony.com/>>

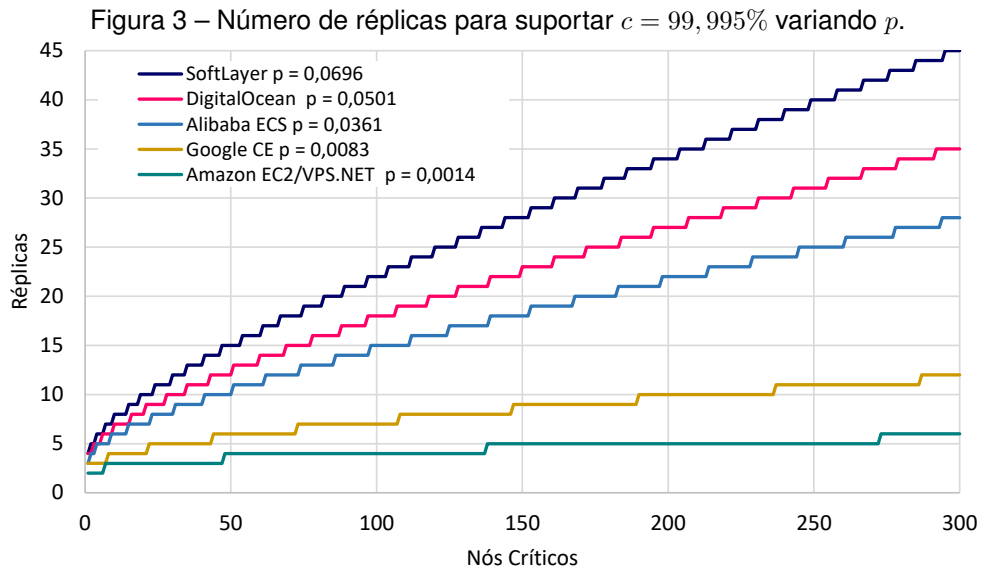
maior número de quedas no período analisado. Desse modo, a confiabilidade é aproximadamente definida como 97,495% e 99,861% para ElasticHosts e Amazon, respectivamente. Vale ressaltar que p é uma aproximação. Qualquer mecanismo capaz de oferecer uma probabilidade com maior acuidade pode ser aplicado. Além disso, a probabilidade representa uma falha independente que pode afetar um único recurso (e.g., um servidor) ou um grupo de recursos (e.g., zona, região). Nesse sentido, fica evidenciado que o espalhamento de VMs e réplicas sobre diferentes grupos de falhas ajuda a diminuir a probabilidade de falha total, aumentando consequentemente a capacidade de sobrevivência da VI (BODÍK *et al.*, 2012; CAVALCANTI; OBELHEIRO; KOSLOVSKI, 2014).

3.1.3 Definição de Réplicas para VMs Críticas

Apesar dos clientes não estarem cientes dos dados de falhas internas dos DCs, eles possuem total conhecimento das suas aplicações, sobretudo dos elementos críticos. Dessa maneira, clientes devem detalhar suas expectativas durante o estabelecimento do SLA (ARMBRUST *et al.*, 2010; KOSLOVSKI *et al.*, 2010), especificamente, o nível de confiabilidade alvo (c) que pode ser superior à configuração padrão dos provedores.

O uso de réplicas é uma abordagem promissora para preencher a lacuna de confiabilidade entre os provedores e a necessidade da VI (RAJAGOPALAN *et al.*, 2012; KOSLOVSKI *et al.*, 2010). Inicialmente, as VMs críticas ($D \subset N$) e a confiabilidade alvo (c) são requisitados. Essa informação é combinada com as probabilidades de falhas dos provedores (regiões e zonas) para aplicar a técnica de ORP (Oportunistic Redundancy Pooling ou Agrupamento Oportunista de Réplicas) (YEOW; WESTPHAL; KOZAT, 2011). O ORP utiliza a função beta incompleta regularizada, $I_{1-p} = (n, k + 1)$, sendo n o número de VMs críticas ($|D|$), $k + 1$ o número de réplicas necessárias, e $1 - p$ o nível de confiabilidade do provedor, região ou zona. Portanto, o número de réplicas é o menor número que garanta c . O número de réplicas necessárias para garantir c com D VMs críticas é calculado para cada zona e representado por M . A Figura 3 exemplifica a composição de M .

Utilizando o ORP, um intervalo de nós críticos suportados por k réplicas é definido. Dado o número de VMs críticas (eixo x), o número de réplicas (eixo y) é calculado para os provedores. Nesse exemplo, a confiabilidade alvo c é 99,995%. Baseando-se nos dados coletados pelo CloudHarmony em Abril/2017, a região VPS.NET (Atlanta) teve uma baixa probabilidade de falha ($p = 0,0014$) e consequentemente apenas 4 réplicas são necessárias para suportar entre 74 e 241 VMs críticas. O conjunto M tem a composição guiada pelo CloudHarmony e indexada por provedor, região, zona, confiabilidade alvo c , e número de VMs críticas.



Fonte: Elaboração própria, 2018.

A Figura 2(b) exemplifica a extensão de uma requisição de VI (Figura 2(a)) adicionando réplicas e enlaces. Para esse exemplo, 2 réplicas são arbitrariamente adicionadas (r_1 e r_2) para atingir a confiabilidade alvo c sobre ambos provedores. As linhas tracejadas representam os novos enlaces virtuais necessários para entregar a conectividade no caso da ocorrência de uma falha.

3.1.4 Alocação de Provedores IaaS para Hospedar VIs

As requisições de VIs confiáveis são individualmente analisadas pelo mecanismo, caracterizando um problema de alocação online (FISCHER *et al.*, 2013). O mapeamento de VMs em zonas é dado por $\mathcal{M} : \mathcal{N} \mapsto \mathcal{Z}$. É relevante mencionar que a política de alocação interna do provedor está fora do escopo do presente trabalho. A seleção de provedores é uma escolha do cliente enquanto os algoritmos de alocação utilizados dentro de um DC (FISCHER *et al.*, 2013; SOUZA *et al.*, 2017; YEOW; WESTPHAL; KOZAT, 2011; OLIVEIRA; KOSLOVSKI, 2017; HOUIDI *et al.*, 2011) são arbitrariamente definidos pelo provedor. Além do mais, o serviço de um corretor pode ser executado a qualquer momento para a utilização de novas probabilidades de falhas. Contudo, mecanismos de realocação e migração não são discutidos, sendo indicados como trabalhos futuros.

Com relação à perspectiva do cliente, o objetivo é a alocação em provedores com menores custos, guiada por requisitos de capacidade de sobrevivência e confiabilidade. A primeira medida visa minimizar o custo de provisionamento de VMs (regulares, críticas e réplicas) e rede (transferência de dados entre VMs), enquanto o segundo objetiva minimizar o impacto de falhas de provedores na VI (CAVALCANTI; OBELHEIRO; KOSLOVSKI, 2014; BODÍK *et al.*, 2012).

Três exemplos de alocação de VI são apresentadas na Figura 4. Para diferenciação dos preços entre provedores e regiões, uma escala de cores é usada. Quanto mais claro o tom da cor, menor o custo da alocação. A mesma abordagem é aplicada nas linhas dos enlaces virtuais. Inicialmente, a Figura 4(a) exemplifica uma alocação diminuindo o custo de provisionamento. Todos os recursos são posicionados em 2 zonas da mesma região. Assim, além de diminuir o custo de provisionamento das VMs, a alocação ameniza os custos de comunicação, já que a transferência de dados dentro de uma mesma zona não é cobrada (cor branca). Uma solução que considera apenas o espalhamento é apresentada na Figura 4(b). As VMs são espalhadas sobre 5 zonas, 4 regiões e 2 provedores, ignorando o custo de provisionamento. De fato, para compor esse cenário, a transferência de dados entre provedores deve ser realizada. Baseado nessa alocação, a probabilidade de uma falha total da VI é minimizada para todos os grupos de falha (provedores, regiões e zonas).

O foco do presente trabalho é identificar uma abordagem intermediária, como a dada pela Figura 4(c). A alocação continua usando 2 provedores e 5 zonas, mas reduz o número de regiões para 3, motivada pelo custo de alocação: a região 1 do provedor 1 é ignorada devido ao alto preço. Além disso, o número de enlaces virtuais comunicando sobre a Internet foi reduzido. Por fim, é importante ressaltar que o nível de confiabilidade c foi atingindo em todos cenários com a adição das réplicas, garantindo a confiabilidade requisitada.

