

## Consolidação de MVs em nuvem IaaS orientada por operações elásticas e focada em consumo energético

Denivy B. Rück, Charles C. Miers, Maurício A. Pillon, Guilherme P. Koslovski

<sup>1</sup>Programa de Pós-Graduação em Computação Aplicada (PPGCA)  
Universidade do Estado de Santa Catarina (UDESC) - Joinville – SC – Brasil  
*denivy.ruck@labp2d.udesc.br, {charles.miers, mauricio.pillon, guilherme.koslovski}@udesc.br*

**Resumo.** *O provisionamento de Infraestruturas Virtuais (IVs) elásticas possibilita o gerenciamento dinâmico de recursos computacionais em nuvens. Com a elasticidade, recursos virtuais de computação, armazenamento e comunicação são ajustados aos requisitos da aplicação hospedada (e.g., carga submetida, número de usuários). Para atender as solicitações elásticas dos usuários, provedores especificam políticas de gerenciamento e aplicam mecanismos de realocação. A preocupação com sustentabilidade e com custos variáveis torna o consumo energético de data centers um tema recorrente no gerenciamento de provedores. Sobretudo, reduzir o consumo energético é interessante tanto para usuários quanto para provedores. O algoritmo proposto considera o compartilhamento proporcional do uso de CPU de servidores hospedeiros para calcular o consumo individual, desativando equipamentos ociosos e reorganizando os recursos virtuais. A análise experimental indica uma redução do consumo energético de até 50%, sem impactar na taxa de aceitação de novas requisições.*

**Abstract.** *The elastic provisioning of virtual infrastructures (IVs) enable a dynamic management of cloud resources. Based on elasticity, virtual computing, storage, and communication the resources are adjusted to fit on hosted application requirements. In order to accomplish elasticity requests, providers rely on reallocation mechanisms and policies. The concern, regarding sustainability and operational costs, turns the data center power consumption a recurring theme in provider policies. Moreover, power-aware provisioning is beneficial for both tenants and providers. Thus, we propose an algorithm considering the proportional sharing of CPU usage of data center servers in order to calculate individual costs, disable idle equipments, and reallocate virtualized resources. The experimental analysis indicates a reduction of energy consumption near to 50% without impacting the acceptance ratio of new requests.*

### Introdução

Os benefícios administrativos e econômicos oferecidos, tanto para provedores quanto para clientes, garantiram o êxito das nuvens IaaS (*Infrastructure as a Service*). Com este modelo, provedores podem oferecer a capacidade computacional ociosa dos seus *data centers* na forma de Infraestruturas Virtuais (IVs), diminuindo custos administrativos. Uma IV é composta por recursos virtuais de processamento e comunicação. Clientes de IaaS contratam IVs para diversas finalidades, sem a preocupação com qualquer tipo de manutenção nos recursos, totalmente a cargo do provedor [Mell and Grance 2011].

Um importante aspecto em nuvens computacionais é o provisionamento elástico de IVs. A variação na carga de processamento das aplicações hospedadas requer mecanismos de elasticidade que consigam ampliar, reduzir e reorganizar, em número e configuração, os recursos já alocados, respeitando o *Service Level Agreement* (SLA) estabelecido [Righi et al. 2015]. Ao invés de provisionar recursos de maneira estática, gerando desperdícios por provisionar sempre para o pior caso, o provisionamento elástico garante a quantidade de recursos necessária para a execução da carga de trabalho do momento.

Dentre as tarefas administrativas efetuadas para provisionar uma IV, destaca-se a utilização de algoritmos de alocação e realocação de recursos. O primeiro analisa a capacidade atual do *data center* para encontrar os componentes que melhor hospedem a IV requisitada [Oliveira and Koslovski 2015]. O segundo, adapta os recursos elásticos considerando todas as IVs já alocadas no *data center*, buscando a sua reorganização, quando necessária, para atender as requisições de elasticidade. Ambos os problemas são de categoria NP-Difícil [Zhu and Ammar 2006]. Tanto a alocação quanto a realocação de recursos são guiadas por políticas que representam o objetivo do provedor IaaS. Atualmente, é crescente a preocupação com o consumo energético dos servidores, visto que implica diretamente em custos administrativos bem como em aspectos de sustentabilidade. Estudos indicam que *data centers* nos Estados Unidos foram responsáveis por 3% do total de energia consumida em 2011 [Duan et al. 2016], com previsão de crescimento. Este consumo é principalmente associado ao resfriamento dos servidores, que pode chegar a 53% do seu total [Zhang et al. 2010].

Diante do exposto, neste trabalho, segue-se o princípio que provedores podem provisionar IVs elásticas simultaneamente reduzindo o consumo energético dos seus *data centers*. Entretanto, a imprevisibilidade de chegada e a definição não determinística das requisições elásticas são fatores críticos para provedores IaaS que buscam a diminuição do consumo energético. Uma simples requisição pode desbalancear um *data center*, criando gargalos de comunicação ou aumentando desnecessariamente o número de servidores ativos. As abordagens comumente aplicadas para consolidação de servidores, muitas vezes desrespeitam o SLA estabelecido com os clientes [Buyya et al. 2010, Duan et al. 2016]. Ainda, um estudo recente [Hinz et al. 2016] indicou que o consumo energético de CPUs pode ser utilizado para efetuar um rateamento mais justo do custo de provisionamento de servidores em nuvens IaaS. O custo individualizado de processamento (CPUs virtuais e operações de rede) das máquinas virtuais (MVs) é combinado com um custo proporcional das atividades de gerenciamento (operações dos hipervisores de MVs) para composição de um custo final, atribuindo somente a fração justa identificada.

Em suma, o algoritmo de realocação de IVs elásticas proposto neste trabalho considera o compartilhamento proporcional do uso de CPU de servidores hospedeiros para calcular o consumo individual, podendo desativar equipamentos ociosos e reorganizar os recursos virtualizados. O mecanismo tem como premissa o modelo de compartilhamento de custo proporcional [Hinz et al. 2016] pelo qual conclui-se que o aumento do gasto energético, na perspectiva do provedor, está diretamente ligado não somente ao número de servidores ativos para atender as IVs provisionadas, mas também a carga de processamento das MVs. A proposta é inovadora ao considerar o compartilhamento proporcional do uso de CPU das MVs, realizando uma consolidação de servidores que, além de atender ao SLA estabelecido, contabiliza a fração justa atribuída a cada cliente. Os resultados

experimentais apontam uma redução no consumo de energia sem diminuir a taxa de aceitação de novas requisições.

