# Comparação de algoritmos para alocação de infraestruturas virtuais

Denivy Braiam Ruck <sup>1</sup> Ramon de Oliveira <sup>1</sup> Guilherme Piegas Koslovski <sup>1</sup>

Resumo: Com o surgimento da Computação em Nuvem, empresas passaram a adotá-la em virtude das diversas vantagens que apresenta, como por exemplo, baixo investimento inicial e alta escalabilidade. Nesse contexto, um provedor de *Infrastructure as a Service* (IaaS) oferece recursos computacionais a seus clientes de forma virtualizada, que, por sua vez, podem realizar a solicitação de máquinas virtuais para implantar e disponibilizar suas aplicações. Uma proposta recente incrementou os serviços do provedor IaaS, possibilitando alocar uma rede de máquinas e roteadores virtuais interconectados por canais virtuais, ou seja, uma Infraestrutura Virtual (IV). Porém, a tarefa de alocar IVs sobre o substrato físico é complexa, caracterizada como um problema NP-Díficil, no qual algoritmos ótimos são impraticáveis em cenários reais (o tempo para encontrar uma solução ótima é exponencial). Tendo em vista esse fato, este trabalho estende o simulador CloudSim, implementando quatro algoritmos *on-line* para a alocação de recursos, comparando as métricas: taxa de aceitação, utilização do substrato computacional e distância entre os recursos virtuais. Tais métricas representam as perspectivas dos provedores e dos usuários requisitantes.

Palavras-chave: Alocação. CloudSim. IaaS. Infraestruturas virtuais. Nuvens computacionais.

**Abstract:** With the emergence of Cloud Computing, companies began to adopt it due to its several advantages, such as low initial investment and high scalability. In this context, a provider of *Infrastructure as a Service* (IaaS) provides computer resources to its clients in a virtualized way. Users can perform requests of virtual machines to deploy and provide their applications. A recent proposal has emerged to improve the services of IaaS providers, enabling the allocation of a network of virtual machines and routers interconnected by virtual channels. In other words, a Virtual Infrastructure (VI). However, the task of allocating VIs on the physical substrate is complex, characterized as an NP-Hard problem, where optimal algorithms are impractical in real scenarios due to its exponential execution time. Considering this fact, this work extends the simulator CloudSim implementing four on-line algorithms to allocate VIs, and compare metrics such as acceptance ratio, substrate load and the distance between the virtual resources. These metrics represent the perspectives of providers and users, respectively.

Keywords: Allocation. CloudSim. Cloud computing. IaaS. Virtual infrastructures.

### 1 Introdução

O conceito de Nuvens Computacionais propõe a entrega de recursos (aplicações, plataformas, infraestruturas, entre outros) como serviços, que podem ser dinamicamente instanciados por múltiplos usuários. Essa proposta visa a diminuição dos custos com gerenciamento e manutenção, bem como o aumento da escalabilidade de aplicações, resultado de uma representação de recursos infinitos que podem ser reservados sob demanda. Dentre os vários modelos de serviço existentes, este trabalho aborda especificamente Infraestruturas como Serviço (*Infrastructures as a Service - IaaS*), que é a entrega de recursos computacionais virtualizados (servidores, armazenamento e rede) sob demanda [1, 2]. Para que um provedor de nuvem tenha consistência sobre seus recursos é preciso abordar

{denivy.ruck@joinville.udesc.br, ramon.oliveira@joinville.udesc.br, guilherme.koslovski@udesc.br}

http://dx.doi.org/10.5335/rbca.2014.4075

<sup>&</sup>lt;sup>1</sup>Programa de Pós-Graduação em Computação Aplicada (PPGCA), Departamento de Ciência da Computação (DCC), Universidade do Estado de Santa Catarina, Joinville - Brasil

os desafios de gerenciamento de sistemas distribuídos: heterogeneidade, abertura, segurança, escalabilidade, manipulação de falhas, concorrência e transparência [3], conjuntamente explorados com a virtualização de recursos computacionais.

Por meio da virtualização, nuvens computacionais oferecem uma abstração em relação aos recursos físicos utilizados. Essa abstração é implementada a partir de tecnologias que permitem a desmaterialização da capacidade física de uma máquina e sua representação por intermédio de entidades e serviços virtuais. Um equipamento computacional virtualizado pode hospedar diversas máquinas virtuais (MVs) simultaneamente, compartilhando seus recursos de processamento, armazenamento e comunicação [4]. Esse compartilhamento e consolidação de recursos proporciona uma redução de custos com consumo de energia, equipamentos de refrigeração, administração e gerenciamento e, ao mesmo tempo, aumenta a possibilidade de exploração e utilização do substrato computacional. Recentemente, o conceito de virtualização de redes de computadores introduziu uma divisão das funcionalidades tradicionalmente desempenhadas de forma integrada pelos Provedores de Serviço Internet (ISPs): em um ambiente virtualizado, tem-se Provedores de Infraestrutura (InPs), responsáveis pela infraestrutura física; e Provedores de Serviço (SPs), responsáveis por criar redes virtuais, agregando recursos dos InPs [5]. O conceito de Infraestruturas Virtuais (IVs) estendeu a entrega de IaaS, incluindo o provisionamento dinâmico de recursos virtuais de comunicação [6]. Esse conceito define que os recursos de rede sejam disponibilizados da mesma forma que os demais serviços em nuvens, e que sejam tarifados de acordo com a capacidade reservada e utilizada. Essa combinação de diversos recursos virtuais disponibilizados para um usuário, durante um determinado período, possibilita a criação de uma infraestrutura completamente virtual com as mesmas características de uma infraestrutura física.

Um dos principais desafios quanto ao gerenciamento de nuvens está relacionado à seleção de recursos físicos para hospedar os recursos virtuais. Diretamente relacionado com a implementação da elasticidade em nuvens [7], o mapeamento e a alocação de recursos é sabido ser um problema NP-Difícil [8]. Usualmente, *frameworks* de gerenciamento consideram os objetivos dos InPs, orientados pelas suas perspectivas financeiras [9]. InPs objetivam maximizar suas receitas alocando o maior número possível de IVs, usando para isso o mínimo de infraestrutura computacional que for viável. O InP é o responsável por realizar a alocação e pode expor uma interface e parâmetros da alocação aos PSs.

Com a decisão sobre quais IVs serão alocadas e quais recursos serão reservados, o InP pode determinar quais tipos de alocações estão de acordo com o seu modelo de negócios. Um InP pode priorizar a alocação da IV de um cliente em detrimento da IV de um outro cliente, ou alocar IVs em recursos dedicados para clientes que tenham requisitos de acesso e violação dos seus dados sensíveis, ou, ainda, alocar IVs de acordo com modelos de custos, segurança, confiabilidade e tolerância a faltas, entre outros. Essa decisão do InP a respeito da alocação é normalmente traduzida como uma função objetivo, e um algoritmo de alocação busca minimizar ou maximizar essa função. Com isso, uma alocação pode ocorrer de duas maneiras distintas: otimização ou heurística. Uma otimização tenta encontrar um valor ótimo para a função objetivo definida pelo InP, e o tempo para encontrar esse valor é exponencial. Programação linear, inteira ou mista, e programação não linear são alguns exemplos de algoritmos que encontram valores ótimos. Uma heurística permite ao InP encontrar um valor subótimo a favor de um tempo linear para encontrar esse valor. Para que a alocação seja efetuada, é necessário que o InP especifique as restrições e os recursos de sua infraestrutura, nas quais as restrições serão aplicadas. Por exemplo, uma restrição pode impedir que um recurso físico hospede mais recursos virtuais além da sua capacidade, ou que um caminho virtual conecte dois nós virtuais e que seja conexo, ou outras restrições que imponham condições de segurança ou outras previstas no seu modelo de negócio.

