Mitigando Ataques de Egoísmo e Negação de Serviço em Nuvens via Agrupamento de Aplicações

Daniel Stefani Marcon, Miguel Cardoso Neves, Rodrigo Ruas Oliveira, Luciana Salete Buriol, Luciano Paschoal Gaspary, Marinho Pilla Barcellos

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

{daniel.stefani, mcneves, ruas.oliveira, buriol, paschoal, marinho}@inf.ufrgs.br

Resumo. *Na computação em nuvem, locatários consomem, sob demanda, recursos de hardware e software oferecidos por um provedor remoto. Entretanto, o compartilhamento da rede interna da nuvem por todos os locatários, aliado à falta de isolamento entre fluxos de dados decorrente do uso dos protocolos TCP e UDP, possibilita a ocorrência de ataques de egoísmo e negação de serviço. Os algoritmos de alocação atuais não impedem que a disponibilidade dos recursos de rede seja afetada por ataques. Este artigo propõe uma estratégia para a alocação de aplicações de locatários que visa mitigar o impacto de ataques de egoísmo e negação de serviço na rede interna da nuvem. A ideia chave, inédita na literatura científica, consiste no agrupamento de aplicações em infraestruturas virtuais considerando níveis de confiança mútua entre os locatários. Resultados de avaliações demonstram que a estratégia proposta é capaz de oferecer proteção contra ataques de egoísmo e negação de serviço com pouco ou nenhum custo extra.*

Abstract. *Cloud computing is a model where tenants consume on-demand hardware and software resources from a remote provider. However, the sharing of the internal network by all tenants, combined to the lack of data flow isolation due to the use of TCP and UDP, allows the occurrence of selfish and denial of service attacks. The current allocation algorithms do not prevent the availability of network resources to be affected by such attacks. In this paper, we propose a strategy for the allocation of tenants applications, which aims at mitigating the impact of selfish and denial of service attacks in the cloud internal network. The key, novel idea is to group applications into virtual infrastructures considering the mutual trust between pairs of tenants. Evaluation results show that the proposed strategy is able to offer protection against attacks of selfishness and DoS with little or no extra cost.*

1. Introdução

Computação em nuvem é um paradigma que possibilita aos locatários consumirem, sob demanda e de maneira elástica, recursos de hardware e software oferecidos por um provedor remoto. Nesse modelo, os provedores, com a finalidade de reduzir custos operacionais e oferecer escalabilidade de acordo com a demanda, implementam *datacenters* de nuvem como ambientes compartilhados altamente multiplexados, com diferentes aplicações coexistindo sobre os mesmos recursos físicos [Armbrust et al. 2010].

 ©2012 SBC — Soc. Bras. de Computação

Entretanto, não há uma abstração ou mecanismo disponível para capturar os requisitos de rede das interações entre as máquinas virtuais (VMs) alocadas [Guo et al. 2010]. Ademais, mecanismos como o controle de congestionamento do TCP e suas variantes (incluindo TFRC e DCCP), apesar de serem escaláveis e proporcionarem alta utilização dos recursos, não isolam de forma robusta o tráfego de dados de aplicações distintas [Abts and Felderman 2012]. Dessa forma, as VMs de uma mesma aplicação se comunicam em um ambiente (rede da nuvem) não controlado, compartilhado por todos os locatários. Consequentemente, locatários egoístas podem implementar suas aplicações de forma a levar vantagem indevida sobre outros usuários, através da utilização de versões do TCP com controle de congestionamento mais brando, e locatários maliciosos podem lançar ataques de negação de serviço, por exemplo gerando tráfego espúrio dentro da nuvem [Shieh et al. 2011].

Tais ataques prejudicam tanto locatários quanto provedores. Os locatários pagam pela utilização dos recursos contratados mesmo quando estão impossibilitados de utilizá-los devido a um ataque. Já os provedores possuem perda de receita [Greenberg et al. 2009], visto que a falta de disponibilidade da rede, causada por ataque de DoS, reduz o *throughput* da nuvem [Shieh et al. 2011].