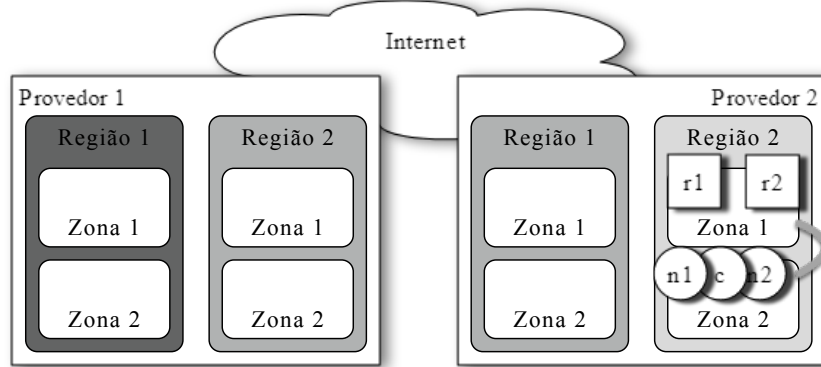
3.2 MODELO EXATO PARA ALOCAÇÃO DE VIS EM MÚLTIPLOS PROVEDORES IAAS

A presente seção apresenta um modelo de Programação Linear Inteira Mista para a alocação de VIs em múltiplos provedores. A decisão de utilizar um MIP se deu pela busca nos trabalhos relacionados discutidos no Capítulo 2 e na natureza do problema, sendo a programação linear tendência para problemas de otimização. A Seção 3.2.1 detalha as variáveis utilizadas e as equações que compõe a função objetivo do MIP. Após, a Seção 3.2.2 apresenta as restrições que precisam ser atendidas pelo mecanismo de alocação para atender corretamente os objetivos.

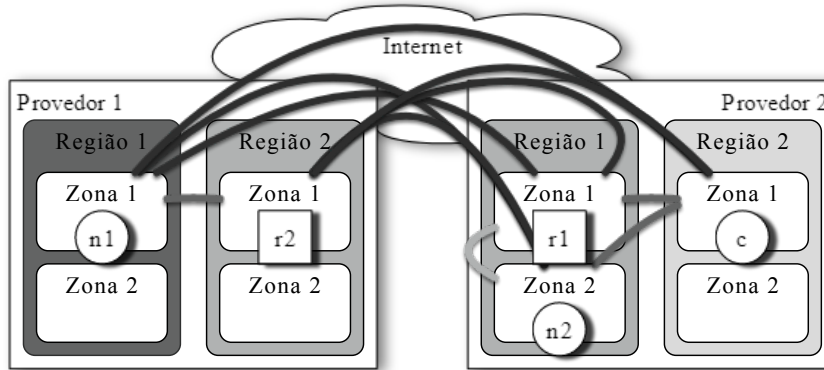
3.2.1 Variáveis e Objetivos

Quatro variáveis são usadas para identificar quais provedores podem hospedar uma requisição de VI, como exemplificado na Figura 2(b). Inicialmente, x_{nijk} , uma variável binária, indica o mapeamento de VMs regulares e críticas ($n \in N$) no provedor i , região j e zona k . Para aplicar a mesma lógica às réplicas, o conjunto B deve ser definido, já que o número exato de réplicas depende de quais provedores, regiões e zonas serão selecionados para hospedar as VMs críticas, sendo tal informação des-

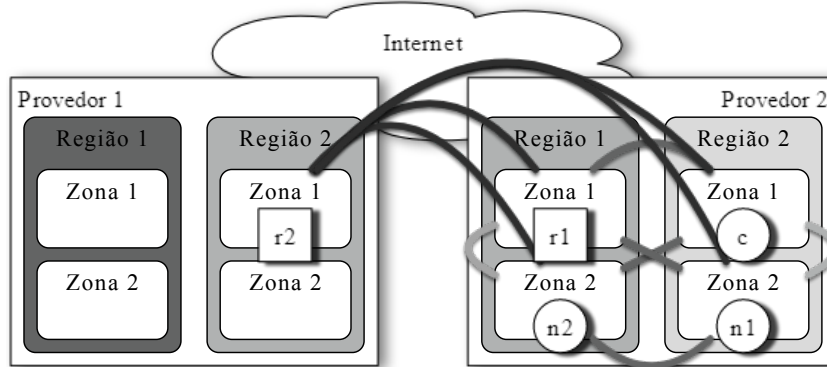
Figura 4 – Alocação de VI com confiabilidade alvo $c = 99,995\%$ e 3 grupos de falha (provedores, regiões e zonas). Quanto mais clara a cor, menor o custo.



(a) Minimização do custo de alocação da VI.



(b) Espalhamento dos recursos da VI sobre provedores.



(c) Alocação visando menor custo, confiabilidade e *survivability*.

Fonte: Elaboração própria, 2018.

conhecida inicialmente. Na perspectiva da capacidade de sobrevivência, B representa o cenário com o pior caso, ou seja, no qual a zona selecionada para hospedar as VMs críticas tem a maior probabilidade de falhas. Contudo, o modelo visa minimizar o número de réplicas necessárias. A alocação de uma réplica é indicada pela variável binária b_{nijk} , na qual $n \in B$.

Para a transferência de dados entre VMs, duas variáveis são usadas, xl e bl .

O primeiro representa a alocação de um enlace virtual l_{mn} entre as VMs n e m , enquanto a segunda segue a mesma lógica para réplicas. A origem n de um enlace l_{nm} é mapeado para a correspondente zona que hospeda n , enquanto o destino m é mapeado para a zona que hospeda m . Para VMs regulares e críticas, l_{nm} são conhecidos inicialmente, enquanto para a conectividade com as réplicas, eles são quantificados *on-the-fly*. Nesse sentido, todas as conexões possíveis entre N (VMs regulares e críticas) e B (réplicas) são analisadas por bl . A conectividade entre réplicas ($B \times B$) também é levada em consideração. Contudo, apenas as conexões necessárias são efetivamente alocadas pelo modelo.

3.2.1.1 Custo de Alocação da VI

Provedores IaaS aplicam diferentes modelos de custos para VMs, normalmente diferenciando por regiões. Nesse sentido, a função $C(i, j, n)$ retorna o custo para hospedar a VM n no provedor i , região j , e as Equações (3.3) e (3.4) contabilizam o custo para hospedar todas as VMs e as réplicas definidas dinamicamente, respectivamente. Em síntese, o custo total de alocação da VI é dado pelo somatório das VMs e réplicas alocadas, multiplicadas pelo custo de alocação na região selecionada.

$$C_{vm}(VI) = \sum_{n \in N} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} x_{nij} \times C(i, j, n) \quad (3.3)$$

$$C_{vmb}(VI) = \sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wij} \times C(i, j, w) \quad (3.4)$$

Os custos para transferência de dados entre VMs são dados pela Equação (3.5) (regulares e críticas) e Equação (3.6) (réplicas). O custo de transferências é diferenciado por zonas, regiões e provedores. Essa informação é abstraída por $C_v(z, k)$, informando o preço por MB para transferência de dados entre as zonas z e k (mesmo entre diferentes provedores). O custo dado pela função $C_v(z, k)$ é multiplicado pelo volume solicitado (v_{nm}) e pela variável binária que indica a alocação (xl para enlaces entre VMs e críticas e bl para com as réplicas).

$$C_{net}(VI) = \sum_{l_{nm} \in V} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} x_{l_{nm} z k} \times v_{nm} \times C_v(z, k) \quad (3.5)$$

$$C_{netb}(VI) = \sum_{l_{nm} \in N \times B} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} b_{l_{nm} z k} \times v_{nm} \times C_v(z, k) + \sum_{l_{nm} \in B \times B} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} b_{l_{nm} z k} \times v_{nm} \times C_v(z, k) \quad (3.6)$$

Por fim, o custo total de alocação de uma VI é dado pela Equação (3.7), agregando as Equações (3.3) até (3.6). Um vetor de pesos (β) é usado para definir o nível de importância de cada componente.

$$C_{total}(VI) = \beta_{vm} \times C_{vm}(VI) + \beta_{ymb} \times C_{ymb}(VI) + \beta_{net} \times C_{net}(VI) + \beta_{netb} \times C_{netb}(VI) \quad (3.7)$$

3.2.1.2 Impacto de Falhas em Provedores IaaS

Uma abordagem intuitiva para diminuir o impacto de falhas em VIs é o espalhamento de recursos virtuais sobre múltiplos domínios (BODÍK *et al.*, 2012; RAJAGOPALAN *et al.*, 2012; CAVALCANTI; OBELHEIRO; KOSLOVSKI, 2014). No contexto deste trabalho, um domínio de falha é um provedor, região ou zona. Uma zona representa a menor unidade, consequentemente apresentando a maior probabilidade de falha. Os demais domínios agregam zonas (ou regiões) e amenizam essa probabilidade. Em resumo, na perspectiva do cliente, quanto maior o espalhamento de recursos virtuais, menor a probabilidade que uma falha possa causar uma indisponibilidade na VI. Formalmente, 3 variáveis inteiras são utilizadas para representar o uso de provedores, regiões e zonas, y_i^p , y_{ij}^r e y_{ijk}^z , respectivamente. As Equações (3.8)-(3.10) contabilizam o número de VMs hospedadas por provedores, regiões e zonas.

$$y_i^p = \sum_{j \in R_i} \sum_{k \in Z_{ij}} \left(\sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \right); \forall i \in P \quad (3.8)$$

$$y_{ij}^r = \sum_{k \in Z_{ij}} \left(\sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \right); \forall i \in P; \forall j \in R_i \quad (3.9)$$

$$y_{ijk}^z = \sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk}; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \quad (3.10)$$

Para espalhar as VMs e réplicas sobre os grupos de falhas, 3 variáveis inteiras e positivas (Eqs. (3.11)-(3.13)) compõem a minimização da função de impacto de falhas (*min* Eq. (3.17)). As variáveis maximizam a distribuição sobre os grupos de falhas respeitando o número de VMs e réplicas (Eqs. (3.14)-(3.16)). Um vetor (γ) diferencia a importância de cada componente.