O restante deste trabalho está organizado da seguinte forma: a Seção 2 discute o provisionamento de IVs elásticas. A Seção 3 apresenta os trabalhos relacionados, enquanto a Seção 4 descreve a solução proposta. A análise experimental é discutida na Seção 5. Por fim, a Seção 6 apresenta as considerações finais.

### **Provisionamento de IVs Elásticas**

O cenário no qual o mecanismo proposto é aplicado constitui-se de um *data center* contendo diversas requisições de IVs já alocadas. O provedor que hospeda este *data center* recebe, ocasionalmente, requisições por parte de seus clientes, tanto para alocação quanto para elasticidade de recursos.

### **Mecanismos para Provisionamento de Elasticidade**

Tanto inquilinos quanto provedores usufruem dos benefícios da elasticidade. Para o inquilino, tais técnicas são usadas para controlar o desempenho da aplicação hospedada, mesmo na ocorrência de variações da carga de trabalho. No lado do provedor, a elasticidade pode ser utilizada para consolidar MVs em servidores, aumentando a taxa de aceitação de novas requisições de IVs.

As requisições de elasticidade provenientes de inquilinos podem ser de quatro tipos [Righi 2013]: (i) liberação, representando o desprovisionamento de um recurso virtual; (ii) redimensionamento, em que as capacidades virtuais são aumentadas ou diminuídas; (iii) replicação, em que um novo recurso virtual é criado a partir de um recurso base, comumente utilizado em contextos de balanceamento de carga de aplicações [Righi et al. 2015] e; (iv) migração, no qual todo o conteúdo de uma MV é transferido para outro hospedeiro. Este último mecanismo é favorável em tarefas administrativas, aplicado para reorganizar os recursos virtuais existentes. Ainda, requisições elásticas de redimensionamento e replicação podem ser convertidas em requisições de migração em cenários em que a capacidade residual do hospedeiro não for suficiente para o seu provisionamento. O redimensionamento é a abordagem mais indicada por ser a menos custosa em relação ao tempo de adaptação [Sharma et al. 2011], requisitando o aumento ou diminuição da configuração (*e.g.*, CPU, RAM, largura de banda).

### **Requisição de Provisionamento Elástico**

Ao requisitar uma IV, os clientes estabelecem um SLA com o provedor. Através dos SLIs (*Service Level Indicators*), o SLA estabelece características que devem ser respeitadas, tais como confiabilidade, custos, desempenho, segurança, localização geográfica, entre outras [Sauvé et al. 2005]. De forma complementar, o SLO (*Service Level Objective*) é responsável por representar estas restrições, permitindo uma forma de quantificá-las em termos do SLI. O SLO mede o desempenho do provedor de serviço de acordo com o SLA, trabalhando com métricas como vazão, disponibilidade, tempo de resposta e qualidade [Righi 2013]. Ou seja, caso ocorram picos de execução, algumas características do SLO podem ser violadas, tais como a disponibilidade do serviço ou o tempo de resposta.

Para que a qualidade do serviço hospedado na IV continue satisfatória, é necessária a aplicação de mecanismos de elasticidade, respeitando outras restrições que possam

estar presentes no SLA. No presente trabalho, além das requisições tradicionais de configuração dos recursos (vCPU, memória, armazenamento e largura de banda), o SLA inclui a definição do tempo máximo permitido para a migração de um recurso virtual. A definição foi incluída pois é de conhecimento comum que a migração de MVs induz a uma sobrecarga na aplicação hospedada [Akoush et al. 2010, Strunk 2012].

O inquilino/usuário tem liberdade para solicitar a reconfiguração da IV apoiado em mecanismos de monitoração e controle [Simões and Kamienski 2014, Pfitscher et al. 2013, Darolt et al. 2016]. Nos provedores de nuvens IaaS atuais, tais como *Amazon EC2*, *Microsoft Azure* e *Google Computing Engine*, a reconfiguração é realizada reativamente, respondendo ao SLI. O algoritmo proposto neste trabalho é agnóstico às métricas e ao mecanismo de monitoração e controle selecionado pelo inquilino.

## Trabalhos Relacionados

A literatura especializada relacionada com o escopo do presente trabalho compreende técnicas para alocação e realocação de IVs, mecanismos para provisionamento de elasticidade e políticas guiadas pela eficiência energética.

### Alocação e Realocação de IVs

Os algoritmos para alocação de IVs devem encontrar uma solução de mapeamento em tempo computacional aceitável, atendendo aos requisitos definidos no SLA e aos objetivos dos provedores. Oliveira *et al.* (2015) propuseram um algoritmo heurístico baseado em árvores para diminuir o tempo necessário para alocação, denominado VITreeM. Por sua vez, as soluções D-ViNE e R-ViNE [Chowdhury et al. 2009] buscaram a alocação com foco na garantia da rota mais curta entre os recursos virtuais. O mecanismo proposto na Seção 4 é agnóstico ao algoritmo de alocação utilizado. Ou seja, dentre as diversas soluções existentes [Fischer et al. 2013, Gong et al. 2014], o provedor pode selecionar o mecanismo que melhor representar os seus objetivos. Baseado na comparação realizada entre algumas abordagens [Oliveira and Koslovski 2016], VITreeM foi selecionado para compor o cenário experimental (Seção 5).

Quanto a realocação de IVs, um algoritmo destinado ao balanceamento de carga do *data center*, priorizando requisições que resultem em maior lucro, foi investigado em [Zhu and Ammar 2006], enquanto [Duan et al. 2016] propôs um algoritmo com mecanismo de predição baseado em fractais e utilizando *Ant Colony Optimization* como estratégia de realocação de modo a reduzir o custo energético. No presente trabalho (Seção 4), a redução do consumo energético do *data center* é viabilizada através da consolidação de MVs guiadas pelo princípio da elasticidade, prioritário na decisão. No SLA, além das definições tradicionais, o inquilino define o tempo máximo aceitável para uma migração (Seção 4.3), que representa o limiar de consolidação que o provedor pode aplicar.