O foco deste trabalho está justamente na comparação de algumas heurísticas para alocação de recursos para hospedar infraestruturas virtuais. Nossa proposta é baseada na simulação de um ambiente de nuvens. Com a simulação, é possível abstrair a complexidade existente no gerenciamento dos recursos distribuídos e analisar as heurísticas com diversas configurações de provedores em um ambiente controlável. Para tal, o simulador Cloud-Sim [10] foi estendido, implementando quatro algoritmos para a alocação de IVs. Os algoritmos representam (separadamente) a visão dos InPs e dos usuários solicitantes de IVs, sendo que os resultados experimentais indicam a existência de um *tradeoff* entre as duas visões. Sobretudo, a extensão realizada no simulador pode ser facilmente adaptável para implementar outras abordagens de alocação.

Este artigo está organizado da seguinte forma: a seção 2 descreve quais recursos podem ser solicitados para alocação (ou seja, o conceito de Infraestruturas Virtuais). Por sua vez, a seção 3 apresenta os algoritmos em detalhes, enquanto a seção 4 descreve o simulador CloudSim e a extensão realizada para implementar a alocação

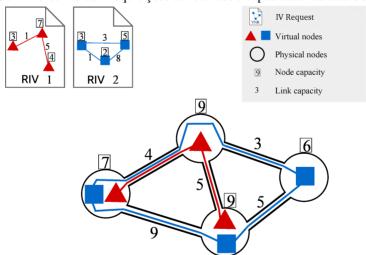
de IVs. Os resultados da simulação são apresentados na seção 5. A seção 6 revisa os trabalhos relacionados, e, por fim, a seção 7 conclui o artigo, apresentando perspectivas para trabalhos futuros.

### 2 Alocando recursos para hospedar Infraestruturas Virtuais

Em Nuvens Computacionais, o usuário não tem acesso nem conhecimento da distribuição física de qualquer dos recursos para ele disponibilizado, confiando as tarefas de gerenciamento ao provedor. Um dos principais problemas que um provedor deve lidar ao oferecer IVs é como realizar o mapeamento dos recursos virtuais sobre os recursos físicos. Assim, como uma Infraestrutura Virtual pode ser vista como uma extensão do paradigma de IaaS, a alocação de recursos para hospedar IVs pode ser vista como uma extensão do problema de mapeamento de redes virtuais [8]. O principal diferencial é que, neste, as extremidades (máquinas virtuais e roteadores virtuais) não são conhecidos antecipadamente e sua alocação deve ser definida juntamente com a definição dos canais virtuais. Ou seja, além de alocar os canais de comunicação, as soluções de gerenciamento devem mapear os recursos computacionais virtuais. Essa característica não diminui a complexidade do problema, mantendo sua complexidade NP-Difícil.

Geralmente, o substrato físico é particionado para hospedar diversos recursos virtuais. Por exemplo, um nó virtual pode ser hospedado sobre qualquer nó físico disponível, além disso, um único nó físico pode hospedar diversos nós virtuais. Dessa forma, o mapeamento de nós virtuais para nós físicos pode ser em uma relação de n:1. Existem casos em que o substrato físico pode ser combinado para formar um novo recurso virtual. Esse é o caso dos canais virtuais que podem estender-se por vários canais (um caminho) do substrato físico. Assim, um canal virtual que conecta dois nós virtuais u e v é mapeado sobre o caminho físico que conecta os nós físicos que hospedam os nós virtuais u e v. Consequentemente, o mapeamento de canais virtuais para caminhos físicos pode ser visto como um relacionamento n:m [9].

Figura 1: Exemplo do mapeamento de duas requisições de infraestruturas virtuais (RIV 1 e 2) sobre o substrato físico. Os números indicam requisições de recursos e capacidade residual do substrato



A Figura 1 descreve um cenário no qual existem duas infraestruturas virtuais contendo três nós cada, hospedadas sobre uma infraestrutura física contendo quatro nós. Os números indicados próximos aos nós representam os recursos requisitados (para as IVs) e a capacidade residual (para o substrato físico). Como pode ser visto, nós físicos conseguem hospedar mais de um nó virtual, de forma semelhante, canais físicos também podem hospedar mais de um canal virtual. Além disso, é possível observar a composição de dois canais físicos para representar um canal virtual (mapeamento de RIV 2). Em cenários típicos, existem, também, restrições que devem ser respeitadas ao realizar o mapeamento. Ou seja, o substrato físico de um mapeamento deve ser capaz de suportar as capacidades requisitadas pelo recurso virtual. Exemplos clássicos de requisitos que devem ser atendidos são: a largura de banda de um canal virtual deve ser menor ou igual a largura de banda suportada pelo caminho físico, o poder de processamento da CPU de um nó virtual deve ser menor ou igual ao poder de processamento provido pelo nó

físico (abordagem similar para recursos de armazenamento e memória) [9].

O mapeamento e a reserva dos recursos não pode ocorrer de forma aleatória, pois existem objetivos a serem alcançados, tanto para o provedor de IVs quanto para o usuário. O principal objetivo de um provedor é alocar o maior número de IVs de uma maneira eficiente, minimizando o número de recursos físicos ativados. Para isso, o algoritmo de alocação deve considerar nós e canais em um único passo, ou seja, com a mesma importância [5]. As infraestruturas virtuais podem ser posicionadas em qualquer lugar sobre o substrato virtualizado, porém o mecanismo de alocação deve minimizar o espalhamento e a fragmentação dos recursos físicos, pois diminuindo o número de recusos físicos ativos reduz também os custos (e.g. energia e resfriamento). Por sua vez, o usuário de uma IV deve ser capaz de realizar requisições de novos serviços para sua IV, como segurança e confiabilidade. Outro objetivo do usuário é a possibilidade de realizar a expansão e contração de sua IV para atender a demanda de seu negócio.