$$I^p \geq y_i^p; \forall i \in P \quad (3.11)$$

$$I^r \geq y_{ij}^r; \forall i \in P; \forall j \in R_i \quad (3.12)$$

$$I^z \geq y_{ijk}^z; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \quad (3.13)$$

$$I^p \leq |N| + |B|; \forall i \in P \quad (3.14)$$

$$I^r \leq |N| + |B|; \forall i \in P; \forall j \in R_i \quad (3.15)$$

$$I^z \leq |N| + |B|; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \quad (3.16)$$

$$I(VI) = \gamma_p \times I^p + \gamma_r \times I^r + \gamma_z \times I^z \quad (3.17)$$

3.2.1.3 Função Objetivo

A minimização da Equação (3.18) resulta em uma alocação que combina menor custo e maior capacidade de sobrevivência, diminuindo o impacto causado por uma falha. O primeiro termo é normalizado pelo custo de hospedagem na zona mais cara ($C_{max}(VI)$), enquanto o segundo termo é normalizado pelo número de VMs e réplicas, representando o espalhamento máximo possível. O peso α , definido entre 0 e 1, permite balancear a equação priorizando custo ou espalhamento, de acordo com a necessidade da alocação.

$$\min : \left(\alpha \times \frac{C_{total}(VI)}{C_{max}(VI)} \right) + \left((1 - \alpha) \times \frac{I(VI)}{|N| + |B|} \right) \quad (3.18)$$

3.2.2 Restrições para Alocação

Além da função objetivo, o mecanismo de alocação precisa atender um conjunto de restrições definidas previamente. Tais restrições tem a função de garantir matematicamente a alocação correta durante a execução do MIP.

3.2.2.1 Restrições de VMs

A Equação (3.19) indica que a VM n precisa ser alocada uma única vez. Semelhantemente, a Equação (3.20) indica que cada réplica deve ser alocada até uma vez, já que o número de réplicas é determinado *on-the-fly*. Ainda sobre as réplicas, o número mínimo indicado pela aplicação do OPR é dado na função $\min(M)$ e garantido pela Equação (3.21) (YEOW; WESTPHAL; KOZAT, 2011). O número máximo de réplicas é determinado com a alocação das VMs na zona com a maior probabilidade de falha, como indicado pela Equação (3.22).

$$\sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} x_{nijk} = 1; \forall n \in N \quad (3.19)$$

$$\sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{nijk} \leq 1; \forall n \in B \quad (3.20)$$

$$\sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \geq \min(M) \quad (3.21)$$

$$\sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \leq |B| \quad (3.22)$$

3.2.2.2 Restrições de Enlaces

Em relação aos enlaces, as restrições (3.23)-(3.25) garantem que os enlaces virtuais (representando a requisição de transferência de dados, V), são hospedados pelas zonas que estão hospedando origem e destino (BAYS *et al.*, 2016). Por fim, as Equações (3.26)-(3.28) garantem que os enlaces virtuais são hospedados uma vez, no máximo.

$$\sum_{q \in Z_{st}} x_{l_{nm}kq} + \sum_{q \in Z_{st}} x_{l_{nm}zq} = x_{nijk} + x_{mijk} \\ i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in V \quad (3.23)$$

$$\sum_{q \in Z_{st}} b_{l_{nm}kq} + \sum_{q \in Z_{st}} b_{l_{nm}zq} = x_{nijk} + b_{mijk} \\ i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in N \times B \quad (3.24)$$

$$\sum_{q \in Z_{st}} b_{l_{nm}kq} + \sum_{q \in Z_{st}} b_{l_{nm}zq} = b_{nijk} + b_{mijk} \\ i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in B \times B \quad (3.25)$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} x_{l_{nm}kq} = 1 \quad i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in V \quad (3.26)$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} b_{l_{nm}kq} \leq 1 \quad i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in N \times B \quad (3.27)$$

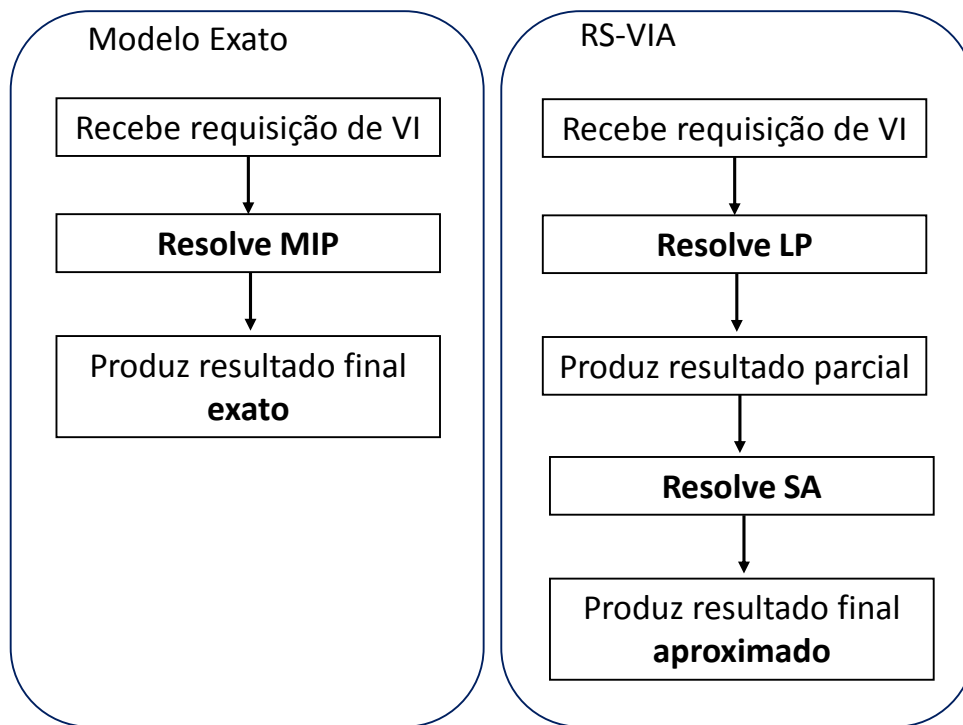
$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} b_{l_{nm}kq} \leq 1 \quad i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in B \times B \quad (3.28)$$

3.3 RELIABLE AND SURVIVABLE VIRTUAL INFRASTRUCTURE ALLOCATION (RS-VIA)

Devida à resolução de um MIP em larga escala ser computacionalmente intratável, possíveis soluções foram buscadas na literatura. Alguns trabalhos aplicaram técnicas determinísticas e aleatórias de arredondamento para interpretar os resultados do LP (CHOWDHURY; RAHMAN; BOUTABA, 2012). Embora eficientes para a alocação de recursos físicos para hospedar redes virtuais, as técnicas não são apropriadas para a seleção de múltiplos provedores. Portanto, nesse trabalho é proposto

o uso de um algoritmo de Cozimento Simulado (*Simulated Annealing* – SA) (KIRKPATRICK; GELATT; VECCHI, 1983) para interpretar o LP, descrito no Algoritmo 1. Assim, o fluxo de execução das etapas do RS-VIA em relação ao modelo exato é ilustrado na Figura 5.

Figura 5 – Fluxograma de execução das etapas



Fonte: Elaboração própria, 2018.

Primeiro, a Seção 3.3.1 apresenta as alterações feitas no MIP proposto no Capítulo 3 para a obtenção de um LP. Após, a Seção 3.3.2 explana a heurística de decisão proposta, considerando o resultado da execução do LP para selecionar a melhor alocação.

3.3.1 RELAXANDO AS VARIÁVEIS E ADICIONANDO RESTRIÇÕES

Para obter um modelo LP (*Linear Programming* ou programação linear) as variáveis x , b , xl e bl são relaxadas (≥ 0 , ≤ 1 , $\in \mathbb{R}$). Ao relaxar o domínio das variáveis, o resultado oferecido pelo MIP é aproximado, não representando uma solução exata.

Para exemplificar o impacto do relaxamento das variáveis, é pertinente ressaltar que o modelo passa a permitir a alocação parcial de uma VM a uma zona. A mesma linha de raciocínio é aplicada para réplicas e enlaces de comunicação. Embora tecnicamente seja viável alocar enlaces sobre múltiplos caminhos e recursos virtuais sobre múltiplos hospedeiros, o presente trabalho busca a solução exata, e para tal uma heurística é proposta para contornar e arredondar os valores aproximados.

Input: VI, x, b, T, α
Output: \mathcal{M} ; VI mapping

```

1  $T = C_{max}(VI)$ 
2  $obj_{best} = T$ 
3  $sol = \emptyset$ 
4 while  $T \geq 1$  do
5   shuffle( $N$ )
6   shuffle( $B$ )
7   for  $n \in N \cup B$  do
8     for  $i \in P, j \in R_i, k \in Z_{ij}$  do
9       if  $n \in N$  then
10         $p_k = x_{nik} \times \sum_{l_{nm} \in V} v_{nm}; m \in \mathcal{M}$ 
11      else
12         $p_k = b_{nik} \times \sum_{bl_{nm} \in V} v_{nm}; m \in \mathcal{M}$ 
13      end
14    end
15     $cand = \emptyset$ 
16    for  $i \in P, j \in R_i, k \in Z_{ij}$  do
17       $s = \lceil \frac{p_k}{\min(p)} \rceil$ 
18      for  $range(1, s)$  do
19         $cand.add(w)$ 
20      end
21    end
22     $c = \text{rand}(cand)$ 
23     $sol \leftarrow [c, z]$ 
24  end
25   $obj = \text{Eq. 3.18}$ 
26  if  $obj \leq obj_{best}$  then
27     $obj_{best} = obj$ 
28     $\mathcal{M} = sol$ 
29   $T = T \times (1 - \alpha)$ 
30 end
31 return  $\mathcal{M}$ 

```

Algorithm 1: RS-VIA: simulated annealing.