### Mecanismos para Provisionamento Elástico

Tendo como principal motivação a diminuição do custo da IV alocada por parte do inquilino, Sharma *et al.* (2011) propuseram um método de elasticidade baseado em programação linear. Quando uma requisição de elasticidade é submetida, o provedor seleciona as instâncias de MVs com menor custo para provisionamento. Por fim, o mecanismo aplica migrações, para consolidação do *data center*, e replicações para balanceamento de carga nas aplicações. Por sua vez, Darolt e seus colegas exploraram a elasticidade dos re-

curso que compõem uma IV para otimizar o desempenho de aplicações *n-layers* [Darolt et al. 2016]. Foi observada a diminuição do tempo médio de processamento e do custo de provisionamento para o inquilino ao hospedar a aplicação em uma IV elástica com abordagem de reconfiguração reativa. A presente proposta é independente dos mecanismos para controle de limiares.

O provisionamento elástico de recursos com foco em processamento de alto desempenho foi proposto em [Righi et al. 2015]. O mecanismo provisiona MVs sempre que o desempenho da aplicação está aquém do esperado. Entretanto, inova ao antecipar a realização das solicitações de elasticidade, diminuindo o tempo de espera pela reconfiguração do serviço. O mecanismo proposto (Seção 4) pode ser incorporado a solução para reorganizar o *data center*, tendo potencial para reduzir o consumo de energia.

### **Eficiência Energética**

Alguns fatores relacionados com o impacto do provisionamento elástico em recursos de nuvens já foram investigados [Assuncao et al. 2016]. As simulações elucidaram que abordagens tradicionais para desativação de servidores devem ser substituídas por estratégias de reserva antecipada. Seguindo essa linha, a Seção 5 aborda a aplicação do mecanismo proposto em cenários com diferentes configurações de elasticidade e requisitos de migração. Quanto ao cálculo do custo de provisionamento guiado pelo consumo de energia ocasionado por processadores, Hinz *et al.* (2016) apresentaram um modelo para distribuição proporcional. Assim, inquilinos são tarifados de acordo com o seu consumo real, enquanto provedores são responsáveis pelos custos adjacentes. O modelo inovou o cálculo tradicionalmente aplicado por provedores públicos, e conseqüentemente foi incorporado ao algoritmo descrito na Seção 4. Por fim, é importante ressaltar que o diferencial deste artigo é a inserção do atendimento de requisições de elasticidade junto com o procedimento de realocação de IVs com foco na redução do consumo de energia, não encontrado em nenhum dos outros trabalhos.

### **EAVIRA**

O algoritmo proposto, denominado *Energy-Aware Virtual Infrastructure Reallocation Algorithm* (EAVIRA), objetiva o atendimento de requisições de reconfiguração de IVs, ao mesmo tempo que busca diminuir o consumo energético do *data center*. Entretanto, este objetivo necessita ser alcançado respeitando o SLA contratado pelo cliente que, neste contexto, adiciona o tempo máximo aceitável para a migração de um recurso como uma restrição. EAVIRA consiste nas etapas de: (i) definição da infraestrutura base; (ii) processamento das requisições de elasticidade; e (iii) realocação das IVs no *data center*. As etapas são discutidas nas próximas subseções.

### **Definição da Infraestrutura Base**

O ponto inicial do algoritmo é a obtenção de uma infraestrutura base (IB) para as migrações. Visto que a realocação é um problema NP-Difícil, essa infraestrutura representa apenas uma fração do *data center*, diminuindo o número de servidores candidatos para cálculo das possibilidades de realocação. Assim, somente os recursos participantes de uma IB são candidatos a receberem a migração de recursos virtuais. Para fins de comparação durante a análise experimental (Seção 5), propõem-se dois tipos de infraestruturas bases, denominadas IB-W e IB-SLA. A primeira é definida como a infraestrutura física

que hospeda a IV de maior peso no *data center*. A sua escolha é realizada pelo cálculo do custo de provisionamento. Considerando que uma IV é composta por um conjunto de MVs ( $R$ ) interconectadas por um conjunto de enlaces virtuais ( $E$ ), o custo de provisionamento  $C(IV)$  é definido por  $C(IV) = \sum_{i \in R} c(i) + \sum_{l \in E} c(l) \times len(p)$ , sendo  $c(\cdot)$  a representação da capacidade virtual solicitada de um recurso. Um enlace virtual  $l$  é hospedado sobre um caminho físico  $p$  no *data center*. Em suma, o custo é dado pelo somatório das capacidades provisionadas para cada recurso virtual da IV. A hipótese para esta abordagem é que a infraestrutura de maior impacto no *data center* corresponde a de maior dificuldade de realocação. Assim, a IB é composta pelos servidores e enlaces compromissados com o provisionamento da IV de maior custo (IB-W).

A segunda alternativa para composição de uma IB é realizada pelas informações do SLA (IB-SLA). Como o EAVIRA busca a consolidação de MVs para desativar servidores ociosos, MVs com restrição de migração (representado por  $t_{mig} = 0$ ) inviabilizam o desligamento dos servidores que as hospedam. Restringir o número de servidores candidatos a desativação pode representar um ponto crítico na reorganização de IVs e, consequentemente, um menor ganho na redução do consumo energético do *data center*. Assim, a IB-SLA é composta por todos os servidores físicos que hospedem ao menos uma MV com esta restrição de migração.

Para reconhecer alterações ocorridas no *data center* (e.g., chegada de novas requisições, reconfigurações), a definição da IB ocorre a cada execução do algoritmo. Entretanto, as opções de IB são aplicadas separadamente, ou seja, uma escolha do provedor.

### Processamento das Requisições Elásticas

Após a definição da IB, EAVIRA processa as requisições de desalocações, redimensionamentos, replicações e migrações, nesta ordem. A ordem da aplicação dos mecanismos foi definida com base no seu custo de realização [Galante and de Bona 2012]. A desalocação oportuniza novas possibilidades para realocação e consolidação. Dentre os três mecanismos de elasticidade, o redimensionamento é o que tem menor custo computacional, seguido da replicação e da migração [Sharma et al. 2011].

**Desalocação de IVs.** Consiste na liberação da capacidade de servidores e enlaces previamente reservada para hospedar as IVs.

**Redimensionamento de Recursos Virtuais.** EAVIRA considera o aumento ou diminuição da configuração de uma MV (CPU, RAM, etc) como uma operação de redimensionamento. Entretanto, caso a capacidade disponível no servidor que hospeda a MV a ser redimensionada não seja suficiente para atender a nova configuração, a operação é transformada em uma solicitação de migração, movendo a MV para um novo hospedeiro.