### 2.1 Formalização do problema

Formalmente, o problema de alocar infraestruturas virtuais sobre o substrato físico é conhecido como graph embedding e pode ser descrito da seguinte forma: dado IF = (N, L, C), uma infraestrutura física, onde N representa o conjunto de nós físicos (servidores e roteadores) e L o conjunto de canais físicos. Cada nó ou canal físico  $a \in N \cup L$  está associado com o vetor de capacidade C(a) que representa a quantidade de recursos que a oferece. De forma semelhante, dado  $RIV^i = (N^i, L^i, C^i)$ , um conjunto de i = 1, ..., n requisições de infraestruturas virtuais, onde  $N^i$  e  $L^i$  representam o conjunto de nós e canais virtuais de uma  $RIV^i$ , respectivamente, e  $C^i$  é o vetor de capacidades de um nó ou de um canal virtual (quando o uso de índice é empregado nos conjuntos  $N^i$ ,  $L^i$  ou  $C^i$  refere-se a conjuntos de infraestruturas virtuais, caso o índice não seja usado referencia-se a conjuntos da infraestrutura física IF).

Logo, o mapeamento de infraestruturas virtuais sobre infraestruturas físicas é descrito como  $M:RIV^i \to (M_N,M_P)$ , onde  $M_N:n^i \in N^i \to n \in N$  é um mapeamento de nó virtual para nó físico e  $M_P:l^i \in L^i \to P$  é um mapeamento de um canal virtual para um conjunto P de enlaces físicos. Cada  $RIV^i$  é mapeada considerando o estado atual de IF, ou seja, considera-se o grafo residual de IF para cada mapeamento. Por grafo residual entende-se o grafo com as capacidade dos nós e arestas reduzidos, de acordo com as alocações feitas sobre esse.  $M(RIV^i)$  é considerado um mapeamento válido se para todo nó ou canal virtual de  $RIV^i$  seu respectivo mapeamento não excede às capacidades residuais em IF. Ou seja, em cada alocação,  $\forall n^i \in N^i, C^i(n^i) \leq C(M_N(n^i))$  e  $\forall l^i \in L^i, C^i(l^i) \leq min(C(l), \forall l \in P)$ .

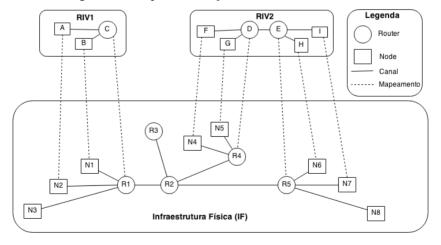


Figura 2: Exemplo de alocação de Infraestruturas Virtuais

Na Figura 2 é mostrado um exemplo de duas requisições de IVs  $(RIV^1 \ e \ RIV^2)$ , que devem ser alocadas sobre a infraestrutura física (IF). Considere que todos os nós e canais de todas infraestruturas (virtuais e física) têm capacidade 1. O mapeamento  $M(RIV^1) = (M_N, M_P)$  formalmente tem seus nós mapeados da seguinte forma:  $M_N(A) \to N2$ ,  $M_N(B) \to N1$  e  $M_N(C) \to R1$ ; e as arestas:  $M_P((C,A)) \to \{(R1,N2)\}$  e  $M_P((C,B)) \to M1$ 

 $\{(R1,N1)\}$ . Já o mapeamento  $M(RIV^2)=(M_N,M_P)$  tem seus nós com o seguinte mapeamento:  $M_N(F) \to N4$ ,  $M_N(G) \to N5$ ,  $M_N(H) \to N6$ ,  $M_N(I) \to N7$ ,  $M_N(D) \to R4$  e  $M_N(E) \to R5$ ; e suas arestas:  $M_P((D,F)) \to \{(R4,N4)\}$ ,  $M_P((D,G)) \to \{(R4,N5)\}$ ,  $M_P((D,E)) \to \{(R4,R2),(R2,R5)\}$  (perceba que, neste caso, houve agregação de dois canais físicos para compor um único canal virtual),  $M_P((E,H)) \to \{(R5,N6)\}$  e  $M_P((E,I)) \to \{(R5,N7)\}$ .

A alocação de IVs sobre uma infraestrutura física está relacionado ao *multi-way separator problem* [11], que já é provado pertencer a classe NP-Difícil. Mesmo simplificando o problema e considerando que canais virtuais sejam mapeados para canais físicos únicos (retirando a noção de caminho), ainda continua com a mesma complexidade, pois está relacionado ao problema do fluxo indivisível, que pertence a esse mesmo conjunto [12]. Logo, soluções ótimas são viáveis apenas para pequenas instâncias do problema, porém, em cenários reais as instâncias não apresentam um valor desprezível (motivadas pela noção de recursos infinitos e escalabilidade oferecida pela Nuvem Computacional). Assim, o foco deste trabalho está na utilização de heurísticas para alocação de IVs.

# 3 Algoritmos para alocação de Infraestruturas Virtuais

Diversos algoritmos podem ser utilizados para definir um plano de alocação de recursos para hospedar IVs [9]. Nesta seção, descrevemos quatro algoritmos que foram implementados no simulador CloudSim (a implementação é discutida na seção 4). Dentre os algoritmos analisados, Melhor Ajuste, Pior Ajuste e FCFS (*First-Come-First-Served*) representam uma visão do InP, ou seja, realizam a alocação considerando a utilização do substrato físico (e sua organização). Por sua vez, o algoritmo *Alocação Orientada por Distância (AOD)* representa uma visão do usuário solicitante: ao alocar recursos virtuais geograficamente próximos, o algoritmo tende a economizar a largura de banda necessária e diminuir a latência entre os recursos, proporcionando uma melhor experiência para o usuário final [13].

As soluções investigadas neste trabalho são caracterizadas como algoritmos *on-line*, ou seja, usuários podem submeter pedidos de IVs a qualquer momento e o provedor analisará a viabilidade de alocação. Optou-se por aceitar e analisar requisições individualmente. Nesse cenário, quando uma IV já está alocada e outra IV é solicitada, o provedor deve escolher uma de duas abordagens: i) realocar todas as IVs, o que pode melhorar a utilização do substrato (por meio do balanceamento de carga), mas aumenta o tempo de alocação e introduz problemas de migração de IVs; ou ii) manter os nós e enlaces pré-alocados em seus respectivos recursos físicos, o que evita descumprir *Service Level Agreements* (SLAs) existentes. Seguindo a abordagem atualmente utilizada por provedores populares de nuvens, optou-se pela estratégia ii.

É importante ressaltar que a decisão quanto à alocação (ou não) de um recurso deve considerar a capacidade disponível para hospedar e garantir o cumprimento do SLA. Conforme formalizado na seção 2.1, esse procedimento pode ser simplificado em um conjunto de vetores que representam os recursos disponíveis e solicitados. Para simplificar a discussão sobre os algoritmos, concentramos a exemplificação em apenas uma métrica (a saber, número de processadores) enquanto a análise experimental (seção 5) considera as demais características do substrato físico e das MVs (memória, largura de banda, entre outros).