3.3.2 HEURÍSTICA DE DECISÃO

O mesmo conjunto de restrições e a função objetivo descritos no Capítulo 3 para o MIP, com as alterações da Seção 3.3.1, foram implementadas como um algoritmo SA. O Algoritmo 1 recebe como entrada a requisição da VI, dois parâmetros (T e α) para controlar a execução do SA, e o resultado da execução do LP dado pelas variáveis relaxadas x, b, xl e bl .

Especificamente no Algoritmo 1, enquanto o critério de cozimento continua a execução (T , linhas 4 e 29), o SA embaralha os conjuntos de VMs N e B para compor uma solução inicial (linhas 5 e 6). Para cada VM, seja crítica, regular ou réplica, um candidato é selecionado baseado nos valores do LP. O conjunto de candidatos, chamado *cand*, é composto de todos os candidatos possíveis previamente identificados

pelo LP ($x > 0, b > 0$). Ao invés de compor *cand* somente baseando-se no LP, RS-VIA leva em consideração o impacto na comunicação analisando o mapeamento anterior (\mathcal{M}) nas linhas 10 e 12. A preferência é dada aos candidatos com maior p_k na medida em que o custo de comunicação é reduzido (linha 17).

Depois de alocar todas as VMs, a função objetivo é calculada (linha 25) e armazenada se melhor que a anterior (linhas 26 até 28). Por fim, uma solução adequada ou um mapeamento (\mathcal{M}) vazio é retornado.

3.4 CONSIDERAÇÕES PARCIAIS

As requisições de VIs são compostas pela descrição da VI com os componentes críticos identificados e a confiabilidade alvo. Somente os elementos críticos são replicados, evitando uma grande aumento no custo de alocação e de comunicação. A alocação pode ser feita em um ou múltiplos provedores.

Como os dados de falhas dos provedores são confidenciais, a utilização de um serviço público de monitoramento permite obter dados de falhas aproximados. Utilizando a probabilidade de falha encontrada é possível aplicar a técnica ORP para chegar ao mínimo número de réplicas que garantem a confiabilidade solicitada (YEOW; WESTPHAL; KOZAT, 2011).

Outro objetivo da alocação é o espalhamento da VI sobre diferentes domínios de falhas, melhorando a capacidade de sobrevivência. O espalhamento é combinado com o objetivo de encontrar o menor custo de alocação, sendo o melhor *tradeoff* entre os dois o objetivo global da alocação.

Dessa forma o MIP é proposto na Seção 3.2, descrevendo matematicamente os objetivos separadamente e combinando suas variáveis numa função de minimização. As restrições são definidas para garantir que os requisitos da VI, como as configurações dos enlaces, sejam cumpridas.

Embora a solução seja exata, um MIP não é escalável a ponto de ser aplicado a grandes requisições ou a um grande conjunto de grupos de falhas, o que o tornaria computacionalmente intratável. Assim, a Seção 3.3 apresenta um modelo LP (*linear programming*) e uma heurística de decisão baseada em SA (*simulated annealing*) para encontrar uma solução aproximada em um tempo viável computacionalmente.

Com a complexidade da solução do modelo exato proposto, o algoritmo RS-VIA permite resolver um LP resultante do relaxamento das variáveis binários do MIP em tempo viável. Assim é possível prover soluções próximas da ótima para cenários em grande escala, tanto de tamanho da VI quanto de número de domínios de falhas.

O Capítulo 4 explora a eficiência tanto do MIP quanto do RS-VIA através de

simulações com dados reais de provedores de nuvens.

4 SIMULAÇÃO E DISCUSSÃO DOS RESULTADOS

Neste capítulo é discutida a eficiência do MIP e da heurística RS-VIA em diferentes cenários de simulação. Inicialmente, a Seção 4.1 apresenta as 7 métricas definidas para avaliar as alocações do ponto de vista do cliente, em relação a custo e capacidade de sobrevivência. Depois, a Seção 4.2 detalha os parâmetros das execuções, como ambiente de execução e configuração das requisições. A Seção 4.3 detalha os cenários de execução, enquanto a Seção 4.4 apresenta e discute os resultados obtidos nesses cenários. Por fim, a Seção 4.5 traz uma revisão e considerações sobre o capítulo.

4.1 MÉTRICAS

Com o objetivo de representar a perspectiva do cliente utilizados dos algoritmos, sete métricas foram selecionadas.

- **Custo de VMs regulares e críticas.** Representado na Equação (3.3), expressa o custo total mensal para manter todas as VMs regulares e críticas da requisição. Esse valor é normalizado pelo custo máximo, que é igual à todas as VMs alocadas na região mais cara.
- **Custo de réplicas.** Semelhante as VMs regulares e críticas, contabiliza o custo total mensal para manter a alocação das réplicas calculadas pelo ORP. É calculado pela Equação (3.4) e normalizado pelo custo máximo necessário para manter as réplicas.
- **Custo de comunicação.** O custo para transferir dados entre VMs, definido pela Equação (3.5). É normalizado pelo custo máximo para transferência, que toma como referência o maior custo entre duas zonas.
- **Custo de comunicação entre réplicas.** o custo para transferir dados entre as VMs e réplicas, definido pela Equação (3.6). Igual anterior, é normalizado pelo custo máximo de comunicação entre duas zonas.
- **Utilização dos provedores.** Métrica utilizada para mensurar o espalhamento sobre os provedores, contabilizando quantos são utilizados. Esse valor é normalizado pelo espalhamento máximo possível, dado por $\min(|N| + |B|, |P|)$, onde P é o número de provedores disponíveis.
- **Utilização das regiões.** Métrica utilizada para mensurar o espalhamento sobre as regiões, contabilizando quantas são utilizadas. Normalizado pelo espalha-

mento máximo possível, dado por $\min(|N| + |B|, |R|)$, onde R é o número de regiões disponíveis.

- **Utilização das zonas.** Métrica utilizada para mensurar o espalhamento sobre as zonas, contabilizando quantas são utilizadas. Normalizado pelo espalhamento máximo possível, dado por $\min(|N| + |B|, |Z|)$, onde Z é o número de zonas disponíveis.

4.2 PARÂMETROS

Para analisar a eficiência do MIP proposto e do RS-VIA, um corretor de nuvem foi implementado em Java v1.8, usando IBM CPLEX optimizer (v12.6.1.0)¹. A simulação foi realizada em um sistema com GNU/Linux Ubuntu 14.04, processador AMD Phenom II X4, 4 GB RAM. A probabilidade de falhas para cada zona foi extraída da plataforma CloudHarmony, especificamente os dados de Agosto/2017. Para compor a função do custo de transferência de dados, os preços foram selecionados uniformemente em 3 intervalos: (i) transferência de dados entre zonas na mesma região: entre \$0,01 e \$0,05; (ii) transferência entre zonas de um mesmo provedor, mas diferentes regiões: entre \$0,1 e \$0,5; (iii) para transferência de dados entre diferentes provedores o preço é selecionado entre \$1,5 e \$2,0. A transferência de dados entre VMs alocadas na mesma zona não possui custo. Caso disponível no futuro, qualquer sistema de precificação mais preciso pode ser aplicado.

Para compor a requisição da VI foi escolhida a instância popular da Amazon EC2, *m3.large* (PERSICO *et al.*, 2015), enquanto para definir a função de custo $C(i, r, n)$, uma configuração similar foi selecionada para cada provedor de nuvem. As VMs foram organizadas seguindo a topologia *full mesh*. Dessa forma, as requisições de transferência em todos os enlaces virtuais foram definidos como 500 MB de volume mensal. Para parametrizar os termos que compõem a função objetivo, cada elemento de β foi configurado com 0,25, enquanto 0,33 foi utilizado para γ . Por representar o termo de importância quanto ao custo ou confiabilidade e espalhamento, os pesos de α são analisados com diversas configurações.

4.3 DEFINIÇÃO DE CENÁRIOS DE SIMULAÇÃO

Para analisar a eficiência e comportamento dos mecanismos propostos na perspectiva do cliente, três cenários de simulação foram definidos.

¹ <<https://www.ibm.com/software/commerce/optimization/cplex-optimizer/>>

4.3.1 MIP

O cenário tem como objetivo analisar a eficiência do MIP proposto na Seção 3.2. A simulação utiliza as informações de 2 provedores de nuvens públicos (Amazon EC2 e Google Computing Engine) totalizando 17 regiões e 24 zonas. Cada requisição de alocação é composta por 5 VMs regulares e 5 VMs críticas. Esse cenário é, ainda, dividido em duas diferentes confiabilidades alvo, 99,95% e 99,995%. A análise de diferentes confiabilidades alvo c requer a procura pelo número exato de réplicas (M , como discutido na Seção 3.1.3).

