Alocação de Infraestruturas Virtuais Confiáveis em Múltiplos Provedores IaaS

Anderson S. Raugust[1], Felipe R. de Souza[1], Maurício A. Pillon[1], Charles C. Miers[1], Guilherme P. Koslovski[1]

[1]Programa de Pós-Graduação em Computação Aplicada (PPGCA)
Universidade do Estado de Santa Catarina (UDESC) – Joinville, SC – Brasil

anderson.raugust@edu.udesc.br, dcc6frs@joinville.udesc.br,
{mauricio.pillon, charles.miers, guilherme.koslovski}@udesc.br

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. Neste contexto, esse trabalho apresenta um corretor de nuvem que visa melhorar dois aspectos: confiabilidade e survivability de uma Infraestrutura Virtual (IV). Para atingir a confiabilidade especificada pelo cliente, utilizam-se réplicas, e para minimizar a probabilidade de falhas críticas, o espalhamento. A alocação de IV, gerenciado pelo corretor de nuvem proposto, considera os requisitos de confiabilidade e survivability como um modelo de Programação Linear Inteira Mista (MIP). Os resultados demonstram que o corretor proposto incrementa o grau de survivability de IVs sem um aumento significativo de custos.*

Abstract. *The cloud computing paradigm consolidated the on-demand provisioning of virtual resources. However, the diversity of services, prices, Data Centers (DCs), 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 service capabilities. Due to management complexity, the survivability and reliability aspects are often disregarded by tenants, eventually resulting on heavy losses due to unavailability of services hosted by Virtual Infrastructures (VIs). We present an alternative to improve VIs survivability and reliability, taking into account the use of replicas and the spreading of virtual resources atop providers, regions, and zones. Replicas are used to achieve an user-defined reliability level while the controlled spreading of VI components decrease the probability of full outages. In addition, the proposal performs a cost-effective provisioning. We formulate the VI allocation, survivability, and reliability requirements as a Mixed Integer Program (MIP). Simulation results using different target reliability levels shows an increase on survivability without inflating costs.*

1. Introdução

Os provedores de nuvem IaaS oferecem IVs seguindo o modelo *pay-as-you-go*, no qual os clientes são cobrados de acordo com os recursos alocados, estes vinculados a Máquinas

Virtuais (MVs) e a enlaces de rede [Mell and Grance 2011]. Atualmente, associado a este modelo, existem serviços que são oferecidos em IVs e, portanto, suscetíveis a eventuais indisponibilidades. Os índices de disponibilidade fornecidos por provedores são divulgados e estabelecidos em contratos, através do Acordo de Nível de Serviço (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 da 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]. Nestes casos, a política, normalmente aplicada, é a recompensa das horas de indisponibilidade em créditos para locação e reinício das IVs. 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 *survivability* são características essenciais pois IVs são dependentes de DCs e quedas não planejadas são comuns nestas infraestruturas [Govindan et al. 2016]. Por um lado, a aplicação crítica precisa de confiabilidade, a qual leva em consideração a probabilidade de uma IV estar operando corretamente. Por outro, precisa da *survivability*, que indica a habilidade da IV permanecer operacional na ocorrência de uma falha no DC.

O nível de confiabilidade de uma IV pode ser melhorado através da utilização de réplicas [Yeow et al. 2011]. A réplica consiste na alocação excedente de recursos para serviços que já encontram-se em operação. Uma réplica não substitui o serviço existente, mas fica pronta para entrar em operação em caso de falha ou lentidão do serviço principal. Dispor de réplicas sempre implica em aumento de custo, o que nem sempre é compatível com o orçamento. 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 minimizando o custo, pode-se restringir a associação de réplicas a apenas instâncias críticas [Koslovski et al. 2010]. Assim, o serviço passa a ser resiliente à falha de instâncias críticas, mas permanece vulnerável a indisponibilidade de provedores. O grau de *survivability* de uma IV pode ser incrementado através do espalhamento de MVs sobre múltiplos domínios de falha, diminuindo a probabilidade de uma falha total [Bodík et al. 2012, Cavalcanti et al. 2014]. Finalmente, a aplicação combinada destas abordagens geram várias possibilidades e, portanto, demandam uma análise complexa dos recursos solicitados pela IVs e dos dados de múltiplos provedores, regiões e zonas.