Para exemplificar os algoritmos utilizados neste trabalho, a Figura 3 apresenta quatro requisições de MVs (identificadas por MV1, MV2, MV3 e MV4), que devem ser sequencialmente alocadas em um datacenter composto por dez hosts (H1 até H10). Na coluna Requisições, os valores representam a capacidade requisitada, enquanto, na coluna Datacenter, representam a capacidade disponível no recurso. Nas demais colunas, identificadas pelo nome do algoritmo, representam o estado do substrato físico após a alocação dos recursos, ou seja, indicam a capacidade residual. Quando uma reserva ocorre, o identificador da MV é acrescentado à célula, indicando um mapeamento. A relação entre os switches e hosts (parte inferior da tabela) representa a topologia simplificada neste datacenter de exemplo. Os hosts vinculados aos switches foram postos de maneira aleatória, com a finalidade de representar o funcionamento do algoritmo que considera a distância entre os recursos virtuais. O Algoritmo 1 apresenta os passos necessários para execução dos algoritmos Melhor Ajuste, Pior Ajuste e FCFS, que são detalhados a seguir.

*Melhor Ajuste:* uma alocação, utilizando essa abordagem, realiza uma busca pelo hospedeiro que deixará a menor capacidade residual. Em outras palavras, esse algoritmo procura deixar lacunas menores, que futuramente poderão ser utilizadas para a alocação de MVs de pequeno porte. Analisando a Figura 3 podemos perceber o empenho do

	<u> </u>	1	1 3		3		
	Requisições		Datacenter	Melhor Ajuste	Pior Ajuste	FCFS	AOD
MV1	3	H1	36	36	36	MV1   MV2   14	36
MV2	19	H2	55	55	55	MV3   MV4   37	55
MV3	17	Н3	77	77	MV2   MV4   57	77	77
MV4	1	H4	18	MV3   MV4   0	18	18	18
		H5	3	MV1   0	3	3	MV1   0
		Н6	78	78	MV1   MV3   58	78	78
		Н7	11	11	11	11	11
Switch	Hosts alocados	Н8	39	39	39	39	MV2   MV3   MV4   2
1	H5   H2   H8   H6   H9	Н9	14	14	14	14	14
2	H3   H10   H7   H1   H4	H10	26	MV2   7	26	26	26

Figura 3: Exemplo de aplicação dos algoritmos de alocação em um Datacenter

**Algorithm 1:** Pseudocódigo baseado em listas de capacidades residuais para alocação de uma infraestrutura virtual. O vetor indicado por cand(N) é ordenado conforme a abordagem utilizada

- 1 cand(N): lista de candidatos físicos ordenados por capacidade residual (crescente, decrescente ou aleatório)
- 2  $RIV^i$ : requisição de uma infraestrutura virtual

```
3 for n^i \in N^i do

4 | while cand(N) != vazio do

5 | n = \operatorname{pr\'oximo} candidato de cand(N)

6 | if C^i(n^i) \leq C(n) then

7 | registre M_N(n^i,n)

8 | break

9 if |M_N| \neq |N^i| then

10 | retorne falso

11 if \exists M_P(l^i,l) \forall l^i \in L^i then

12 | retorne verdadeiro

13 retorne falso
```

algoritmo *Melhor Ajuste* ao tentar reduzir ao máximo a capacidade residual após a alocação, de modo que a MV4, que necessitava de apenas uma unidade de processamento, pôde ser alocada no mesmo hospedeiro que a MV3, que havia deixado uma unidade residual.

*Pior Ajuste:* de maneira oposta ao algoritmo anterior, neste caso, os recursos físicos candidatos a hospedar a MV são ordenados por ordem decrescente de capacidade residual. O primeiro item da lista é escolhido para hospedar a MV. Essa abordagem tende, inicialmente, a distribuir os recursos no substrato, deixando lacunas maiores. A longo prazo, essa abordagem pode prejudicar a alocação de MVs maiores (que necessitam de mais capacidade). O exemplo descrito na Figura 3 indica que as MVs são alocadas nos recursos físicos que apresentam mais capacidade. O hospedeiro H5, por exemplo, não vai ser selecionado para uma alocação enquanto houverem recursos físicos de/com maior capacidade residual.

FCFS: o algoritmo First-Come-First-Served (FCFS), como o próprio nome sugere, efetua a alocação da MV no primeiro hospedeiro disponível que for encontrado. Esse algoritmo tem um desempenho praticamente igual ao Melhor Ajuste caso a lista de candidatos esteja ordenada de maneira crescente, e semelhante ao Pior Ajuste caso a lista esteja ordenada de maneira decrescente. Essa abordagem foi selecionada para análise por ser o algoritmo padrão para alocação no CloudSim. No exemplo de alocação da Figura 3, o algoritmo tende a ocupar as primeiras posições vagas do datacenter. Qualquer tipo de otimização ou de melhor escolha de alocação, como discutido nos algoritmos anteriores, não é efetuado.

Alocação Orientada por Distância (AOD): esse algoritmo busca diminuir o número de saltos entre as MVs, ou seja, o algoritmo busca a centralização de MVs que pertencem a um mesmo usuário. Ao utilizar essa abordagem, o provedor pode reduzir os caminhos físicos necessários para hospedar os canais virtuais. Consequentemente, o somatório de largura de banda alocada para a comunicação entre as MVs é reduzido. Cada salto equivale a um encaminhamento realizado por um roteador/comutador. Especificadamente, MVs, alocadas em um mesmo hospedeiro físico têm uma unidade de salto caso possuam comunicação (existência de um canal virtual). Nesse algoritmo (representado pelo Algoritmo 2), há a necessidade de informações relacionadas a qual hospedeiro um determinado usuário tem MVs alocadas, bem como qual a topologia do substrato físico. Com base nessas informações, o algoritmo aloca uma determinada MV em algum hospedeiro já ocupado por esse mesmo usuário requisitante (ou seja, já existe uma outra MV do usuário alocada no hospedeiro). Caso a abordagem não tenha sucesso, AOD busca selecionar um hospedeiro próximo (provavelmente conectado no mesmo comutador). Essa última abordagem é repetida, aumentando a distância permitida. Por fim, caso hajam vários candidatos disponíveis (independente da distância), o algoritmo Melhor Ajuste é utilizado para selecionar o candidato, buscando diminuir a capacidade residual do recurso físico. No pior caso, AOD busca um hospedeiro qualquer para alocar essa MV requisitada, visto que não foi possível alocar, considerando a heurística principal.

**Algorithm 2:** Pseudocódigo para alocação de uma infraestrutura virtual considerando a distância entre os recursos

- 1 cand(N): lista de candidatos físicos em ordem crescente de capacidade residual 2 ;  $cand(M_N)$ : lista de candidatos físicos que já têm recursos alocados em ordem crescente de
- 3 RIV<sup>i</sup>: requisição de uma infraestrutura virtual

capacidade residual

```
4 for n^i \in N^i do
       while cand(M_N) != vazio do
5
            n = \text{pr}óximo candidato de cand(M_N)
6
            if C^i(n^i) \leq C(n) then
7
                registre M_N(n^i, n)
8
9
                break
       if \exists n^i \in M_N then
10
          continue;
11
        while cand(N) != vazio do
12
            n = \text{pr\'oximo candidato de } cand(N)
13
            if C^i(n^i) \leq C(n) then
14
15
                registre M_N(n^i, n)
                continue
16
17 if |M_N| \neq |N^i| then
      retorne falso
19 if \exists M_P(l^i, l) \forall l^i \in L^i then
    retorne verdadeiro
21 retorne falso
```

Na Figura 3, por exemplo, para a primeira requisição não existe nenhuma MV já alocada por esse usuário. Portanto, AOD seleciona um candidato por meio da abordagem *Melhor Ajuste* e a MV é alocada em H5. Para a próxima requisição, MV2, como não há nenhum hospedeiro com espaço livre, o algoritmo busca a alocação no mesmo comutador que conecta H5. Nesse caso, utilizando o *Melhor Ajuste*, H8 é selecionado. Por fim, para as máquinas virtuais MV3 e MV4, AOD obtêm sucesso ao alocar as duas MVs no mesmo hospedeiro (H8). Consequentemente, o número de saltos entre as MVs foi reduzido.