**Replicação de Recursos Virtuais** A replicação é um método comum quando a finalidade é o balanceamento de carga de aplicações compostas por um despachante de requisições, que contém a visão de todas as réplicas disponíveis, e que decide qual réplica atenderá uma nova requisição do usuário [Righi 2013] [Righi et al. 2015] [Darolt et al. 2016]. Em suma, é um processo que consiste na cópia da imagem de uma MV para o servidor de destino. Após a cópia, o sistema operacional hospedado e as aplicações são iniciados, ingressando no conjunto de recursos disponíveis. O tempo total para replicação ( $t(i)$ ) de uma MV  $i$  pode ser calculado por  $t(i) = \frac{D(i)}{b} + \beta$ , sendo  $D(i)$  o tamanho da imagem do sistema,  $b$  a largura de banda disponível para mover a MV até o hospedeiro destino e  $\beta$

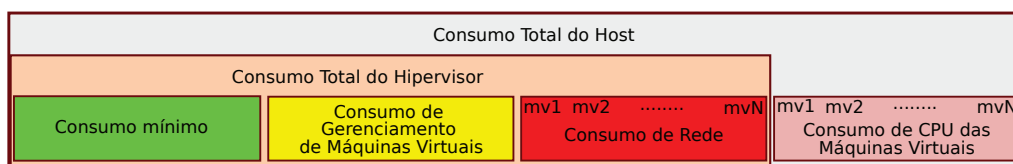
uma constante representando o tempo necessário para inicialização do sistema operacional no equipamento de destino [Sharma et al. 2011].

EAVIRA opta por replicar a MV no mesmo servidor hospedeiro, visto que a imagem do sistema já está disponível localmente. Quando o servidor não possui capacidade ociosa, o mecanismo seleciona os hospedeiros da IB como candidatos a receber o novo recurso replicado. Caso não tenha êxito, a última possibilidade de atendimento da solicitação é a replicação em um servidor disponível no *data center* que ainda não pertença a IB. Em caso de sucesso, este servidor será incorporado a IB, caso contrário, a solicitação não será atendida. O hospedeiro escolhido deve deixar as menores lacunas de recursos ociosos (abordagem *best-fit*) alinhada com a consolidação de recursos [Ruck et al. 2014].

### Realocação das IVs no *Data center*

Após as etapas de desalocações, redimensionamentos e replicações requisitadas pelos inquilinos, inicia-se a fase de sucessíveis migrações de IVs no *data center* buscando a consolidação dos servidores.

**Representação da Eficiência Energética.** EAVIRA busca diminuir o número de servidores ativos, bem como realizar a quantificação proporcional do custo guiado pelo consumo das MVs. Conforme representado pela Figura 1, o consumo energético de um servidor pode ser proporcionalmente compartilhado entre as MVs. O consumo total é composto por (i) uma fração mínima, necessária para manter o equipamento ativo; (ii) uma fração atribuída as operações de gerenciamento das MVs, efetuadas pelo hipervisores; (iii) a representação das operações de rede efetuadas pelas MVs; e (iv) o consumo efetivamente oriundo do processamento da carga de trabalho nas MVs. Como os itens (iii) e (iv) dependem da carga de trabalho aplicada, o provedor busca diminuir sua responsabilidade sobre as frações (i) e (ii). Quando não existem MVs hospedadas em um servidor, as frações (i) e (ii) são atribuídas ao provedor.



**Figura 1. Compartilhamento proporcional do consumo energético de um servidor hospedeiro [Hinz et al. 2016].**

Ao atender uma solicitação de redimensionamento de MV localmente, o provedor IaaS percebe a consolidação de seus servidores. O mesmo raciocínio é aplicado as solicitações de replicação: quando a operação é atendida localmente, as frações (i) e (ii) são diluídas entre os inquilinos. Entretanto, o escalonamento de uma migração para atender uma requisição de elasticidade representa um ponto crítico pois potencialmente um novo servidor será ativado. Embora EAVIRA busque a realocação com foco na diminuição do consumo energético, as operações de elasticidade são priorizadas para cumprir o SLA.

**Estimando o Tempo Total de uma Migração de MV.** Dentre os desafios para implementação da migração de MVs, a correta predição do tempo total de migração desponta como uma tarefa complexa. Existem vários parâmetros e métricas aplicáveis para essa predição, tais como a vazão do canal disponível, número de páginas modificadas e sobrecarga pré e pós migração [Akoush et al. 2010]. Neste cenário, [Strunk 2012] estabelece

um algoritmo para se obter uma estimativa do tempo de migração baseado no mecanismo de *live-migration* [Clark et al. 2005], descrito no Algoritmo 1.

---

**Algoritmo 1:** Estimativa do tempo de migração [Strunk 2012].

---

**input :**  $m_{th}, v_{th}, p_{th}, v_{mem}, b, d, l$   
**output:** Tempo de migração

```

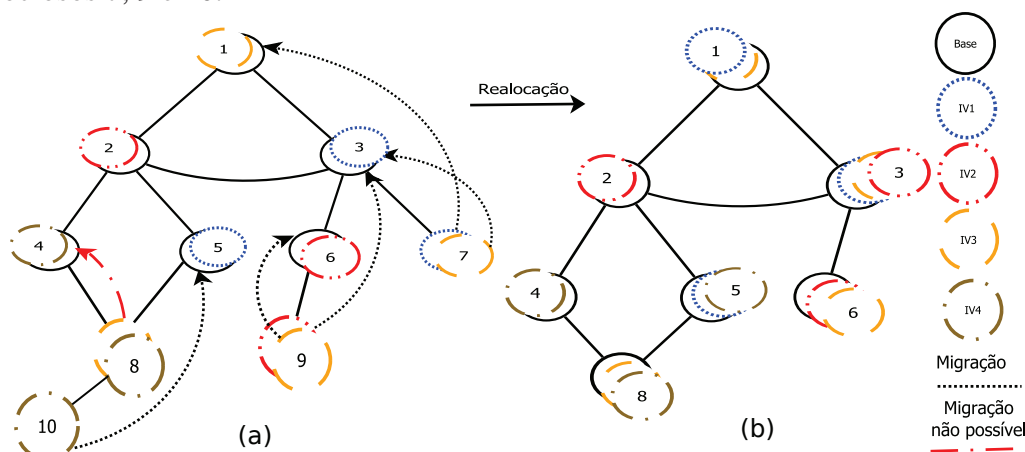
1 let  $v_0 = v_{mem}, v_{mig} = 0, t_{mig} = 0, t_{down} = 0;$ 
2 for  $i = 0; i < m_{th}$  do
3    $t_i = v_i/b;$ 
4    $v_{i+1} = t_i * d * l;$ 
5    $t_{mig} = t_{mig} + t_i;$ 
6    $v_{mig} = v_{mig} + v_i;$ 
7   if  $v_{mig} > v_{th} \vee \frac{v_{i+1}}{l} < p_{th}$  then
8     break;
9    $t_{down} = v_{i+1}/b;$ 
10   $t_{mig} = t_{mig} + t_{down};$ 
11 return  $t_{mig};$ 