Esta proposta consiste na concepção de um corretor (*broker*) de nuvem IaaS que incrementa o grau de confiabilidade e de *survivability* minimizando o custo. O corretor de nuvem proposto, guiado pela perspectiva do cliente, é agnóstico aos requisitos da aplicação hospedada e de mecanismos de alta disponibilidade internos (*e.g.*, replicação ativa ou passiva). O restante do artigo é organizado da seguinte maneira: a Seção 2 discorre sobre a motivação e os desafios da alocação de IV sobre múltiplos provedores IaaS e a formulação do problema. A Seção 3 detalha o MIP proposto, enquanto os resultados das simulações são apresentados na Seção 4. A Seção 5 revisa os trabalhos relacionados.

2. Motivação e Formulação do Problema

Uma Infraestrutura Virtual (IV) é constituída por um conjunto de máquinas virtuais (MV) associadas a recursos virtuais de redes [Sotomayor et al. 2009]. Em nu-

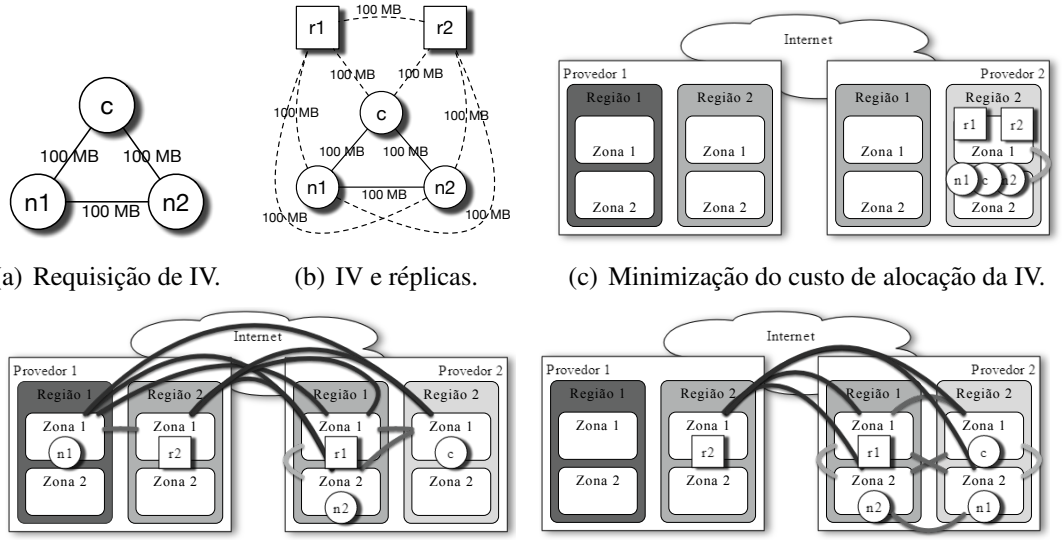
vens IaaS, os clientes escolhem seus provedores, especificam os recursos virtuais requeridos e geram solicitações de IVs a provedores. Os recursos contratados e as condições de fornecimento desses recursos são estabelecidos no SLA. Normalmente, clientes têm seu foco na aplicação, ressaltando aspectos relacionados ao custo e a QoS [Rosenberg and Mateos 2010]. Em nuvem IaaS, o custo pode ser minimizado com a redefinição do SLA de acordo com a demanda, evitando custos de sobre-provisionamento.

No presente trabalho, destacam-se dois aspectos essenciais no estabelecimento do SLA para IVs que hospedam aplicações críticas: confiabilidade e *survivability*. Embora essenciais para os clientes, provedores não fornecem estatísticas sobre o grau de confiabilidade e *survivability* dos serviços hospedados em seus DCs. A única informação pública é o tempo de indisponibilidade do DC que, em alguns casos, já causaram perdas milionárias [Avram 2011]. Como os clientes não possuem acesso direto ao DC, a única alternativa para minimizar o impacto das falhas é contratar serviços adicionais dos próprios provedores ou implementar soluções em nível de aplicação [Rajagopalan et al. 2012, Cavalcanti et al. 2014]. Enfim, clientes preocupam-se com a aplicação e ignoram os aspectos técnicos vinculados à gerência de IVs ou critérios de seleção de provedores.