### 4 Estendendo o simulador CloudSim para alocação de IVs

CloudSim é um *framework* para simulação de cenários em Computação em Nuvem. Conforme [14], esse é um dos mais completos simuladores de eventos discretos para nuvens computacionais. Em relação às suas principais características, Calheiros et al. [10] definem CloudSim como um *framework* extensível de simulação que permite a modelagem, a simulação e a experimentação de infraestruturas emergentes de computação em nuvem e serviços de aplicações. Esse simulador oferece classes básicas para descrição de *datacenters*, MVs, aplicações, usuários, recursos computacionais e políticas para o gerenciamento de diversas partes do sistema, tais como alocação e provisionamento. Desenvolvido na linguagem Java, os componentes do CloudSim podem ser agrupados para avaliar novas estratégias na utilização de nuvens (por exemplo: algoritmos de alocação e políticas de balanceamento de cargas, entre outros). Além disso, pode também ser usado para avaliar a eficiência de estratégias sob diferentes perspectivas, tais como a relação custo-benefício ou até em relação ao tempo de execução de uma aplicação.

O simulador permite, por intermédio da programação (utilização das classes existentes ou definição de novos componentes), a especificação de características da simulação, ou seja, o modelo do cenário da nuvem, os requisitos do usuário, as configurações da aplicação, além de definir a política de alocação da nuvem. Nesse ponto, acrescentamos os algoritmos discutidos na seção 3. O núcleo de simulação do CloudSim considera um *datacenter* como um provedor de recursos de infraestrutura, ou seja, uma entidade que recebe as requisições de alocação de recursos virtuais dos *brokers* (ou usuários) e as instancia nos hospedeiros do *datacenter*. Essa camada de simulação provê suporte para modelagem e simulação de memória, armazenamento, processamento e largura de banda.

Embora ofereça diversos recursos previamente modelados, a implementação original do CloudSim não oferece suporte para a simulação de redes (roteadores, comutadores e canais de comunicação). Essa limitação é suprida por meio de uma extensão do CloudSim, denominada NetworkCloudSim, que introduz conceitos e classes para modelar a topologia física e o comportamento geral de uma aplicação distribuída [14]. Essa extensão forneceu a base para implementação dos algoritmos para alocação de infraestruturas virtuais. Mais especificadamente, os algoritmos implementados consideram simultaneamente a alocação de recursos computacionais (MVs) e de comunicação (roteadores e canais de comunicação).

Em relação à implementação dos algoritmos no CloudSim, essas foram realizadas por meio da definição de cenários específicos para os testes e, também, da alteração das classes referentes às políticas de alocação (*AllocationPolicy*, na nomenclatura interna do simulador). Por padrão, no CloudSim, as IVs são individualmente alocadas (respeitando o modelo discutido na seção 2.1), ou seja, a tomada de decisão deve ser realizada considerando unicamente a IV recebida e o estado atual do substrato.

Para realizar uma simulação com o *framework* são necessários três componentes básicos: criação de um (ou vários) *datacenter*, definição de um (ou vários) *brokers* – que representam os usuários da nuvem – e, por fim, a criação da lista de IVs e *cloudlets* que cada usuário irá possuir. Particularmente, um *cloudlet* define uma representação abstrata do comportamento computacional e comunicante de uma aplicação que será executada sobre os recursos virtuais. Quando um *datacenter* é criado no CloudSim, esse necessita também de uma política de alocação que será utilizada para reserva e instanciação das IVs. A política de alocação apresenta, além de outras informações, uma lista contendo todos os hospedeiros do *datacenter* e sua topologia física. A partir dessas informações, cada um dos algoritmos realiza uma busca por um hospedeiro adequado para a alocação dos recursos virtuais.

### 5 Simulação da alocação de Infraestruturas Virtuais

Para analisar o comportamento dos algoritmos discutidos na seção 3, realizamos uma simulação com o *framework* CloudSim versão 3.0.3 com a extensão NetworkCloudSim ativa. Os experimentos foram realizados em um computador AMD Phenom II X4 com 4 GB de RAM e Linux Ubuntu 12.04.02. O simulador foi executado com o Java versão 1.7.0\_09. A análise considerou quatro configurações de *datacenters* com topologia *fat tree*, seguindo as especificações de projeto da arquitetura de referência Cisco [15], a qual é normalmente implementada em grandes *datacenters* de alta resiliência. Baseada em trabalhos anteriores [16], a Tabela 1 apresenta as configurações dos substratos físicos e das IVs requisitadas. Cada hospedeiro físico tem uma configuração de número de processores,

capacidade de memória e armazenamento definida, seguindo uma distribuição uniforme. Para processadores, os valores foram distribuídos entre 1 e 8; para RAM, entre 512 MB e 8192 MB; e, para armazenamento, entre 500 GB e 5 TB. As infraestruturas virtuais solicitam entre 3 e 5 MVs, sendo suas requisições de capacidade uniformemente distribuídas entre 1 e 8 para processadores e entre 512 MB e 8192 MB para RAM. A capacidade de armazenamento apresenta um valor fixo requisitado de 10 GB. Valores para RAM foram selecionados em intervalos de 512 MB.

Tabela 1: Configuração dos quatro cenários de teste implementados. Cada cenário representa uma topologia física *fat tree*. Os cenários são analisados individualmente

Cenário	# Hospedeiros	# Switches
Cenário 1	25	5
Cenário 2	100	20
Cenário 3	200	40
Cenário 4	300	60

A análise dos resultados das simulações consideram três métricas: Taxa de Aceitação (TA), Carga do Substrato (CS) e Distância Média entre os Recursos Virtuais (DMRV). TA é calculada pela razão entre o número de IVs alocadas com sucesso e o total de IVs solicitadas pelos usuários, representando o nível de aceitação de cada cenário, ou seja, a porcentagem de IVs que foram alocadas e instanciadas pelo InP. Já CS é calculada pela razão entre o somatório da capacidade física de todas as MVs que foram alocadas e a capacidade física total do *datacenter*, e representa o percentual dos recursos computacionais utilizadas para hospedar as IVs. Por sua vez, DMRV representa a distância média entre as MVs de um mesmo cliente, sendo calculada em número de saltos. Em média, cada simulação recebeu 160 requisições de MVs, sendo essas o somatório de todas as 40 IVs pelos usuários. Cada algoritmo foi executado quinhentas vezes e os gráficos apresentam a média das execuções, considerando um intervalo de confiança de 95%. As próximas subseções discutem o comportamento dos algoritmos em relação a cada métrica e cenário.