```

---

O número máximo de iterações realizadas pelo algoritmo é dado por  $m_{th}$ , enquanto  $v_{th}$  indica a quantidade máxima de memória para ser migrada a cada iteração. Por sua vez,  $p_{th}$  representa a quantidade máxima aceitável de páginas sujas para uma iteração,  $v_{mem}$  a memória provisionada para a MV,  $d$  o número de páginas sujas no momento da execução do algoritmo e  $l$  o tamanho de cada página. Quanto a configuração da rede do *data center*,  $b$  representa a largura de banda disponível para migração. Iterativamente, o algoritmo quantifica a porção de memória que deve ser copiada nas iterações futuras (linhas 3 e 4). Se os limiares de páginas sujas ou memória máxima forem ultrapassados (linha 7), o algoritmo passa para a fase de *stop-and-copy* (linha 9), em que a MV encerra sua execução e o restante da memória é copiada para o servidor de destino.

**Algoritmo e Exemplo de Realocação.** Para exemplificar as migrações executadas por EAVIRA, a Figura 2 retrata 5 migrações realizadas com sucesso, desativando os servidores ociosos 7, 9 e 10.



**Figura 2.** Cenário de realocação de IVs.

As arestas tracejadas indicam a ocorrência de migrações. Após o sucesso de uma migração, os enlaces virtuais são realocados para o caminho mais curto entre os recursos conectados à MV migrada, diminuindo o custo de provisionamento (discutido na Seção 4.1). No caso de impossibilidade na migração, representado pela tentativa com a MV pertencente à IV3 no hospedeiro 8, a IB será expandida para comportar este nó (grafo Figura 2(b)). Na próxima iteração o hospedeiro 8 poderá ser alvo de outras migrações. Além disso, as migrações das MVs de um nó são realizadas em conjunto, ou seja, basta



que uma migração seja impossibilitada para que a IB seja expandida. O procedimento será repetido para todas as MVs ainda não analisadas (busca em largura). O Algoritmo 2 representa o mecanismo implementado na forma de pseudo-código.

---

### Algoritmo 2: Etapa de realocação

---

```

input :  $BI(V, E)$ 
output: Número de migrações realizadas
1   $resources = get\_resources(BI)$ ;
2   $nodes = \{\forall neighb; neighb \in adj(node) \wedge node \in resources\}$ ;
3   $visited = \{\}$ ;
4   $total\_migrations = 0$ ;
5  for  $\forall node; node \in nodes$  do
6       $nodes = nodes - \{node\}$ ;
7       $visited = visited \cup \{node\}$ ;
8       $resources = get\_resources(BI)$ ;
9       $nodes = nodes \cup \{\forall neighb; neighb \in adj(node) \wedge neighb \notin resources \wedge neighb \notin$ 
       $nodes \wedge neighb \notin visited\}$ ;
10      $rollback\_list = \{\}$ ;
11      $migrations = 0$ ;
12     for  $\forall vnode \in virtual\_resources(node)$  do
13          $resources = get\_resources(BI, vnode.type)$ ;
14         if  $\neg migrate(vnode, resources)$  then
15              $failed = True$ ;
16             break;
17         else
18              $migrations = migrations + 1$ ;
19              $rollback\_list = rollback\_list \cup \{vnode\}$ ;
20     if  $failed$  then
21          $rollback(rollback\_list)$ ;
22          $rollback\_list = \{\}$ ;
23     else
24          $total\_migrations = total\_migrations + migrations$ ;
25 return  $total\_migrations$ ;

```

---

As linhas 1, 2 e 3 inicializam as listas iniciais da busca em largura a partir das folhas da IB. Os novos nós são inseridos iterativamente conforme os forem visitados (linha 9). Para cada nó pertencente a esta lista (linha 5), busca-se a migração de todos os seus recursos virtuais (linha 12-19). Em caso de falha, todo o processo é desfeito para o servidor em análise (linhas 20-22). Ao final do algoritmo (linha 25), é retornado o número de migrações realizadas nesta iteração. É importante ressaltar que em tempo de análise de possibilidades, nenhuma migração é efetivada, apenas analisada.

Ao término da execução, a fração ativa de recursos no *data center* será menor ou igual à fração anterior, visto que, no pior caso, a infraestrutura física não sofrerá alterações, e, no melhor caso, o número de servidores ativos será reduzido por conta das consolidações. Havendo consolidações, mantém-se a mesma quantidade de recursos virtuais alocados, porém, tem-se economia de energia com o desligamento de recursos físicos.

Em resumo, para que a migração seja possível, é necessário que: (i) a restrição de tempo máximo para a migração do SLA contratado seja respeitada; (ii) exista um hospedeiro de destino com recursos disponíveis; e (iii) todos os recursos virtuais conectados à MV migrada tenham seus enlances reconectados para o hospedeiro destino. Caso contrário, a migração não é realizada. EAVIRA propõe o atendimento de requisições elásticas reduzindo o consumo energético do *data center*.

## Análise Experimental

A análise experimental segue o plano de testes resumido na Tabela 1. EAVIRA e um simulador de eventos discretos foram implementados em Python<sup>1</sup>. Para compor o cenário experimental, o algoritmo VITreeM [Oliveira and Koslovski 2015], discutido na Seção 3, foi selecionado para alocação *online* de requisições. O estudo com outros mecanismos é indicado como trabalho futuro. Os testes apresentam os resultados com VITreeM (sem elasticidade) e com EAVIRA (elasticidade e foco na redução do consumo energético).