Do lado do provedor, gerentes de recursos apoiam-se em algoritmos *on-line* para associar os recursos virtuais contratados aos recursos físicos existentes [Fischer et al. 2013, Souza et al. 2017, Oliveira and Koslovski 2017, Houidi et al. 2011]. Provedores buscam, em sua maioria, maximizar o lucro diminuindo custos e aumentando a QoS entregue aos clientes [Fischer et al. 2013, Chowdhury and Boutaba 2010]. Como parte da função do provedor é gerenciar IVs, é natural que os mecanismos de provisionamento sejam orientados pela visão do próprio provedor. Portanto, de um lado, clientes não tem conhecimento técnico nem informações para escolher o provedor mais adequado às suas necessidades e, por outro, os mecanismos de provisionamento são orientados pela visão do provedor. Neste contexto, torna-se essencial a atuação de corretores de nuvem que buscam associar as necessidades dos clientes a serviços fornecidos por provedores públicos. Este trabalho propõe um corretor de nuvem para alocar IVs sobre múltiplos provedores IaaS, cujo foco é ser economicamente viável, confiável e *survivable*.

2.1. Requisições de IV e Provedores IaaS

A alocação de IV, com o auxílio do corretor proposto, exige que o cliente identifique os componentes críticos e o grau de confiabilidade alvo [Yeow et al. 2011, Koslovski et al. 2010]. Uma IV passa a ser constituída de MVs regulares e críticas. O cliente tem ciência que somente as MVs críticas possuem réplicas. Formalmente, uma requisição de IV é representada por $VI(N, D, V, c)$, sendo N o conjunto de MVs, $D \subset N$ representa as MVs críticas, e V representa os enlaces virtuais entre MVs (cada enlace com uma requisição de transferência de dados, denotada em $v_{nm} \in N$). A confiabilidade alvo é dada por c ($99,995\%$, por exemplo). Por exemplo, uma IV sem réplicas é representada na Figura 1(a). Para esta mesma IV, o cliente deseja que o nível de confiabilidade atinja $c = 99,995\%$. Portanto, réplicas das MVs críticas e novos enlaces virtuais são acrescentados (Figura 1(b)). Neste exemplo, as requisições de transferência de dados são definidas como 100MB.



(a) Requisição de IV. (b) IV e réplicas. (c) Minimização do custo de alocação da IV.

(d) Espalhamento dos recursos da IV sobre provedores. (e) Alocação visando menor custo, confiabilidade e *survivability*.

Figura 1. Alocação de IV com confiabilidade alvo $c = 99,995\%$ e 3 grupos de falha (provedores, regiões e zonas). Quanto mais clara a cor, menor o custo.

A requisição da IV 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 1 resume a notação utilizada no trabalho.

Notação	Descrição
$P(R, Z)$	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.
$VI(N, D, V, c)$	Uma IV composta de N MVs, $D \subset N$ MVs críticas, uma matriz de tráfego (V) e o nível de confiabilidade alvo (c).
$n \in N$	Uma MV regular n.
$m \in D \subset N$	Uma MV crítica m.
$l_{nm} \in V$	Enlace virtual entre MV n e m. Cada enlace solicita um volume de dados a ser transferido v_{nm}.
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 MVs críticas no provedor i, região j e zona k.
$C(i, j, n)$	Custo de alocação da MV 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}	MV 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 MVs hospedadas pelo provedor i (inteiro).
y_{ij}^r	Número de MVs hospedadas pelo provedor i e região j (inteiro).
y_{ijk}^z	Número de MVs hospedadas pelo provedor i, região j e zona k (inteiro).

Tabela 1. Notação para representar requisições de IVs, provedores IaaS, e detalhes fundamentais do modelo.

2.2. Probabilidade de Falha

Há normalmente uma sequência de eventos que resulta em falhas. Inicialmente, um defeito causa um erro que é propagado até uma falha [Ucla et al. 2001]. A falha de um subsistema pode causar um defeito em outro sistema que interage com ele, seguindo a cadeia de propagação. Tais falhas podem acontecer em servidores e recursos de redes (*e.g.*, switches e roteadores). *Logs* e dados de DCs são úteis para entender e reduzir a probabilidade de novas falhas acontecerem [Govindan et al. 2016]. Contudo, os dados brutos são contabilizados de forma privada e mantidos confidencialmente. Clientes de nuvem são somente informados sobre os dados de disponibilidade, especificados durante o estabelecimento do SLA. Informações precisas como Tempo Médio entre Falhas (MTBF), Tempo Médio para Recuperação (MTTR) e Tempo Médio entre Quedas (MTBO) são normalmente mantidas em sigilo.

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 IV, 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 tem-se que $\frac{720-\sum \text{duração da queda}}{\text{número de quedas}}$, sendo as durações dadas em horas, conforme normalmente utilizado por provedores IaaS públicos. Seguindo o raciocínio, a probabilidade de falhas (p) é dada por $\frac{1}{\text{MTBF}}$, e finalmente, a confiabilidade é definida por $1 - p$.