#### 5.1 Cenário 1

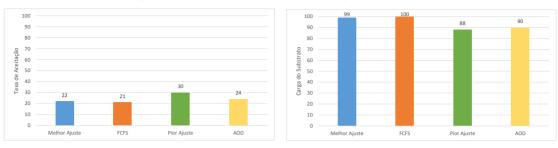
Os resultados da simulação com o Cenário 1 são apresentados na Figura 4. A Figura 5(a) apresenta a TA para todos os algoritmos analisados. Devido ao limite computacional imposto pelo cenário (capacidade limitada do substrato físico), nenhuma implementação conseguiu alocar mais do que 30% das IVs solicitadas. O algoritmo guiado por uma abordagem *Pior Ajuste* conseguiu alocar aproximadamente 10% e 7% mais requisições que as outras abordagens. Por sua vez, a Figura 5(b) apresenta os resultados para a métrica de CS. Nessa figura, observase que os algoritmos *Melhor Ajuste* e *FCFS* saturaram o substrato computacional e consequentemente resultaram em uma menor taxa de aceitação, conforme discutido na Figura 5(a). Em relação à métrica DMRV (resultados apresentados na Figura 5(c)), não existe uma variação significativa para nesse cenário. Essa percepção tende a aumentar nos cenários subsequentes.

#### 5.2 Cenário 2

Os resultados para o Cenário 2 são apresentados na Figura 5. Considerando a métrica TA para esse cenário (Figura 6(a)), observa-se que o algoritmo *Melhor Ajuste* apresenta um comportamento mais eficiente comparado com o cenário anterior. Percebe-se, também, pela Figura 6(b), que esse algoritmo obteve 75% de TA justamente por gerenciar os espaços de alocação de maneira mais eficiente. O *FCFS*, por outro lado, utilizou 88% da capacidade do *datacenter*, mas não conseguiu alocar mais que 58% das requisições de IVs.

Por sua vez, a abordagem com *Pior Ajuste* não permite a total ocupação do *datacenter* e ao mesmo tempo, não alocada mais do que 67% das IVs requisitadas. Ou seja, *Pior Ajuste* fragmentou o substrato computacional, não permitindo a alocação de novas requisições. Em relação à DMRV, o algoritmo AOD (Figura 6(c)) diminuiu a distância média entre os recursos, mas, em contrapartida, 34% das suas requisições não foram atendidas (comportamento similar ao algoritmo de *Pior Ajuste* e 9% inferior ao *Melhor Ajuste*).

Figura 4: Métricas obtidas com a simulação do Cenário 1



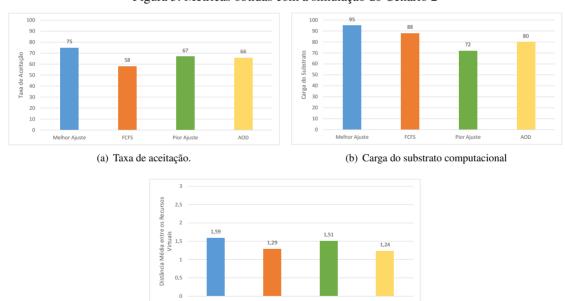
(a) Taxa de aceitação.

(b) Carga do substrato computacional



(c) Distância média entre os recursos virtuais

Figura 5: Métricas obtidas com a simulação do Cenário 2



(c) Distância média entre os recursos virtuais

## 5.3 Cenário 3

Analisando os resultados do Cenário 3 (Figura 6), percebe-se que a abordagem com *Melhor Ajuste* foi capaz de alocar todas as requisições (valores de TA na Figura 7(a)). Por sua vez, a *Alocação Orientada por Distância* teve um resultado semelhante em relação ao *Pior Ajuste*, tanto em relação à TA quanto na CS (Figura 7(c)). Ambos

os algoritmos tiveram aproximadamente 87% das IVs alocadas, mesmo tendo uma grande porção do *datacenter* livre, ou seja, fragmentado, de modo a prejudicar a alocação de MVs de grande porte e limitando assim a sua TA.

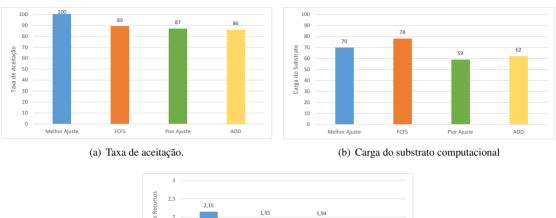
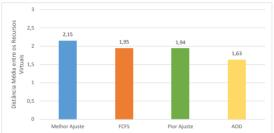


Figura 6: Métricas obtidas com a simulação do Cenário 3



(c) Distância média entre os recursos virtuais

Em relação à ocupação dos recursos físicos, apresentado na Figura 7(b), o *Melhor Ajuste* apresentou melhores resultados. Embora acomode todas as requisições submetidas, o substrato permanece com recursos disponíveis para receber novas requisições (aproximadamente 30%). Por outro lado, o algoritmo *FCFS*, que resultou em 89% das IVs alocadas, impôs uma carga no substrato de aproximadamente 78%. Como mencionado anteriormente, a abordagem *AOD* é similar ao *Pior Ajuste* quando levados em conta a TA e a CS. Em relação a DMRVs, nota-se um valor médio menor comparado aos demais algoritmos.

#### 5.4 Cenário 4

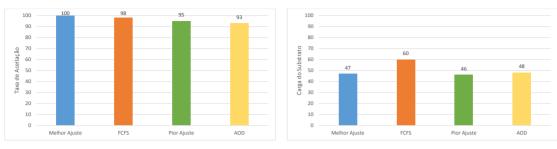
Neste último cenário, os algoritmos conseguiram alocar quase totalmente as IVs submetidas (Figura 8(a)). Em relação à CS (Figura 8(b)), observa-se que o *Melhor Ajuste* apresenta uma melhor utilização do substrato (baixa carga). Resultados similares são observados nas abordagens com *Pior Ajuste* e *AOD*, mas tais algoritmos não conseguiram alocar todas as requisições submetidas (conforme ilustrado pela métrica TA), e, portanto, podem apresentar um *datacenter* fragmentado.

Por fim, é nesse cenário que se observa a maior eficiência do algoritmo *Alocação Orientada por Distância*, visto que a média dos outros algoritmos é de aproximadamente 2,1 saltos, e já no *Alocação Orientada por Distância* esse número não passa de 1,8. É de se esperar que em *datacenter* maiores essa diferença torne-se mais visível, visto que, conforme se aumentoue o número de hospedeiros (analisando sequencialmente os Cenários 1, 2, 3 e 4), as MVs tendem a se espalhar *datacenter*, aumentando, assim, o tamanho da banda necessária para estabelecer a comunicação entre os recursos virtuais hospedados. A *Alocação Orientada por Distância* objetiva limitar esse espalhamento, induzindo uma menor latência entre os recursos virtuais comunicantes, além de um menor custo quanto ao provisionamento de largura de banda [13] [17].