**Tabela 1. Configurações utilizadas nos experimentos.**

Algoritmos	VITreeM – EAVIRA
Infraestrutura de base	IB-W – IB-SLA
Seleção dos Candidatos	best-fit – worst-fit
Requisições de IVs	fat-tree – n-layers – VPC – mixed
Recursos virtuais restritos (%)	10% – 40% – 80%

As métricas escolhidas para identificar o impacto da política de redução de energia são: (i) número de operações, contabilizando as operações em IVs (alocação e desalocação) e de recursos virtuais (desalocação, reconfiguração, replicação e migração); (ii) taxa de aceitação (TA), representando o percentual de IVs alocadas com sucesso; e (iii) consumo de energia, calculado a partir do modelo de custos de [Hinz et al. 2016]. Segundo as características dos algoritmos VITreeM e EAVIRA, foram identificados os quatro principais elementos (Tabela 1) que impactam no consumo de energia de um *data center*.

O *data center* modelado é constituído de servidores idênticos, totalizando 432, interconectados por 90 switches, seguindo padrões de projeto da Cisco [Cisco 2007]. Optou-se por um *data center* de grande porte para evidenciar melhor os aspectos de consolidação da proposta, visto que quanto menor o *data center* ficam mais restritas as operações de consolidação. Cada servidor é composto por dois processadores com 12 núcleos cada, 256GB de memória, 1TB de armazenamento e uma interface de rede de 1Gbps. A relação de carga de processamento versus consumo de energia destes servidores foram obtidos em [Hinz et al. 2016]. O provedor de IaaS simulado não dispõe de locação por pacotes de recursos, deixando livre ao seu inquilino a composição de suas IVs. Portanto, definiu-se limiares individuais para cada recurso, sem vínculo entre eles. Dentre estes limiares, todas as combinações são possíveis. Assim, as IVs foram compostas por vCPUs (de 1 a 8), RAM (de 1 a 32GB), capacidade de armazenamento (de 8 a 128), enlances de redes (de 1 a 100Mbps) e tempo de SLA para migração (de 1 a 600s).

Por definição, o EAVIRA inicia seu provisionamento identificando a IB (Seção 4.1) que servirá como alvo para as requisições elásticas. A escolha da IB afeta diretamente o número total de nós físicos utilizados após o atendimento de todas as requisições do período. Na Tabela 1, a característica infraestrutura de base pode ser IB-SLA ou IB-W, indicando o critério de escolha. O terceiro componente da Tabela 1 trata da forma de seleção dos candidatos, *best-fit* ou *worst-fit*. Dentre um conjunto de solicitações, a configuração *best-fit* seleciona o candidato que deixará o menor residual de recurso disponível, após seu provisionamento, e, o *worst-fit*, escolhe aquele que deixará o maior residual. O tipo de requisição de IVs especifica a topologia escolhida. Foram realizados testes com quatro categorias: *fat-tree*, *Virtual Private Cloud (VPC)*, *n-layers* e *mixed*. Requisições do tipo *fat-tree* são topologias que organizam os *switches* em camadas, sendo os servidores virtuais posicionados nas folhas da árvores. É uma topologia de propósito geral que

<sup>1</sup>Disponível em: <https://bitbucket.org/denivyruck/eavira>

utiliza de maneira eficiente os recursos de comunicação oferecidos [Al-Fares et al. 2008]. Recentemente, a Amazon<sup>2</sup> passou a oferecer a topologia VPC, constituída de uma rede virtual controlada completamente pelo próprio inquilino. O terceiro tipo, as topologias *n-layers*, foram incluídas por representarem aplicações que operam em diferentes camadas, tais como aplicações *web* [Darolt et al. 2016]. Por fim, *mixed* é constituído de um conjunto de IVs cujo as topologias podem ser *fat-tree*, VPC e/ou *n-layers*.

O último componente dos experimentos descreve o percentual de recursos virtuais restritos dentre o conjunto de solicitações de realocação de IVs requisitadas. Quanto maior o percentual de restrição, maior o número de MVs que não admitem migração ( $t_{mig} = 0$ , Seção 4) e menor o grau de maleabilidade na alocação de recursos. Foram feitos testes com percentuais de restrição em 10%, 40% e 80%, porém, neste trabalho, escolheu-se por discutir somente os resultados do pior caso (80%). Os gráficos com os resultados com 10% e 40% encontram-se disponíveis no repositório do EAVIRA.