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 CloudHarmony[1]. 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 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 MVs e réplicas sobre diferentes grupos de falhas ajuda a diminuir a probabilidade de falha total, aumentando consequentemente a *survivability* da IV [Bodík et al. 2012, Cavalcanti et al. 2014].

2.3. Definição de Réplicas para MVs 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 IV [Rajagopalan et al. 2012,

[1]CloudHamony: `https://cloudharmony.com/`

Koslovski et al. 2010]. Inicialmente, as MVs 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 Agrupamento Oportunista de Réplicas (AOR) [Yeow et al. 2011]. O AOR utiliza a função beta incompleta regularizada, $I_{1-p} = (n, k+1)$, sendo n o número de MVs 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 MVs críticas é calculado para cada zona e representado por M. A Figura 2 exemplifica a composição de M.

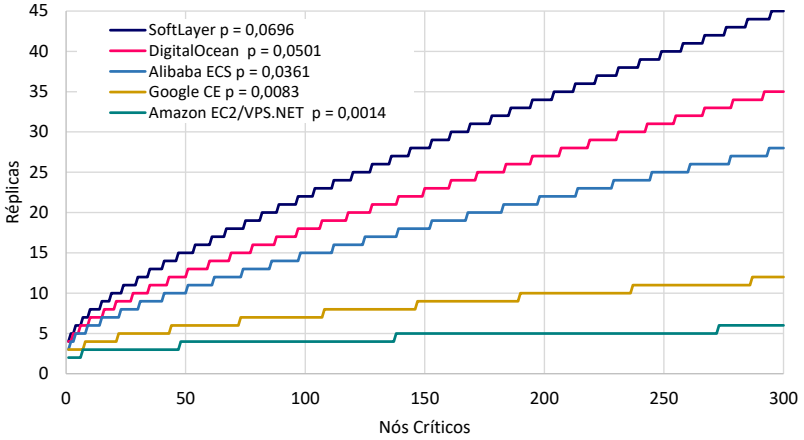


Figura 2. Número de réplicas para suportar $c = 99,995\%$ variando p.

Utilizando o AOR, um intervalo de nós críticos suportados por k réplicas é definido. Dado o número de MVs 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 MVs críticas. O conjunto M tem a composição guiada pelo CloudHarmony e indexada por provedor, região, zona, confiabilidade alvo c, e número de MVs críticas.

A Figura 1(b) exemplifica a extensão de uma requisição de IV (Figura 1(a)) adicionando réplicas e enlaces. Para esse exemplo, 2 réplicas são arbitrariamente adicionadas ($r1$ e $r2$) 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.

2.4. Alocação de Provedores IaaS para Hospedar IVs

As requisições de IVs confiáveis são individualmente analisadas pelo mecanismo, caracterizando um problema de alocação online [Fischer et al. 2013]. O mapeamento de MVs em zonas é dado por $\mathcal{M} : N \mapsto Z$. 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 et al. 2011, Oliveira and 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 *survivability* e confiabilidade. A primeira medida visa minimizar o custo de provisionamento de MVs (regulares, críticas e réplicas) e rede (transferência de dados entre MVs), enquanto o segundo objetiva minimizar o impacto de falhas de provedores na IV [Cavalcanti et al. 2014, Bodík et al. 2012].

Três exemplos de alocação de IV são apresentadas na Figura 1. 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 1(c) 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 MVs, 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 1(d). As MVs 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 IV é minimizada para todos os grupos de falha (provedores, regiões e zonas).

O foco do nosso trabalho é identificar uma abordagem intermediária, como a dada pela Figura 1(e). 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.

3. MIP Exato para Alocação *Survivable* e Confiável de IVs

3.1. Variáveis e Objetivos

Quatro variáveis são usadas para identificar quais provedores podem hospedar uma requisição de IV, como exemplificado na Figura 1(b). Inicialmente, x_{nijk}, uma variável binária, indica o mapeamento de MVs 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 MVs críticas, sendo tal informação desconhecida inicialmente. Na perspectiva da *survivability*, B representa o cenário com o pior caso, ou seja, no qual a zona selecionada para hospedar as MVs 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 MVs, duas variáveis são usadas, xl e bl. O primeiro representa a alocação de um enlace virtual l_{mn} entre as MVs 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 MVs 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 (MVs 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.