#### 5.5 Discussão e considerações

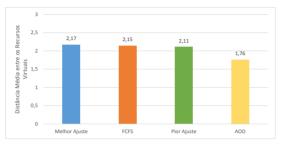
Fazendo uma análise geral dos cenários de testes, percebe-se que o aumento da TA ocorreu conforme o aumento do número de recursos disponíveis no substrato físico. Entretanto, nos algoritmos *Pior Ajuste* e *AOD*, por

Figura 7: Métricas obtidas com a simulação do Cenário 4



(a) Taxa de aceitação.

(b) Carga do substrato computacional



(c) Distância média entre os recursos virtuais

exemplo, observou-se, também, o aumento da fragmentação dos recursos do *datacenter*, inclusive, no Cenário 4 (cenário com mais recursos) os algoritmos não conseguiram alocar todas as solicitações de IVs. Essa fragmentação não foi observada no algoritmo de *Melhor Ajuste*, visto que, mesmo em um *datacenter* com duzentos hospedeiros (Cenário 3), 100% das IVs foram alocadas com sucesso, restando ainda capacidade residual para futuras alocações.

É importante notar a semelhança em todos os cenários dos resultados entre as abordagens com *Pior Ajuste* e *AOD* para as métricas de TA e CS. Ou seja, ao custo de manter ao máximo possível a proximidade dos recursos virtuais, o algoritmo da AOD tende a ter um desempenho similar ao *Pior Ajuste*, de modo a apresentar os seus mesmos defeitos, tais como a fragmentação do substrato físico. Em contrapartida, *AOD* obteve o melhor desempenho, considerando a métrica DMRV para todos os cenários analisados.

As heurísticas discutidas neste trabalho podem ser facilmente implementadas e adaptadas pelos provedores de nuvens computacionais existentes. Prováveis adaptações compreendem a introdução dos aspectos financeiros, ou seja, a representação dos custos computacionais (métrica CS) em valores monetários, bem como a introdução de métricas, representando a receita obtida ao provisionar uma IV.

### 6 Trabalhos relacionados

A alocação de recursos para hospedar máquinas e canais de comunicação virtuais já foi abordada em diversos trabalhos divulgados nos últimos anos. Usualmente, os trabalhos são diferenciados pela abordagem implementada: alocação ótima ou heurísticas com soluções aproximadas; diferentes funções objetivo (economia de recursos, fragmentação, custo de alocação, entre outros); formulação *on-line* ou *off-line*, com ou sem migração de recursos; e implementação em simuladores de eventos discretos ou simuladores específicos. Inicialmente, o problema foi modelado, considerando os requisitos para implementação de VPNs [18] [19] [20]. Nesse cenário, as MVs não eram consideradas e o problema era modelado apenas considerando canais de comunicação (as extremidades eram conhecidas *a priori*).

Recentemente, a comunidade científica voltou-se para formulações que consideram a alocação simultânea de recursos computacionais e de comunicação. Essa nova abordagem revelou uma complexidade NP-Difícil [8] e motivou o desenvolvimento de novas heurísticas, visando uma solução eficiente em tempo computacional li-

near. A literatura especializada compreende abordagens com funções objetivo específicas, tais como prover a alta disponibilidade [21] [22], alocar IVs tolerantes a falhas [23] e minimizar a latência entre os recursos virtuais [24].

Diversas implementações buscam limitar o espaço de busca por meio da organização dos candidatos em grupos [25], ou da limitação do tamanho máximo de um canal de comunicação virtual [26]. Recentemente, uma abordagem indicou a possibilidade de ranqueamento dos recursos virtuais, considerando sua complexidade ou importância para o usuário [27]. Já um ranqueamento, dada a probabilidade de um recurso físico alocar recursos virtuais foi abordado, demonstrando que a carga do substrato computacional pode ser diminuída e, consequentemente, o lucro do provedor maximizado [28]. Embora apresentem resultados satisfatórios para funções objetivo específicas, essas formulações não representam o cenário real vivenciado por InPs e apresentam limitações quanto à escalabilidade do sistema. Esse é um dos pontos em que o presente trabalho diferencia-se dos demais. Por intermédio da implementação de algoritmos para alocar Infraestruturas Virtuais no simulador CloudSim, permitiu-se a simulação não somente na visão do InP, mas, também, do usuário final requisitante (ou do provedor de serviço - SP). Ainda, a alocação de IVs compreende o mapeamento combinado de recursos computacionais e de comunicação.

Considerando a simulação específica de recursos de Nuvens Computacionais, o trabalho desenvolvido em [29] permite a definição de um *datacenter* de nuvem em detalhes, mas não considera os aspectos de comunicação, como a abordagem AOD discutida na seção 3. Os algoritmos implementados neste trabalho permitem a reserva de recursos de comunicação, além dos recursos computacionais.

#### 7 Conclusão

Este artigo abordou a complexa tarefa de alocar recursos para hospedar Infraestruturas Virtuais. Para tal, quatro algoritmos foram implementados e analisados no simulador CloudSim (considerando os recursos da extensão NetworkCloudSim). Originalmente, CloudSim permite a modelagem de substratos computacionais, máquinas virtuais e aplicações distribuídas, não oferecendo recursos para alocação de canais de comunicação. As heurísticas implementas no presente estudo introduziram a alocação combinada de recursos computacionais e de comunicação no simulador. A análise experimental considerou quatro topologias físicas distinticas no formato *fat tree*, que separadamente receberam a submissão de diversas requisições de IVs com valores uniformemente distribuídos. Para efetuar a análise dos algoritmos, a perspectiva dos InPs foi representada pelas métricas de taxa de aceitação e carga do substrato computacional, enquanto a perspectiva do usuário foi representada pela distância média entre os recursos virtuais. Como perspectivas de trabalhos futuros, podem ser investigadas heurísticas, considerando outras métricas (e objetivos), tais como confiabilidade, computação sustentável e custo de alocação, entre outros. Ainda, a migração de recursos virtuais previamente reservados pode permitir uma melhor distribuição de carga no substrato e diminuir os impactos resultantes de sua fragmentação.

### Agradecimentos

Os autores agradecem à Udesc pelo auxílio financeiro e ao Laboratório de Processamento Paralelo e Distribuído (LabP2D) pela disponibilização dos recursos computacionais para a realização do trabalho.

#### Referências

- [1] MELL, P.; GRANCE, T. *The NIST Definition of Cloud Computing*. 26. ed. [S.l.], 2009. Disponível em: <a href="http://goo.gl/PsJP5R">http://goo.gl/PsJP5R</a>.
- [2] BUYYA, R.; YEO, C. S.; VENUGOPAL, S. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In: *Proc. of the 10th IEEE Int. Conf. on High Performance Computing and Communications*. Washington, DC, USA: IEEE Computer Society, 2008. (HPCC '08), p. 5–13. ISBN 978-0-7695-3352-0.