### Análise dos Resultados

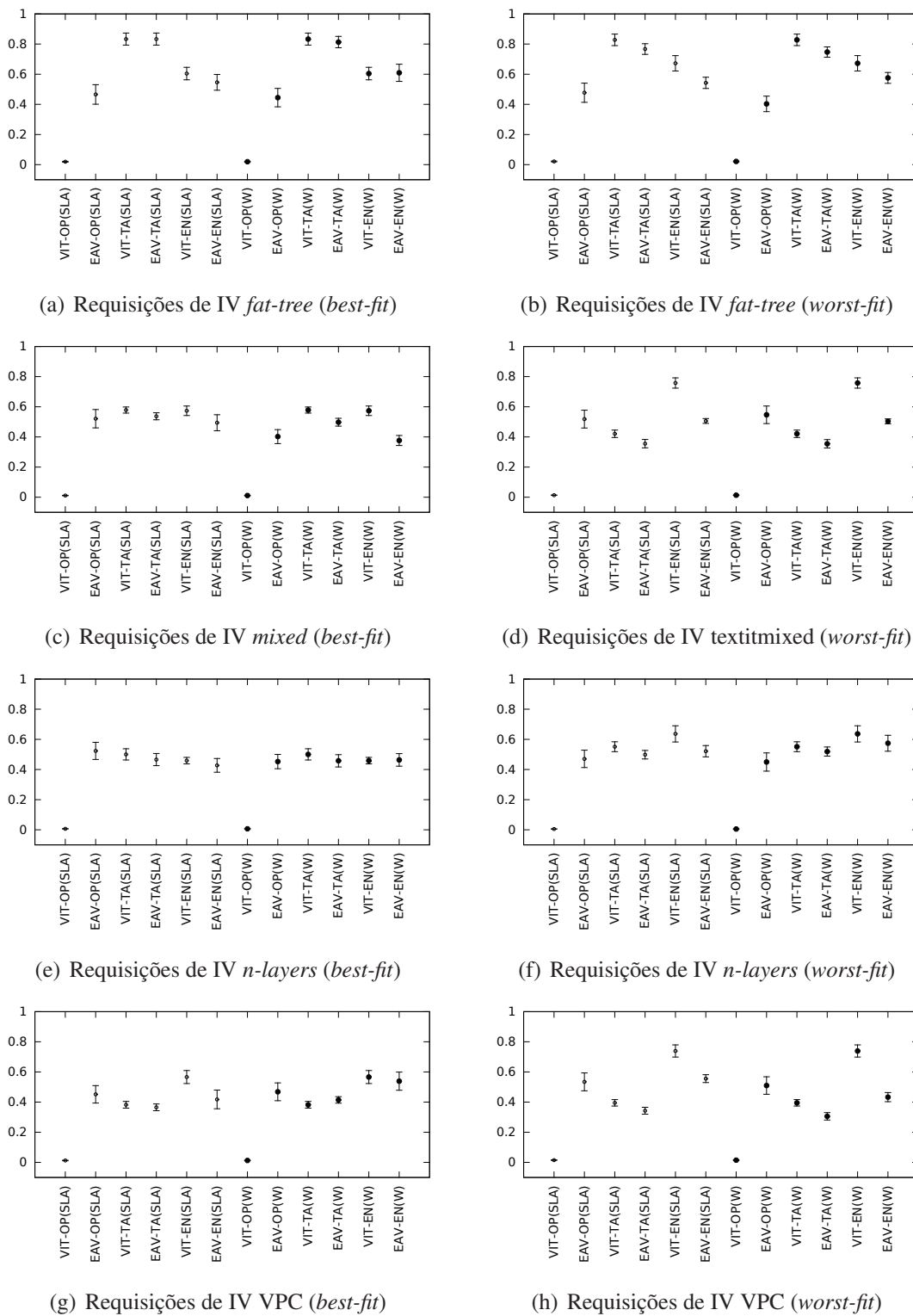
O conjunto de resultados obtidos é oriundo do cruzamento das características descritas na Tabela 1. Utilizou-se, simultaneamente, um total de 76 servidores da plataforma *Grid'5000*<sup>3</sup>. Os gráficos apresentados na Figura 3 correspondem aos resultados com o percentual de recursos virtuais restritos em 80%. A amostra tem tamanho 100, sendo que os pontos são as médias dos valores e as barras o desvio padrão com um intervalo de confiança de 95%. Cada gráfico possui os valores normalizados por métrica, iniciando por número de operações com VITreeM, representado por VIT-OP, seguido pelo número de operações do EAVIRA (EAV-OP). A mesma representação e ordem foi utilizada para as demais métricas, taxa de aceitação (TA) e consumo de energia (EN), respectivamente identificadas por VIT-TA, EAV-TA, VIT-EN e EAV-EN.

Quanto a análise das métricas, para OP, espera-se quantificar o número de operações elásticas efetuadas pelo EAVIRA, tomando como base o número mínimo necessário de operações efetuadas pelo VITreeM. Para TA, quanto maior o valor, melhor, tendo em vista que este corresponde a um melhor aproveitamento dos recursos físicos. Para EN, deseja-se a redução do consumo, portanto valores menores são buscados. As primeiras seis barras de cada gráfico correspondem aos resultados com a infraestrutura de base IB-SLA e as demais com IB-W. Finalmente, os gráficos da esquerda descrevem os resultados com o algoritmo de seleção de candidatos *best-fit* e os da direita com *worst-fit*.

Na Figura 3 pode-se observar que VIT-OP, por representar a linha de base para comparações, possui, para todos os dezesseis casos analisados, média igual a zero. Ao comparar os valores de VIT-OP e EAV-OP, pode-se concluir que o número de operações elásticas efetuadas pelo EAVIRA varia entre 40% e 60%, independente do tipo de requisição, da escolha da infraestrutura de base ou da forma de seleção dos candidatos. Os resultados permitem ainda concluir que a escolha entre IB-SLA ou IB-W, para o pior caso de 80% de recursos restritos, gera pouco impacto nos resultados. A diferença dos valores obtidos com IB-SLA (primeiras 6 colunas) comparados com o valor correspondente de IB-W (últimas 6 colunas) são inferiores a 10%.

<sup>2</sup><https://aws.amazon.com/vpc/>

<sup>3</sup><http://www.grid5000.fr>



**Figura 3. Resultados normalizados das métricas número de operações (OP), taxa de aceitação (TA) e consumo de energia (EN) obtidas nas infraestruturas de bases, IB-SLA (SLA) e IB-W (W). Nestes testes, 80% das IVs não permitem migração, caso que reduz a possibilidade de realocação.**

Quanto a taxa de aceitação (TA), o VITreeM e o EAVIRA são equivalentes em todos os casos. Ou seja, o atendimento das requisições elásticas não gera prejuízos ao provedor. No entanto, o impacto do tipo de requisição na TA é evidente, enquanto *fat-tree* atinge taxas próximas a 80%, melhor resultado, VPC obtém aproximadamente 40%. Por fim, para a métrica de consumo de energia (EN), cujo objetivo é a redução dos valores consumidos, o EAVIRA é igual ou inferior ao VITreeM. Os consumos são equivalentes em 7 dos 16 casos, sendo 5 deles nos resultados com *best-fit* (gráficos a esquerda). O EAVIRA tem consumo inferior ao VITreeM com *best-fit* para requisições *mixed/IB-W* e *VPC/IB-SLA*. A redução de consumo de energia com *mixed/IB-W* (Figura 3(c)) foi próximo a 20%. Com *worst-fit* (gráficos a direita), o EAVIRA surpreende obtendo resultados positivos em 7 dos 8 casos analisados, ocasionando uma única equivalência, com requisições *mixed/IB-W*. Pode-se ainda destacar o gráfico da Figura 3(h) com uma diferença de 50%. A redução oportunizada por EAVIRA é o resultado das consolidações de servidores, combinada com o cálculo proporcional de custo energético.

Os resultados indicam que o algoritmo EAVIRA flexibiliza o provisionamento de recursos com operações elásticas, beneficiando provedores e inquilinos, e ainda permite a redução dos custos com energia elétrica. A redução do consumo de energia, sem prejuízo a outras métricas, é de interesse do provedores, de inquilinos e da própria sociedade (sustentabilidade). No que se refere ao objeto principal de estudo, o consumo de energia, o EAVIRA é no mínimo equivalente ao VITreeM e, no melhor caso, 50% mais eficiente.