Para analisar o MIP proposto, 5 abordagens são definidas. Primeiro, a abordagem somente custo (SC, $\alpha = 1$) tem como objetivo apenas minimizar o custo de alocação (Figura 5(a)). Uma abordagem somente espalhamento (SE, ($\alpha = 0$)) considera o espalhamento máximo dos recursos virtuais como principal objetivo, sem se preocupar com o custo de alocação (Figura 5(b)). A alocação exata combina os objetivos de melhor custo e maior espalhamento ($\alpha = 0,5$), provendo uma maior confiabilidade e capacidade de sobrevivência (Figura 5(c)). Para aprofundar a discussão, outras 2 alocações com diferentes valores para α (0,25 e 0,75) foram executadas, resultando em diferentes composições para a alocação (Figura 5(c)).

4.3.2 Alocação Exata e RS-VIA

Nesse cenário são utilizadas as mesmas configurações de VI detalhadas na Seção 4.3.1. Para a comparação, são utilizadas as abordagens somente custo (SC, $\alpha = 1$, Fig. 5(a)), somente espalhamento (SE, ($\alpha = 0$), Fig. 5(b)), alocação exata (EA, $\alpha = 0,5$, Fig. 5(c)), e a RS-VIA. O objetivo da análise é verificar os valores obtidos por RS-VIA frente as linhas de base, quando analisando poucos provedores (a escalabilidade é limitada pela busca da solução exata).

4.3.3 RS-VIA em Larga Escala

Esse cenário analisa a aplicabilidade do algoritmo RS-VIA em dados reais de provedores. A simulação compreende 31 provedores de nuvem pública com uma distribuição geográfica de 133 regiões e 153 zonas.

O cenário compara 3 requisições de VIs compostas de 50 VMs cada, com configurações distintas no número de VMs regulares e críticas (40–10, 25–25, 10–40). Como há na literatura uma lacuna em algoritmos para a alocação confiável e com capacidade de sobrevivência de VIs sobre múltiplos provedores (conforme constatado na Seção 2.3.1), a comparação é realizada com um algoritmo baseado na seleção aleatória de candidatos, representando o comportamento de diversos clientes. Cada

abordagem é executada 10 vezes e os valores apresentados são a média com o desvio padrão.

4.4 RESULTADOS

Os resultados são apresentados através de gráficos de colunas e radar. No gráfico de colunas, utilizado na Seção 4.4.3 as variáveis são posicionadas lado a lado. Já no gráfico radar, utilizado nas demais seções, os eixos partem do mesmo ponto. O ponto central representa o valor zero, e cada eixo representa uma variável. Assim, as séries são sobrepostas facilitando a comparação.

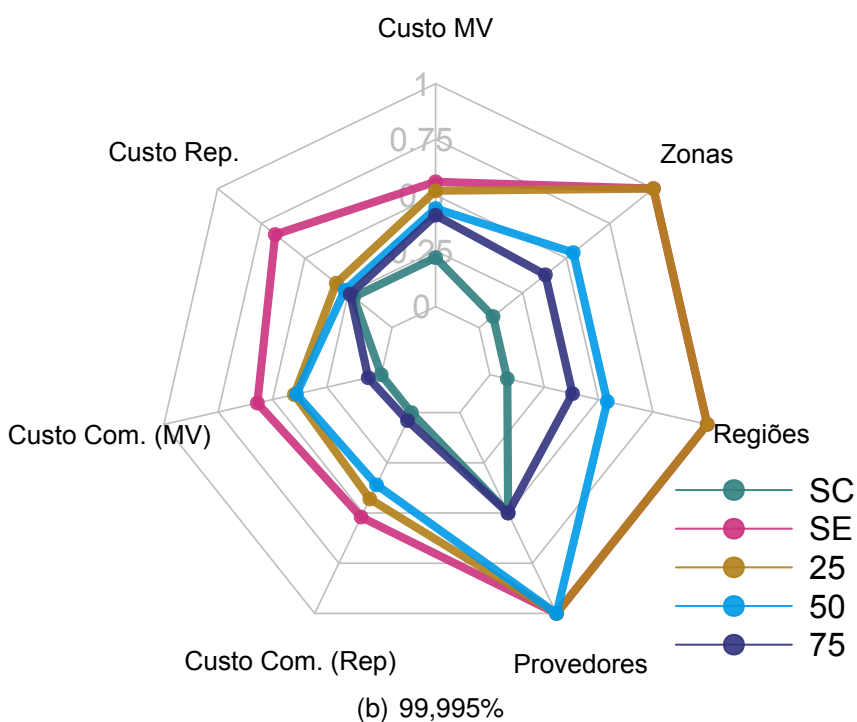
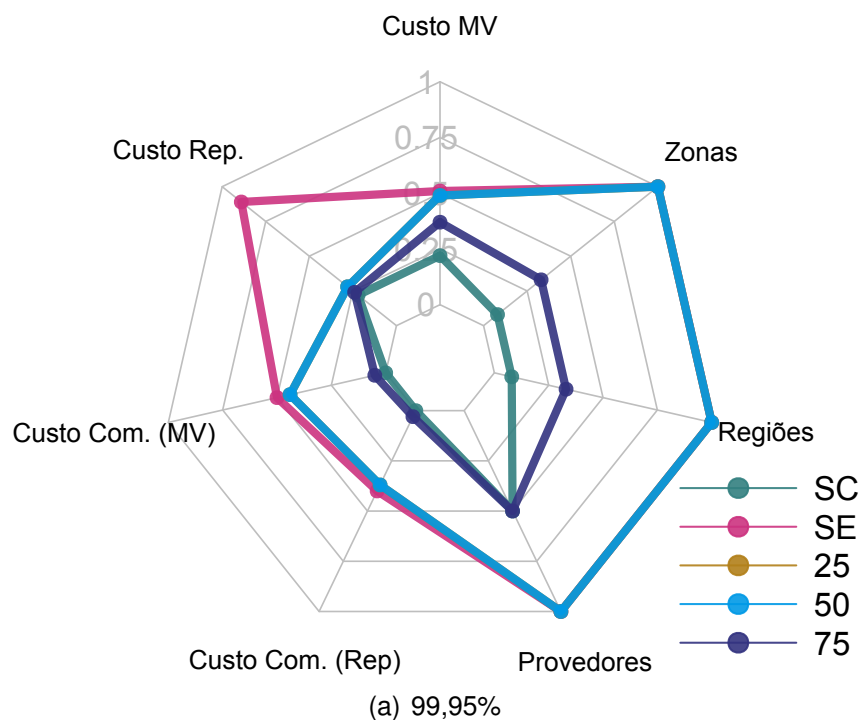
Em ambos os gráficos, os dados devem ser lidos considerando dois grupos de variáveis. As variáveis de custo, nas quais os melhores resultados apresentam os menores valores, e as variáveis de capacidade de sobrevivência que representam a utilização de provedores, regiões e zonas, nas quais as melhores alocações se caracterizam pelos maiores valores.

4.4.1 Execução do MIP

Os resultados das simulações são exibidos na Figura 6. A Figura 6(a) mostra a alocação para a confiabilidade alvo de 99,95%, enquanto a Figura 6(b) para $c = 99,995\%$. Como esperado, a abordagem SC possui a menor área, priorizando a concentração das VMs em regiões com o menor preço. Além disso, as réplicas são hospedadas pelas mesmas zonas em ambos os cenários (o menor preço). Já a abordagem SE resultou no comportamento contrário: alocou mais provedores, regiões e zonas, aumentando tanto o custo de alocação quanto o de comunicação. Nessa abordagem, o espalhamento da VI é realizado sem considerar o custo de provisionamento. Todos os provedores, regiões e zonas são candidatos para receber a VM, e certamente zonas mais caras são selecionadas.

A alocação exata se beneficia das vantagens de ambas as abordagens. Na Figura 6(a), as alocações com $\alpha = 0,25$ e $0,5$ obtiveram os mesmos resultados, alcançando o espalhamento máximo com um custo menor que a abordagem SE, enquanto a alocação com peso $\alpha = 0,75$ obteve custos próximos da abordagem SC provendo um maior espalhamento. No cenário representado pela Figura 6(b), cada peso obteve uma alocação diferente. Com $\alpha = 0,25$, o espalhamento máximo foi alcançado com um custo menor que a abordagem SE. Com $\alpha = 0,5$ os valores obtidos foram intermediários, com menor espalhamento e custo que o observado no cenário anterior. Por fim, $\alpha = 0,75$ proporcionou um custo próximo da abordagem SC com uma melhor capacidade de sobrevivência. Analisando tais resultados fica evidente a eficiência da alocação exata realizada com o MIP proposto.

Figura 6 – Resultados da alocação com MIP.



Fonte: Elaboração do autor, 2018.

Por fim, os resultados evidenciam a possibilidade de configuração oferecida ao cliente. Através da definição de pesos, é possível indicar uma maior preferência para redução de custos ou para aumento da confiabilidade e capacidade de sobrevivência. A análise apresentada indicou a aplicabilidade do modelo proposto, paralelamente evidenciando a limitação de escalabilidade oriunda da natureza do problema (NP-Difícil).