3.1.1. Custo de Alocação da IV

Provedores IaaS aplicam diferentes modelos de custos para MVs, normalmente diferenciando por regiões. Nesse sentido, a função $C(i, j, n)$ retorna o custo para hospedar a MV n no provedor i, região j, e as Eqs. (1) e (2) contabilizam o custo para hospedar todas as MVs e as réplicas definidas dinamicamente, respectivamente.

$$C_{vm}(VI) = \sum_{n \in N} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} x_{nijk} \times C(i, j, n) \tag{1}$$

$$C_{vmb}(VI) = \sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \times C(i, j, w) \tag{2}$$

Os custos para transferência de dados entre MVs são dados pela Eq. (3) (regulares e críticas) e Eq. (4) (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).

$$C_{net}(VI) = \sum_{l_{nm} \in V} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} xl_{nmzk} \times v_{nm} \times C_v(z, k) \tag{3}$$

$$\begin{aligned} C_{netb}(VI) = & \sum_{l_{nm} \in N \times B} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} bl_{nmzk} \times v_{nm} \times C_v(z, k) + \\ & \sum_{l_{nm} \in B \times B} \sum_{i_s \in P} \sum_{j_s \in R_{i_s}} \sum_{z \in Z_{i_s j_s}} \sum_{i_t \in P} \sum_{j_t \in R_{i_t}} \sum_{k \in Z_{i_t j_t}} bl_{nmzk} \times v_{nm} \times C_v(z, k) \end{aligned} \tag{4}$$

Por fim, o custo total de alocação de uma IV é dado pela Eq. (5). Um vetor de pesos (β) é usado para definir o nível de importância de cada componente.

$$\begin{aligned} C_{total}(VI) = \beta_{vm} \times C_{vm}(VI) + \\ \beta_{vmb} \times C_{vmb}(VI) + \beta_{net} \times C_{net}(VI) + \beta_{netb} \times C_{netb}(VI) \end{aligned} \tag{5}$$

3.1.2. Impacto de Falhas em Provedores IaaS

Uma abordagem intuitiva para diminuir o impacto de falhas em IVs é o espalhamento de recursos virtuais sobre múltiplos domínios [Bodík et al. 2012, Rajagopalan et al. 2012, Cavalcanti et al. 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 IV. 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 (6)-(8) contabilizam o número de MVs hospedadas por provedores, regiões e zonas.

$$y_i^p = \sum_{j \in R_i} \sum_{k \in Z_{ij}} \left(\sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \right); \forall i \in P \tag{6}$$

$$y_{ij}^r = \sum_{k \in Z_{ij}} \left(\sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk} \right); \forall i \in P; \forall j \in R_i \tag{7}$$

$$y_{ijk}^z = \sum_{n \in N} x_{nijk} + \sum_{w \in B} b_{wijk}; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \tag{8}$$

Para espalhar as MVs e réplicas sobre os grupos de falhas, 3 variáveis inteiras e positivas (Eqs. (9)-(11)) compõem a minimização da função de impacto de falhas (min Eq. (15)). As variáveis maximizam a distribuição sobre os grupos de falhas respeitando o número de MVs e réplicas (Eqs. (12)-(14)). Um vetor (γ) diferencia a importância de cada componente.

$$I^p \geq y_i^p; \forall i \in P \tag{9}$$

$$I^r \geq y_{ij}^r; \forall i \in P; \forall j \in R_i \tag{10}$$

$$I^z \geq y_{ijk}^z; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \tag{11}$$

$$I^p \leq |N| + |B|; \forall i \in P \tag{12}$$

$$I^r \leq |N| + |B|; \forall i \in P; \forall j \in R_i \tag{13}$$

$$I^z \leq |N| + |B|; \forall i \in P; \forall j \in R_i; \forall k \in Z_{ij} \tag{14}$$

$$I(VI) = \gamma_p \times I^p + \gamma_r \times I^r + \gamma_z \times I^z \tag{15}$$

3.1.3. Função Objetivo

A minimização da Equação (16) resulta na alocação com o menor custo, além de diminuir 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 MVs e réplicas. O peso α, definido entre 0 e 1, permite balancear a equação, priorizando custo ou espalhamento, de acordo com a necessidade da alocação.

$$min : \left(\alpha \times \frac{C_{total}(VI)}{C_{max}(VI)} \right) + \left((1 - \alpha) \times \frac{I(VI)}{|N| + |B|} \right) \tag{16}$$

3.2. Restrições

Para garantir o QoS do SLA, um conjunto restrições devem ser satisfeitas (Eqs. (17)-(26)). As Restrições (17) e (18) indicam que MVs e réplicas, respectivamente, devem ser alocadas até uma vez. A respeito das réplicas, o número mínimo indicado pelo AOR é

garantido pela Eq. (19), enquanto o limite superior é a alocação sobre a zona com a maior probabilidade de falha, como indicado pela Eq. (20). As restrições (21)-(23) 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, Eqs. (24)-(26) garantem que os enlaces virtuais são hospedados uma vez, no máximo.