### Considerações & Trabalhos futuros

Em nuvens computacionais, a elasticidade é uma característica essencial, tanto para provedores quanto para inquilinos. Enquanto provedores apoiam-se em operações de realocação de IVs para gerenciar a estrutura física, inquilinos as usam para adequar a disponibilidade de recursos virtuais as suas necessidades. A alocação e realocação de recursos são problemas NP-Difícil [Zhu and Ammar 2006]. Considerando os custos administrativos e o impacto ambiental, a redução do consumo energético de *data centers* é uma preocupação de provedores, de inquilinos e da sociedade. Neste contexto, este trabalho propõe um algoritmo de realocação de IVs elásticas, o EAVIRA, com foco na redução do consumo energético de *data centers*. EAVIRA baseia-se em consolidação de MVs e no modelo de compartilhamento de custo proporcional de [Hinz et al. 2016]. Os resultados obtidos foram analisados com base em três métricas: número de operações, taxa de aceitação e consumo de energia; e indicam que o EAVIRA: (i) efetua um maior número de operações, o que é natural, afinal admite operações elásticas; (ii) é equivalente ao VITreeM em taxa de aceitação, ou seja, não reduz o número de inquilinos; e (iii) é equivalente ou superior ao VITreeM na economia de energia, obtendo ganhos de até 50%.

O EAVIRA é agnóstico ao mecanismo de alocação de recursos para hospedar IVs. Assim, um estudo com outras soluções é um tema para trabalhos futuros. A função objetivo de realocação de EAVIRA é guiada pelo consumo de energia. Uma investigação futura compreende abordagens multi-objetivos, acrescentando outras perspectivas.

### Agradecimentos

Os experimentos foram conduzidos na plataforma Grid'5000.

### Referências

Akoush, S., Sohan, R., Rice, A., Moore, A. W., and Hopper, A. (2010). Predicting the Performance of Virtual Machine Migration. In *MASCOTS, 2010 IEEE Int. Symp. on*, pages 37–46.

- Al-Fares, M., Loukissas, A., and Vahdat, A. (2008). A Scalable, Commodity Data Center Network Architecture. *SIGCOMM Comput. Commun. Rev.*, 38(4):63–74.
- Assuncao, M. D., Lefevre, L., and Rossignaux, F. (2016). On the impact of advance reservations for energy-aware provisioning of bare-metal cloud resources. In *Int. Cont. on Network and Service Management*.
- Buyya, R., Beloglazov, A., and Abawajy, J. H. (2010). Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *CoRR*, abs/1006.0308.
- Chowdhury, N., Rahman, M. R., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783–791. IEEE.
- Cisco (2007). Cisco Data Center Infrastructure 2.5 Design Guide. Technical report, Cisco Systems, Inc.
- Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., and Warfield, A. (2005). Live migration of virtual machines. In *Proc. of the Symp. on Networked Systems Design & Implementation*.
- Darolt, D., Souza, F., and Koslovski, G. (2016). Explorando a elasticidade de nuvens iaas para reconfigurar dinamicamente aplicações n-camadas. *Revista Brasileira de Computação Aplicada*, 8(2):2–15.
- Duan, H., Chen, C., Min, G., and Wu, Y. (2016). Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Generation Computer Systems*.
- 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.
- Galante, G. and de Bona, L. (2012). A survey on cloud computing elasticity. In *Utility and Cloud Computing (UCC), 2012 IEEE Fifth International Conference on*, pages 263–270.
- Gong, L., Wen, Y., Zhu, Z., and Lee, T. (2014). Toward profit-seeking virtual network embedding algorithm via global resource capacity. In *INFOCOM, 2014 Proceedings IEEE*, pages 1–9. IEEE.
- Hinz, M., Miers, C. C., Pillon, M. A., and Koslovski, G. P. (2016). Um modelo de custo para nuvens iaas baseado no consumo de energia de máquinas virtuais. In *Simpósio Brasileiro de Sistemas de Informação*.
- Mell, P. and Grance, T. (2011). The NIST definition of cloud computing. NIST SP 800-145, National Institute of Standards and Technology.
- Oliveira, R. and Koslovski, G. (2015). VITreeM - um algoritmo baseado em árvores para alocação de infraestruturas virtuais. In *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.
- Oliveira, R. and Koslovski, G. (2016). A tree-based algorithm for virtual infrastructure allocation with joint virtual machine and network requirements. *International Journal of Network Management*. nem.1958.
- Pfitscher, R., Pillon, M., and Obelheiro, R. (2013). Diagnóstico do Provisionamento de Recursos para Máquinas Virtuais em Nuvens IaaS. In *Simp. Brasileiro de Redes de Computadores e Sistemas Distribuídos*.
- Righi, R. (2013). Elasticidade em cloud computing: conceito, estado da arte e novos desafios. *Revista Brasileira de Computação Aplicada*, 5(2):2–17.
- Righi, R., Rodrigues, V., da Costa, A., Galante, G., Bona, L., and Ferreto, T. (2015). Autoelastic: Automatic resource elasticity for high performance applications in the cloud. *Cloud Computing, IEEE*, PP(99):1–1.
- Ruck, D. B., Oliveira, R., and Koslovski, G. P. (2014). Comparação de algoritmos para alocação de Infraestruturas Virtuais. *Revista Brasileira de Computação Aplicada*, 6(2):98–112.
- Sauvé, J., Marques, F., Moura, A., Sampaio, M., Jornada, J., and Radziuk, E. (2005). Sla design from a business perspective. In *Int. Workshop on Distributed Systems: Operations and Management*.
- Sharma, U., Shenoy, P., Sahu, S., and Shaikh, A. (2011). A cost-aware elasticity provisioning system for the cloud. In *31st Int. Conf. on Distributed Computing Systems*, pages 559–570.
- Simões, R. and Kamienski, C. (2014). Gerenciamento de elasticidade em nuvens privadas e híbridas. In *Anais do XII Workshop de Computação em Clouds e Aplicações, WCGA*, pages 67–78.
- Strunk, A. (2012). Costs of virtual machine live migration: A survey. In *Services (SERVICES), 2012 IEEE Eighth World Congress on*, pages 323–329.
- Zhang, Q., Cheng, L., and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18.
- Zhu, Y. and Ammar, M. H. (2006). Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM*, pages 1–12.