4.4.2 MIP Exato com RS-VIA

Na Figura 7(a) é possível destacar a proximidade que a abordagem RS-VIA teve da AE. RS-VIA teve um custo de réplicas pouco menor que o AE, porém teve um menor custo de comunicação entre réplicas. Já na Figura 7(b), os valores encontrados pelo RS-VIA na utilização de regiões e zonas assim como no custo de VMs e réplicas ficaram muito próximos da abordagem AE. Porém, RS-VIA utiliza um provedor a menos na alocação, o que reduz drasticamente os custos de comunicação e explica a diferença aparente entre as abordagens.

O comportamento identificado na busca por uma solução aproximada, efetuada pelo algoritmo RS-VIA, é explicado pela Equação 3.18. Nela, uma solução aproximada pode ser alcançada com um diferente *tradeoff*. No caso da Figura 7(b), RS-VIA ficou próximo do valor da função objetivo encontrado pelo EA (0,68 e 0,64, respectivamente).

Em suma, RS-VIA rompe a limitação de escalabilidade indicada na Seção 4.4.1, apresentando resultados competitivos com aqueles encontrados pelo EA. É importante ressaltar que a liberdade de configuração de pesos permanece disponível para os usuários, uma vez que RS-VIA utiliza a solução relaxada como entrada do algoritmo.

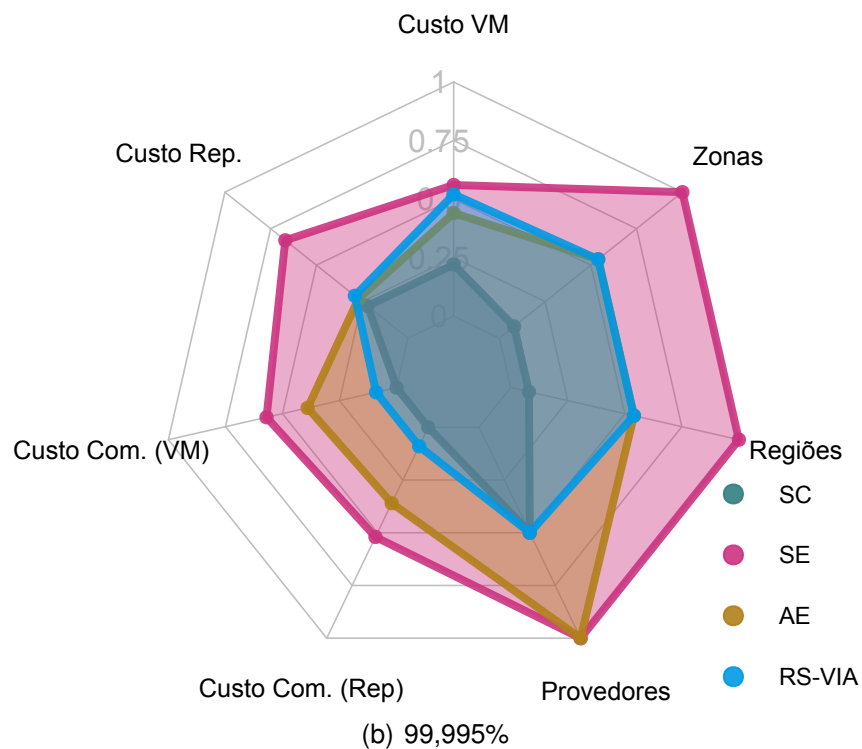
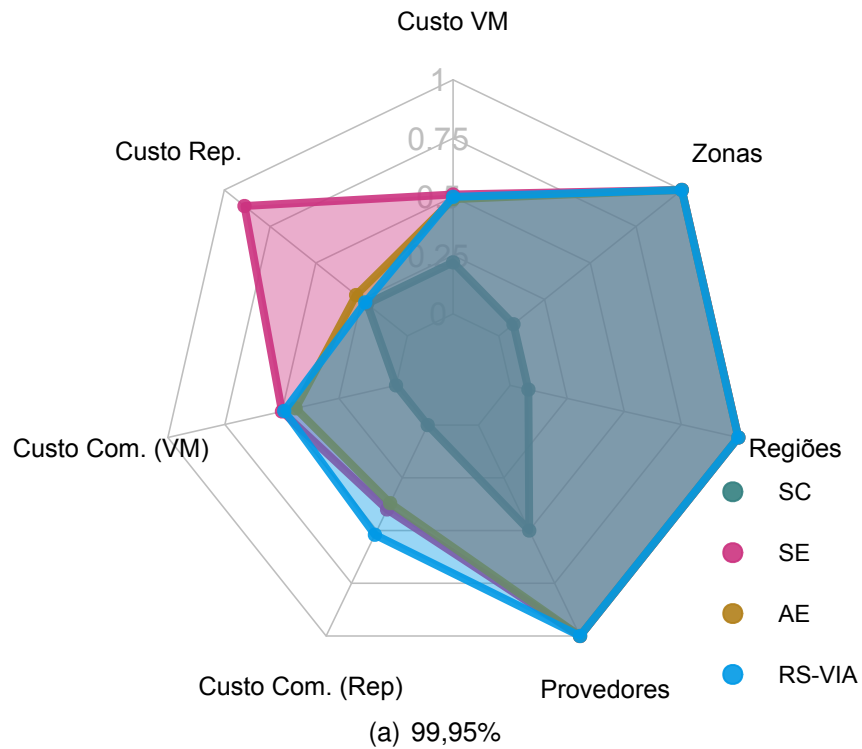
4.4.3 RS-VIA

Para um cenário em grande escala, a Figura 8 exhibe os diferentes comportamentos da alocação com RS-VIA e com a seleção aleatória de candidatos (RND). A abordagem aleatória espalha os recursos aumentando os custos de VMs. Porém, os custos de comunicação permanecem próximos ao RS-VIA, sendo estatisticamente equivalentes.

As alocações com o RS-VIA reduziram drasticamente o custo de VMs e réplicas, atingindo uma solução eficiente. Além disso, o RS-VIA manteve o espalhamento controlado baseado no modelo exato, provendo capacidade de sobrevivência para a VI.

Quanto a escalabilidade, RS-VIA analisou 31 provedores com requisições superiores aquelas discutidas nas Seções 4.4.1 e 4.4.2. Ou seja, a proposta permite a análise de diversos provedores simultaneamente, aumentando a possibilidade de espalhamento dos recursos virtuais.

Figura 7 – Resultados da alocação com MIP e RS-VIA.

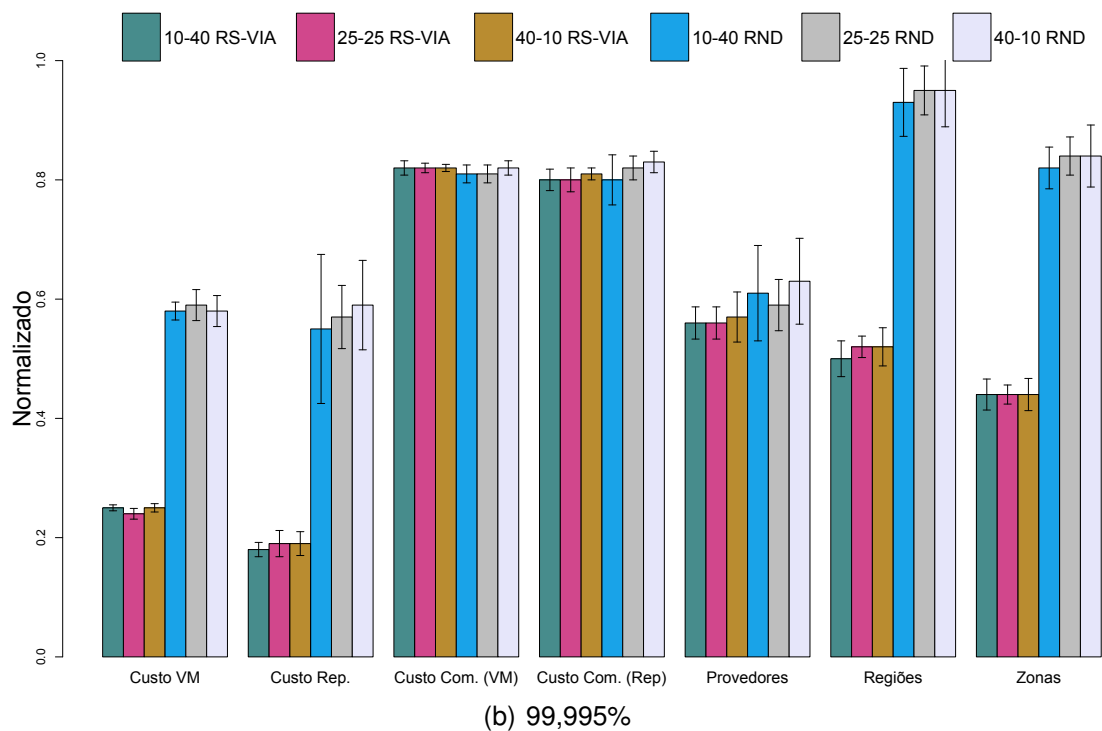
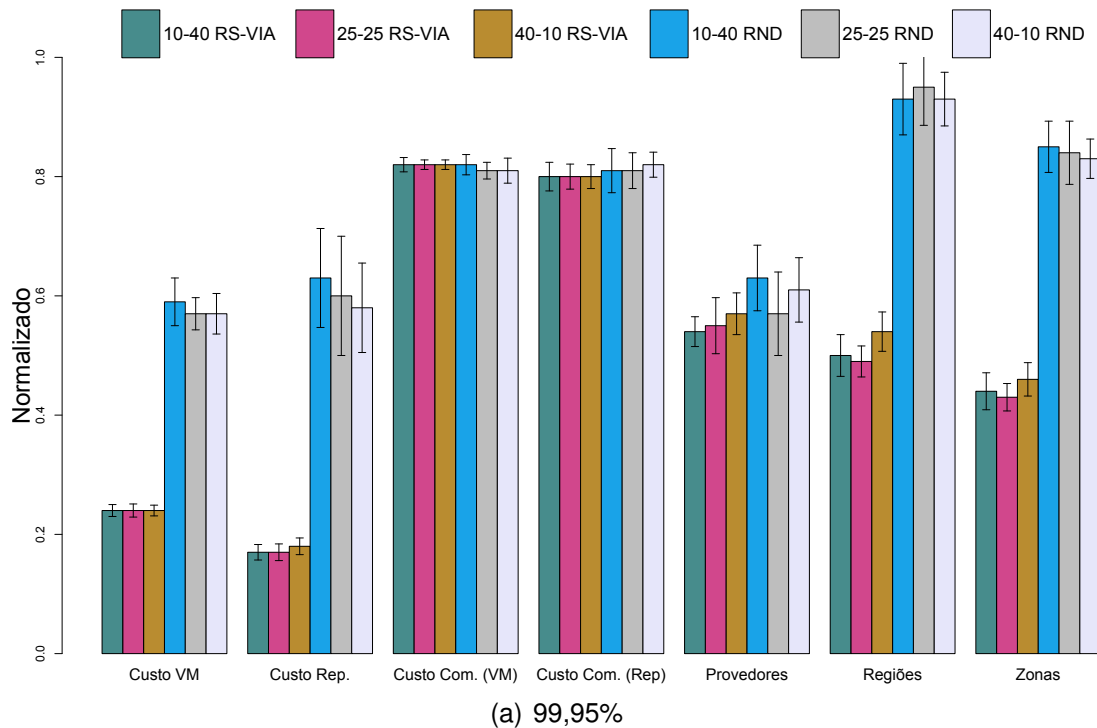