$$\sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} x_{nijk} = 1; \forall n \in N \tag{17}$$

$$\sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{nijk} \leq 1; \forall n \in B \tag{18}$$

$$\sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \geq min(M) \tag{19}$$

$$\sum_{w \in B} \sum_{i \in P} \sum_{j \in R_i} \sum_{k \in Z_{ij}} b_{wijk} \leq |B| \tag{20}$$

$$\sum_{q \in Z_{st}} xl_{nmkq} + \sum_{q \in Z_{st}} xl_{nmzq} = x_{nijk} + x_{mijk}$$
$$i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in V \tag{21}$$

$$\sum_{q \in Z_{st}} bl_{nmkq} + \sum_{q \in Z_{st}} bl_{nmzq} = x_{nijk} + b_{mijk}$$
$$i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in N \times B \tag{22}$$

$$\sum_{q \in Z_{st}} bl_{nmkq} + \sum_{q \in Z_{st}} bl_{nmzq} = b_{nijk} + b_{mijk}$$
$$i \in P, j \in R_i, k \in Z_{ij}, s \in P, t \in R_s, l_{nm} \in B \times B \tag{23}$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} xl_{nmkq} = 1 i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in V \tag{24}$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} bl_{nmkq} \leq 1 i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in N \times B \tag{25}$$

$$\sum_{k \in Z_{ij}} \sum_{q \in Z_{st}} bl_{nmkq} \leq 1 i \in P, j \in R_i, s \in P, t \in R_s, l_{nm} \in B \times B \tag{26}$$

4. Simulação e Análise

Para analisar a eficiência do MIP proposto, um corretor de nuvem foi implementado em Java v1.8, usando IBM CPLEX optimizer (v12.6.1.0)[2]. A simulação foi realizada em um sistema com GNU/Linux Ubuntu 14.04, processador AMD Phenom II X4, 4 GB RAM.

4.1. Métricas

Com o objetivo de representar a perspectiva do cliente, 7 métricas foram selecionadas. *(i) Custos de MVs regulares e críticas. (ii) Custo de réplicas. (iii) Custo de comunicação*: o custo para transferir dados entre MVs, definido pela Eq (3). *(iv) Custo de comunicação entre réplicas*: o custo para transferir dados entre as MVs e réplicas, definido pela Eq. (4).

[2]`https://www.ibm.com/software/commerce/optimization/cplex-optimizer/`

(v)-(vii) Utilização dos grupos de falhas: o espalhamento nos zonas, regiões e provedores, respectivamente. Todas as métricas são normalizadas para a composição dos gráficos. As medidas de custo são normalizadas pelo custo máximo, enquanto os grupos de falhas (zonas, regiões e provedores) são normalizados pelo espalhamento máximo possível, dado por $min(|N| + |B|, G)$ sendo G o tamanho máximo do grupo analisado. 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.

4.2. Cenários de Simulação

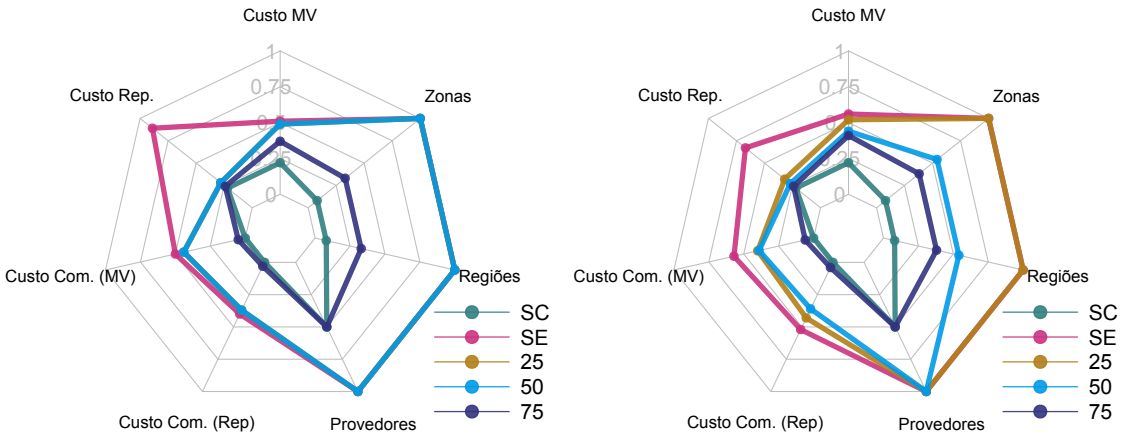
A probabilidade de falhas para cada zona foi extraída da plataforma CloudHarmony, especificamente os dados de Agosto/2017. Para compor a função do custo de transferência de dados, os preços foram selecionados uniformemente em 3 intervalos: *(i)* transferência de dados entre zonas na mesma região: entre $\$0,01$ 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 MVs 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 IV foi escolhida a instância popular da Amazon EC2, $m3.large$ [Persico et al. 2015]. Para definir a função de custo $C(i, r, n)$, uma configuração similar foi selecionada para cada provedor de nuvem. As MVs 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.