- [3] JENNINGS, B.; STADLER, R. Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, Springer US, p. 1–53, 2014. ISSN 1064-7570. Disponível em: <a href="http://dx.doi.org/10.1007/s10922-014-9307-7">http://dx.doi.org/10.1007/s10922-014-9307-7</a>>.
- [4] FOSTER, I. et al. Cloud computing and grid computing 360-degree compared. In: IEEE. *Grid Computing Environments Workshop*, 2008. GCE'08. [S.I.], 2008. p. 1–10.
- [5] CHOWDHURY, N. M. K.; RAHMAN, M. R.; BOUTABA, R. Virtual Network Embedding With Coordinated Node and Link Mapping. In: *Proc. IEEE INFOCOM*. [S.l.: s.n.], 2009. p. 783–791.
- [6] ANHALT, F.; KOSLOVSKI, G.; PRIMET, P. V.-B. Specifying and Provisioning Virtual Infrastructures with HIPerNET. *Int. J. Network Management*, John Wiley & Sons, Inc., v. 20, n. 3, p. 129–148, 2010.
- [7] RIGHI, R. da R. Elasticidade em cloud computing: conceito, estado da arte e novos desafios. *Revista Brasileira de Computação Aplicada (RBCA)*, v. 5, n. 2, p. 2–17, 2013.
- [8] CHOWDHURY, N. M. K.; BOUTABA, R. Network Virtualization: State of the Art and Research Challenges. *Comm. Magazine, IEEE*, v. 47, n. 7, p. 20–26, 2009.
- [9] FISCHER, A. et al. Virtual network embedding: A survey. Communications Surveys Tutorials, IEEE, v. 15, n. 4, p. 1888–1906, Fourth 2013. ISSN 1553-877X.
- [10] CALHEIROS, R. N. et al. Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exper.*, John Wiley & Sons, Inc., New York, NY, USA, v. 41, n. 1, p. 23–50, jan. 2011. ISSN 0038-0644. Disponível em: <a href="http://dx.doi.org/10.1002/spe.995">http://dx.doi.org/10.1002/spe.995</a>.
- [11] ANDERSEN, D. G. Theoretical approaches to node assignment. Computer Science Department, p. 86, 2002.
- [12] KOLLIOPOULOS, S. G.; STEIN, C. Improved approximation algorithms for unsplittable flow problems. In: IEEE. *Foundations of Computer Science*, 1997. Proceedings., 38th Annual Symposium on. [S.l.], 1997. p. 426–436.
- [13] KOSLOVSKI, G. et al. Locating Virtual Infrastructures: Users and InP Perspectives. In: 12th IEEE/IFIP International Symposium on Integrated Network Management Special Track on Management of Cloud Services and Infrastructures (IM 2011 STMCSI). Dublin, Ireland: IEEE, 2011. Disponível em: <a href="http://www.ieee-im.org/">http://www.ieee-im.org/</a>>.
- [14] GARG, S.; BUYYA, R. Networkcloudsim: Modelling parallel applications in cloud simulations. In: *Utility* and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on. [S.l.: s.n.], 2011. p. 105–113.
- [15] CISCO Data Center Infrastructure: 2.5 Design Guide. 2007.
- [16] CAVALCANTI, G. A. de S.; OBELHEIRO, R. R.; KOSLOVSKI, G. Optimal resource allocation for survivable virtual infrastructures. In: 10th International Conference on Design of Reliable Communication Networks DRCN. [S.l.]: IEEE, 2014. p. 1–8.
- [17] POPA, L. et al. Faircloud: Sharing the network in cloud computing. In: *Proceedings of the 10th ACM Workshop on Hot Topics in Networks*. New York, NY, USA: ACM, 2011. (HotNets-X), p. 22:1–22:6. ISBN 978-1-4503-1059-8. Disponível em: <a href="http://doi.acm.org/10.1145/2070562.2070584">http://doi.acm.org/10.1145/2070562.2070584</a>>.
- [18] DUFFIELD, N. G. et al. A flexible model for resource management in virtual private networks. In: SIG-COMM '99: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication. New York, NY, USA: ACM, 1999. p. 95–108. ISBN 1-58113-135-6.
- [19] GUPTA, A. et al. Provisioning a virtual private network: a network design problem for multicommodity flow. In: *STOC '01: Proceedings of the thirty-third annual ACM symposium on Theory of computing*. New York, NY, USA: ACM, 2001. p. 389–398. ISBN 1-58113-349-9.
- [20] LU, J.; TURNER, J. *Efficient Mapping of Virtual Networks onto a Shared Substrate*. [S.l.], 2006. Disponível em: <a href="http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_report\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.edu/~{}jl1/research/tech\\_2006.p>">http://www.arl.wustl.ed

- [21] BODÍK, P. et al. Surviving Failures in Bandwidth-Constrained Datacenters. In: *Proc. ACM SIGCOMM*. [S.l.: s.n.], 2012. p. 431–442.
- [22] ZHANG, Q. et al. Venice: Reliable virtual data center embedding in clouds. In: *INFOCOM*, 2014 Proceedings IEEE. Toronto, Ontario, Canada: [s.n.], 2014.
- [23] RAHMAN, M. R.; AIB, I.; BOUTABA, R. Survivable Virtual Network Embedding. In: *Proc. IFIP Networking*. [S.l.: s.n.], 2010. p. 40–52.
- [24] BARLA, I. B.; SCHUPKE, D. A.; CARLE, G. Resilient Virtual Network Design for End-to-End Cloud Services. In: *Proc. IFIP Networking*. [S.l.: s.n.], 2012. p. 161–174.
- [25] RICCI, R.; ALFELD, C.; LEPREAU, J. A Solver for the Network Testbed Mapping Problem. *SIGCOMM Computer Communications Review*, v. 33, p. 30–44, 2003.
- [26] LISCHKA, J.; KARL, H. A Virtual Network Mapping Algorithm Based on Subgraph Isomorphism Detection. In: ACM. *Workshop on Virtualized Inf. Sys. and Arch.* [S.l.], 2009. p. 81–88.
- [27] CHENG, X. et al. Virtual network embedding through topology-aware node ranking. *SIGCOMM Comput. Commun. Rev.*, ACM, New York, NY, USA, v. 41, n. 2, p. 38–47, abr. 2011. ISSN 0146-4833. Disponível em: <a href="http://doi.acm.org/10.1145/1971162.1971168">http://doi.acm.org/10.1145/1971162.1971168</a>>.
- [28] GONG, L. et al. Toward profit-seeking virtual network embedding algorithm via global resource capacity. In: *INFOCOM*, 2014 Proceedings IEEE. Toronto, Ontario, Canada: [s.n.], 2014. p. 1–9.
- [29] ALKMIN, C. P.; CORDEIRO, D. SimMyCloud, simulando o gerenciamento de recursos virtualizados em plataformas de computação em nuvem. In: *Salão de Ferramentas SBRC 2014*. [S.l.: s.n.], 2014.