Fonte: Elaboração do autor, 2018.

4.5 CONSIDERAÇÕES PARCIAIS

Neste trabalho foram propostos um modelo exato de alocação de VIs em múltiplos provedores IaaS e a heurística de aproximação RS-VIA, descritos no Capítulo 3). Para avaliar a eficiência de tais mecanismos, um conjunto de simulações com dados

Figura 8 – Resultados da alocação com RS-VIA.



Fonte: Elaboração do autor, 2018.

reais de provedores foram realizadas.

Foram analisados três cenários. O primeiro buscou analisar a eficiência do modelo exato através de simulação em cenário reduzido, comparando as abordagens SC, SE e AE com diferentes balanceamentos (representado em α). O segundo cenário comparou a alocações feitas pelas abordagens SC, SE e AE com o algoritmo RS-VIA.

O terceiro explorou a eficiência do algoritmo RS-VIA em um cenário de larga escala, comparando com uma abordagem de alocação aleatória.

Os resultados do primeiro cenário indicaram que a alocação exata consegue realizar em uma alocação intermediária, utilizando o conhecimento sobre os provedores para realizar um espalhamento controlado dos recursos sobre os provedores sem que o custo seja aumentado drasticamente. Tomando como base a alocação exata mais equilibrada ($\alpha = 0,5$), os resultados da simulação do segundo cenário mostraram que o RS-VIA obtém valores muito próximos do AE, variando minimamente o *tradeoff* da alocação. No terceiro e último cenário utilizando uma grande escala de dados, o RS-VIA mostrou um comportamento similar ao cenário de menor porte, diferenciando principalmente no custo de comunicação, visto que utiliza mais provedores o que impacta nesse custo.

Todas as alocações garantem a confiabilidade requisitada por meio de réplicas. O tempo de execução do RS-VIA em grande escala de dados é tratável computacionalmente. Por fim, é possível afirmar que ambos os mecanismos propostos cumprem os objetivos iniciais, aumentando a capacidade de sobrevivência e a confiabilidade, sem aumentar drasticamente o custo.

5 CONCLUSÃO

A computação em nuvem continua se popularizando entre as empresas. Entre os modelos de serviço oferecidos, o modelo IaaS fornece infraestrutura virtualizada aos clientes, que não mais precisam se preocupar em manter seu próprio DC. Contudo, falhas ocorrem em nuvens computacionais. Embora o DC tente garantir a qualidade do serviço via SLA, ele não pode garantir a não ocorrência de falhas. Isso prejudica serviços de precisam estar operantes ininterruptamente. Além do mais, os dados de falhas dos provedores de nuvens são mantidos em sigilo, o que prejudica possíveis soluções partindo do cliente, que não tem conhecimento suficiente para encontrar uma forma de contornar o problema.

Dessa forma, é necessário um serviço ou mecanismo que garanta a confiabilidade solicitada pelo cliente. Analisando os papéis no modelo de computação em nuvem, tal serviço deve se posicionar como um corretor de nuvem, utilizando informações públicas dos provedores para gerenciar a alocação da VI solicitada pelo cliente.

Analisando essa lacuna, o presente trabalho apresentou uma alternativa para aumentar a capacidade de sobrevivência de uma VI, garantindo a confiabilidade alvo através de réplicas, sem aumentar significativamente o custo de alocação da VI. Para isso, um MIP foi formulado para definir a alocação exata da VI sobre múltiplos provedores. Como a execução de um MIP para uma grande escala de dados é inviável, o algoritmo de aproximação RS-VIA também foi proposto utilizando um modelo LP e baseando-se em SA como critério de parada.

Para avaliar a eficiência dos mecanismos propostos, três cenários de simulação foram propostos. O primeiro buscou analisar a eficiência do modelo exato com simulações em cenário reduzido, comparando as abordagens SC, SE e a alocação exata. O segundo cenário teve como objetivo avaliar a eficiência do algoritmo RS-VIA em relação ao MIP. Dessa forma, comparou as alocações feitas pelas abordagens SC, SE, AE e RS-VIA. O terceiro explorou a eficiência do algoritmo RS-VIA com uma grande escala de dados. Como não foram encontrados na literatura outros algoritmos com o mesmo objetivo, ele foi comparado com uma alocação totalmente aleatória.

Os resultados mostram que o MIP é eficiente em termos de confiabilidade e capacidade de sobrevivência, sem aumentar o custo de provisionamento. O custo total permanece o mais próximo possível do mínimo, respeitando a confiabilidade alvo. O algoritmo RS-VIA apresenta valores próximos ao MIP no cenário reduzido. Executando o algoritmo com uma grande escala de dados ele se mantém eficiente, diminuindo o custo de alocação e permitindo os espalhamento controlado dos recursos que compõe

a VI.

5.1 TRABALHOS FUTUROS

As principais limitações do trabalho foram identificadas nas simulações e análise dos resultados. A implementação de um corretor de nuvem e a atuação sobre um ambiente real permitirá fazer uma análise mais precisa da aplicabilidade do modelo exato e do RS-VIA. Além dos algoritmos aqui propostos, mais mecanismos poderão ser incorporados ao corretor.

5.2 PUBLICAÇÕES

As publicações realizadas se dividem em diretas, que foram originadas da presente pesquisa, e em colaboração.

5.2.1 Diretas

As publicações originadas a partir do presente trabalho são apresentadas na Tabela 3.

Tabela 3 – Publicações diretas.

Evento/Veículo	Título	Ano
Escola Regional de Alto Desempenho (ERAD)	Alocação de Infraestruturas Virtuais Confiáveis em Múltiplos Provedores IaaS: Um Estudo Preliminar	2017
International Conference on Advanced Information Networking and Applications (AINA)	Allocation of Virtual Infrastructures on Multiple IaaS Providers with Survivability and Reliability Requirements	2018
Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)	Alocação de Infraestruturas Virtuais Confiáveis em Múltiplos Provedores IaaS	2018

Fonte: Elaboração própria, 2018.

5.2.2 Trabalhos em Colaboração

A Tabela 4 apresenta os dados do trabalho gerado em colaboração, com o intuito de auxiliar um estudante de Iniciação Científica na escrita e revisão do artigo.

Tabela 4 – Publicações em colaboração.

Autores	Evento/Veículo	Título	Ano
XAVIER, G.; RAUGUST, A.; KOSLOVSKI, G.	Escola Regional de Alto Desempenho (ERAD)	Aplicação de Ferramentas de Big Data para Alocação de Infraestruturas Virtuais	2017

Fonte: Elaboração própria, 2018.