O cenário tem como objetivo analisar a eficiência do MIP. A simulação utiliza as informação 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 MVs regulares e 5 MVs 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 2.3).

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 (Fig. 1(c)). 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 (Fig. 1(d)). A alocação exata combina os objetivos de melhor custo e maior espalhamento ($\alpha = 0,5$), provendo uma maior confiabilidade e *survivability* (Fig. 1(e)). Para aprofundar a discussão, outras 2 alocações com diferentes valores para α ($0,25$ e $0,75$) foram executadas, resultando em diferentes composições para (Fig. 1(e)).

4.3. Resultados da Simulação

Os resultados das simulações são exibidos na Figura 3. A Figura 3(a) mostra a alocação para a confiabilidade alvo de $99,95\%$, enquanto a Figura 3(b) para $c = 99,995\%$. Como esperado, a abordagem SC possui a menor área, priorizando a concentração das MVs 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 o custo de comunicação.



(a) 99.95% (b) 99.995%

Figura 3. Resultados da alocação.

Nessa abordagem, o espalhamento da IV é realizado sem considerar o custo de provisionamento. Todos os provedores, regiões e zonas são candidatos para receber a MV, e certamente zonas mais caras são selecionadas.

A alocação exata se beneficia das vantagens de ambas as abordagens. Na Figura 3(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 3(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 *survivability*. Analisando tais resultados fica evidente a eficiência da alocação exata.

5. Trabalhos Relacionados

A literatura especializada, relacionada ao presente trabalho, compreende duas linhas principais: a alocação de recursos físicos para hospedar IVs; e a definição de técnicas que melhorem a *survivability* e confiabilidade de recursos virtuais.

5.1. Alocação de Recursos Físicos para Hospedar IV

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 [Houidi et al. 2011]. 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 *et al.* (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 proposta pode ser aplicada em conjunto com nossa abordagem para melhorar a seleção de provedores.

A respeito da alocação de IVs em um único DC, algumas técnicas para minimizar o consumo de largura de banda combinado com suporte a privacidade foi proposto em [Bays et al. 2016]. Em [Oliveira and Koslovski 2017], uma heurística baseada em árvores foi proposta para diminuir o tempo de alocação da IV. Contudo, a heurística tende

a agrupar recursos virtuais, aumentando assim o impacto de uma eventual falha. Apesar de não ter como objetivo uma alocação *survivable*, um espalhamento controlado de recursos virtuais sobre um DC foi aplicada em [Souza et al. 2017], entretanto total conhecimento e controle no DC são necessários.

Resumindo, a literatura sobre alocação de IVs 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.

5.2. Técnicas para Provisionamento *Survivable* de IVs

O provisionamento *survivable* de IVs foi proposto em [Koslovski et al. 2010]. Similarmente ao presente trabalho, o mecanismo utilizou o AOR 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. Nossa abordagem inova por definir juntamente réplicas e o espalhamento de MVs em múltiplos provedores, regiões e zonas.

A técnica AOR é também aplicada para alocação de redes virtuais [Yeow et al. 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 *survivable*. Cabe ressaltar que a presente proposta combinou o menor custo com uma alocação *survivable* e confiável em múltipos provedores de nuvem, preenchendo uma lacuna de pesquisa no que diz respeito à perspectiva do cliente.

6. Considerações & Trabalhos Futuros

O trabalho apresenta uma alternativa para aumentar a *survivability* de uma IV, garantindo a confiabilidade alvo através de réplicas, sem aumentar significativamente o custo de alocação da IV. Para isso, um MIP foi formulado para definir a alocação exata da IV sobre múltiplos provedores. Os resultados mostram que a solução é eficiente em termos de confiabilidade e *survivability*, sem aumentar o custo de provisionamento. O custo total permanece o mais próximo possível do mínimo, respeitando a confiabilidade alvo. Como resolver um MIP em larga escala é computacionalmente intratável, trabalhos futuros visam relaxar as variáveis inteiras e definir uma heurística para aproximação. Uma segunda linha indica a implementação da solução proposta como um serviço público de nuvem.