REFERÊNCIAS

- AMAZON AWS. **Produtos em nuvem**. 2018. Disponível em: <<https://aws.amazon.com/pt/products/>>. Acesso em: 19 jan. 2018.
- AMAZON EC2. **Amazon Web Services Summary of the Amazon S3 Service Disruption in the Northern Virginia (US-EAST-1) Region**. 2017. Disponível em: <<https://aws.amazon.com/pt/message/41926/>>.
- ANHALT, F.; KOSLOVSKI, G.; PRIMET, P. V.-B. Specifying and provisioning virtual infrastructures with hipernet. **International Journal of Network Management**, Wiley Online Library, v. 20, n. 3, p. 129–148, 2010.
- ARMBRUST, M. *et al.* A View of Cloud Computing. **Commun. ACM**, ACM, New York, NY, USA, v. 53, n. 4, p. 50–58, abr. 2010. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/1721654.1721672>>.
- AVIZIENIS, A. *et al.* Basic concepts and taxonomy of dependable and secure computing. **IEEE transactions on dependable and secure computing**, IEEE, v. 1, n. 1, p. 11–33, 2004.
- AVRAM, A. **Amazon EC2 Outage Explained and Lessons Learned**. [S.l.], 2011. Disponível em: <<https://www.infoq.com/news/2011/04/Amazon-EC2-Outage-Explained>>.
- BAYS, L. R. *et al.* A toolset for efficient privacy-oriented virtual network embedding and its instantiation on SDN/OpenFlow-based substrates. **Computer Communications**, Elsevier, v. 82, p. 13–27, 2016.
- BENSON, T.; AKELLA, A.; MALTZ, D. A. Network traffic characteristics of data centers in the wild. In: ACM. **Proceedings of the 10th ACM SIGCOMM conference on Internet measurement**. [S.l.], 2010. p. 267–280.
- BODÍK, P. *et al.* Surviving failures in bandwidth-constrained datacenters. In: **Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication**. New York, NY, USA: ACM, 2012. (SIGCOMM '12), p. 431–442. ISBN 978-1-4503-1419-0. Disponível em: <<http://doi.acm.org/10.1145/2342356.2342439>>.
- CARON, E.; ASSUNÇÃO, M. D. de. Multi-criteria malleable task management for hybrid-cloud platforms. In: **2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)**. Marrakech, Morocco: IEEE, 2016. p. 326–333.
- CAVALCANTI, G. A. S.; OBELHEIRO, R. R.; KOSLOVSKI, G. Optimal resource allocation for survivable virtual infrastructures. In: **Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the**. Ghent, Belgium: IEEE, 2014. p. 1–8.

CHOWDHURY, M.; RAHMAN, M.; BOUTABA, R. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. **IEEE/ACM Transactions on Networking**, v. 20, n. 1, p. 206–219, Feb 2012. ISSN 1063-6692.

CHOWDHURY, N. M. K.; BOUTABA, R. A survey of network virtualization. **Computer Networks**, Elsevier, v. 54, n. 5, p. 862–876, 2010.

DANTAS, M. A. **Computação distribuída de alto desempenho: redes, clusters e grids computacionais**. [S.l.]: Axcel books, 2005.

FICCO, M. *et al.* A coral-reefs and game theory-based approach for optimizing elastic cloud resource allocation. **Future Generation Computer Systems**, Elsevier, v. 78, p. 343–352, 2018.

FISCHER, A. *et al.* Virtual network embedding: A survey. **IEEE Communications Surveys & Tutorials**, IEEE, v. 15, n. 4, p. 1888–1906, 2013.

GLOVER, F. W.; KOCHENBERGER, G. A. **Handbook of metaheuristics**. [S.l.]: Springer Science & Business Media, 2006. v. 57.

GOOGLE CLOUD PLATAFORM. **Produtos e Serviços**. 2018. Disponível em: <<https://cloud.google.com/products/>>. Acesso em: 19 jan. 2018.

GOVINDAN, R. *et al.* Evolve or die: High-availability design principles drawn from googles network infrastructure. In: **ACM. Proc. of the 2016 ACM SIGCOMM conference**. Florianópolis, Brazil: ACM, 2016. p. 58–72.

HERSHER, R. **Amazon And The \$150 Million Typo**. 2017. Disponível em: <<https://www.npr.org/sections/thetwo-way/2017/03/03/518322734/amazon-and-the-150-million-typo>>. Acesso em: 19 jan. 2018.

HILLIER, F. S.; LIEBERMAN, G. J. **Introdução à pesquisa operacional**. [S.l.]: McGraw Hill Brasil, 2013.

HOUIDI, I. *et al.* Virtual network provisioning across multiple substrate networks. **Computer Networks**, Elsevier, v. 55, n. 4, p. 1011–1023, 2011.

HOUIDI, I.; LOUATI, W.; ZEGHLACHE, D. Exact multi-objective virtual network embedding in cloud environments. **The Computer Journal**, Br Computer Soc, v. 58, n. 3, p. 403–415, 2015.

JUNIOR, E. C.; MIERS, C. C.; KOSLOVSKI, G. P. Uma taxonomia para corretagem de nuvens iaas baseada na migração de infraestruturas virtuais. **Revista Brasileira de Computação Aplicada**, v. 9, n. 1, p. 15–30, 2017.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **Science**, American Association for the Advancement of Science, v. 220, n. 4598, p. 671–680, 1983. ISSN 00368075. Disponível em: <<http://www.jstor.org/stable/1690046>>.

KOSLOVSKI, G. *et al.* Reliability support in virtual infrastructures. In: **2010 IEEE Second International Conference on Cloud Computing Technology and Science**. Indianapolis, USA: IEEE, 2010. p. 49–58.

MELL, P. M.; GRANCE, T. **SP 800-145. the NIST definition of cloud computing**. [S.l.], 2011.

MICROSOFT AZURE. **Microsoft Azure: Azure products**. 2018. Disponível em: <<https://azure.microsoft.com/en-us/services/>>. Acesso em: 19 jan. 2018.

MIERS, C. C.; KOSLOVSKI, G. P. Análise de segurança para soluções de computação em nuvem. 2014.

NOORMOHAMMADPOUR, M.; RAGHAVENDRA, C. S. Datacenter traffic control: Understanding techniques and trade-offs. **IEEE Communications Surveys & Tutorials**, IEEE, 2017.

OLIVEIRA, R. de; KOSLOVSKI, G. P. A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. **International Journal of Network Management**, John Wiley & Sons, Ltd, v. 27, n. 1, p. e1958, 2017. ISSN 1099-1190.

PERSICO, V. *et al.* Measuring network throughput in the cloud. **Comput. Netw.**, Elsevier North-Holland, Inc., New York, NY, USA, v. 93, n. P3, p. 408–422, dez. 2015. ISSN 1389-1286. Disponível em: <<http://dx.doi.org/10.1016/j.comnet.2015.09.037>>.

PFAFF, B. *et al.* The design and implementation of open vswitch. In: **NSDI**. [S.l.: s.n.], 2015. p. 117–130.

RACKSPACE. **Rackspace products**. 2018. Disponível em: <<https://blog.rackspace.com/category/products>>. Acesso em: 19 jan. 2018.

RAJAGOPALAN, S. *et al.* Secondsite: Disaster tolerance as a service. **SIGPLAN Not.**, ACM, New York, NY, USA, v. 47, n. 7, p. 97–108, mar. 2012. ISSN 0362-1340. Disponível em: <<http://doi.acm.org/10.1145/2365864.2151039>>.

RIGHI, R. da R. Elasticidade em cloud computing: conceito, estado da arte e novos desafios. **Revista Brasileira de Computação Aplicada**, v. 5, n. 2, p. 2–17, 2013.

ROSENBERG, J.; MATEOS, A. **The Cloud at Your Service**. 1st. ed. [S.l.]: Manning Publications Co., 2010. ISBN 1935182528, 9781935182528.

SCHOO, P. *et al.* Challenges for cloud networking security. In: SPRINGER. **International Conference on Mobile Networks and Management**. [S.l.], 2010. p. 298–313.

SILVA, M. P. da; OBELHEIRO, R. R.; KOSLOVSKI, G. P. Adaptive remus: adaptive checkpointing for xen-based virtual machine replication. **International Journal of Parallel, Emergent and Distributed Systems**, Taylor & Francis, v. 32, n. 4, p. 348–367, 2017.

SOTOMAYOR, B. *et al.* Virtual infrastructure management in private and hybrid clouds. **IEEE Internet computing**, IEEE, v. 13, n. 5, p. 14–22, 2009.

SOUZA, F. R. de *et al.* QoS-Aware Virtual Infrastructures Allocation on SDN-based Clouds. In: **Proceedings of the 2017 IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing**. Madrid, Spain: IEEE Computer Society, 2017. (CC-GRID '17).

STALLINGS, W. **Foundations of Modern Networking: SDN, NFV, QoE, IoT, and Cloud**. [S.l.]: Addison-Wesley Professional, 2015.

UCLA, A. A. *et al.* **Fundamental concepts of dependability**. Newcastle, U.K.: University of Newcastle upon Tyne, Computing Science, 2001.

UNBEHAGEN, P. *et al.* Cloud reference framework. 2015.

VIEIRA, C. C.; BITTENCOURT, L. F.; MADEIRA, E. R. Reducing costs in cloud application execution using redundancy-based scheduling. In: IEEE COMPUTER SOCIETY. **Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing**. [S.l.], 2014. p. 117–126.

YEOW, W.-L.; WESTPHAL, C.; KOZAT, U. C. Designing and embedding reliable virtual infrastructures. **ACM SIGCOMM Computer Communication Review**, ACM, v. 41, n. 2, p. 57–64, 2011.

YU, M. *et al.* Rethinking virtual network embedding: substrate support for path splitting and migration. **ACM SIGCOMM Computer Communication Review**, ACM, v. 38, n. 2, p. 17–29, 2008.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of internet services and applications**, Springer, v. 1, n. 1, p. 7–18, 2010.