Agradecimentos: Ao LabP2D e UDESC. Às instituições de apoio: FAPESC e CAPES.

Referências

Amazon EC2 (2017). Amazon Web Services summary of the amazon s3 service disruption in the northern virginia (us-east-1) region. Technical report, Amazon.

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., and Zaharia, M. (2010). A View of Cloud Computing. *Commun. ACM*, 53(4):50–58.

Avram, A. (2011). Amazon EC2 Outage Explained and Lessons Learned. Technical report, InfoQ.

Bays, L. R., Oliveira, R. R., Buriol, L. S., Barcellos, M., and Gaspary, L. P. (2016). A toolset for efficient privacy-oriented virtual network embedding and its instantiation on SDN/OpenFlow-based substrates. *Computer Communications*, 82:13–27.

Bodík, P., Menache, I., Chowdhury, M., Mani, P., Maltz, D. A., and Stoica, I. (2012). Surviving failures in bandwidth-constrained datacenters. In *Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '12, pages 431–442, New York, NY, USA. ACM.

Caron, E. and de Assunção, M. D. (2016). Multi-criteria malleable task management for hybrid-cloud platforms. In *2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech)*, pages 326–333, Marrakech, Morocco. IEEE.

Cavalcanti, G. A. S., Obelheiro, R. R., and Koslovski, G. (2014). Optimal resource allocation for survivable virtual infrastructures. In *Design of Reliable Communication Networks (DRCN), 2014 10th International Conference on the*, pages 1–8, Ghent, Belgium. IEEE.

Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54(5):862–876.

Fischer, A., Botero, J. F., Beck, M. T., De Meer, H., and Hesselbach, X. (2013). Virtual network embedding: A survey. *IEEE Communications Surveys & Tutorials*, 15(4):1888–1906.

Govindan, R., Minei, I., Kallahalla, M., Koley, B., and Vahdat, A. (2016). Evolve or die: High-availability design principles drawn from googles network infrastructure. In *Proc. of the 2016 ACM SIGCOMM conference*, pages 58–72, Florianopolis, Brazil. ACM, ACM.

Houidi, I., Louati, W., Ameur, W. B., and Zeghlache, D. (2011). Virtual network provisioning across multiple substrate networks. *Computer Networks*, 55(4):1011–1023.

Koslovski, G., Yeow, W. L., Westphal, C., Huu, T. T., Montagnat, J., and Vicat-Blanc, P. (2010). Reliability support in virtual infrastructures. In *2010 IEEE Second International Conference on Cloud Computing Technology and Science*, pages 49–58, Indianapolis, USA. IEEE.

Mell, P. M. and Grance, T. (2011). SP 800-145. the NIST definition of cloud computing. Technical report, National Institute of Standards & Technology.

Oliveira, R. d. and Koslovski, G. P. (2017). A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. *International Journal of Network Management*, 27(1):e1958.

Persico, V., Marchetta, P., Botta, A., and Pescapè, A. (2015). Measuring network throughput in the cloud. *Comput. Netw.*, 93(P3):408–422.

Rajagopalan, S., Cully, B., O'Connor, R., and Warfield, A. (2012). Secondsite: Disaster tolerance as a service. *SIGPLAN Not.*, 47(7):97–108.

Rosenberg, J. and Mateos, A. (2010). *The Cloud at Your Service*. Manning Publications Co., 1st edition.

Sotomayor, B., Montero, R. S., Llorente, I. M., and Foster, I. (2009). Virtual infrastructure management in private and hybrid clouds. *IEEE Internet computing*, 13(5):14–22.

Souza, F. R. d., Miers, C. C., Fiorese, A., and Koslovski, G. P. (2017). QoS-Aware Virtual Infrastructures Allocation on SDN-based Clouds. In *Proceedings of the 2017 IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, CCGRID '17, Madrid, Spain. IEEE Computer Society.

Ucla, A. A., Avizienis, A., claude Laprie, J., and Randell, B. (2001). *Fundamental concepts of dependability*. University of Newcastle upon Tyne, Computing Science, Newcastle, U.K.

Yeow, W.-L., Westphal, C., and Kozat, U. C. (2011). Designing and embedding reliable virtual infrastructures. *ACM SIGCOMM Computer Communication Review*, 41(2):57–64.