Visando alocar recursos com garantias de disponibilidade de rede, as abordagens presentes na literatura focam em consolidação de VMs [Breitgand and Epstein 2012, Meng et al. 2010, Wang et al. 2011] ou no compartilhamento da rede baseado em pesos [Shieh et al. 2011, Lam and Varghese 2010]. Entretanto, elas não impedem ataques de egoísmo e de DoS, visto que a utilização dos recursos da rede de uma aplicação depende das demais aplicações da nuvem.

Nesse contexto, o presente artigo propõe uma estratégia de alocação de recursos para provedores de IaaS (*Infrastructure as a Service*), apresentada como um modelo de otimização, que visa mitigar ataques de egoísmo e de comportamento malicioso na rede interna da nuvem. Diferentemente de trabalhos anteriores, investiga-se uma estratégia baseada no agrupamento de aplicações em infraestruturas virtuais[1] (VIs), que são capazes de oferecer um certo nível de isolamento entre aplicações. São estudadas diferentes estratégias de agrupamento de locatários, considerando o nível de confiança mútuo e os requisitos de rede (largura de banda) das suas aplicações.

O agrupamento visa prover isolamento do tráfego de rede entre aplicações provenientes de locatários mutuamente não confiáveis. Além disso, o agrupamento permite fornecer isolamento entre o tráfego de aplicações com requisitos de largura de banda distintos, minimizando a interferência nociva que pode ser causada ao misturar tráfego com diferentes características [Shieh et al. 2011, Abts and Felderman 2012]. Além da proteção contra os ataques supracitados, o isolamento/agrupamento propiciado pela estratégia proposta pode ser benéfico contra outros tipos de ataques, como ataques contra a privacidade dos dados dos locatários [Ristenpart et al. 2009]. Os mesmos serão explorados em trabalhos futuros. O presente artigo tem as seguintes contribuições:

- discussão e avaliação do impacto de ataques de egoísmo e de negação de serviço no contexto de redes internas de nuvens;
- proposta de uma estratégia inédita para alocação de recursos, baseada no agrupamento de aplicações em VIs considerando níveis de confiança mútua

[1] O termo infraestrutura virtual é utilizado para representar uma rede virtual com as suas máquinas virtuais.

 ©2012 SBC — Soc. Bras. de Computação

entre os locatários, e aplicável a diferentes topologias de datacenters, tais como Fat-Tree [Greenberg et al. 2009] e VL2 [Al-Fares et al. 2008];
- avaliação analítica da estratégia proposta perante diferentes níveis de agrupamento e a comparação da mesma com um esquema padrão sem agrupamento.

O restante desse artigo está organizado da seguinte maneira. A Seção 2 apresenta os principais trabalhos relacionados. A Seção 3 define o modelo de ataque considerado e a Seção 4 define as formulações básicas utilizadas no restante do artigo. Na Seção 5, a estratégia de alocação de recursos é apresentada. Na Seção 6, o modelo de alocação proposto é avaliado e, por fim, as considerações finais são apresentadas na Seção 7.

2. Trabalhos Relacionados

As redes internas das nuvens atuais são compartilhadas por todos os locatários, honestos ou não. Apesar disso, não há um mecanismo disponível para capturar e controlar os recursos de rede utilizados pelas VMs alocadas [Grobauer et al. 2011].

Nesse sentido, Liu [Liu 2010] descreve como um atacante pode obter informações sobre a topologia da rede da nuvem, com a finalidade de realizar um ataque de negação de serviço. Já Ristenpart et al. [Ristenpart et al. 2009] conduziram um estudo de caso na Amazon EC2 (*Elastic Compute Cloud*), mostrando diversas vulnerabilidades do ambiente. Os autores relatam que não possuem nenhum dado privilegiado da nuvem, tendo conhecimento apenas das informações divulgadas publicamente pela Amazon. Desse modo, eles assumem que os atacantes são clientes comuns da nuvem. Eles demonstram que a EC2 pode ser mapeada de acordo com as suas zonas de disponibilidade e que é possível verificar a co-residência entre VMs, inclusive listando diversas possibilidades de efetuar tal verificação. O referido trabalho mostra que um locatário egoísta e/ou malicioso, conforme abordado no presente artigo, pode realizar diversos tipos de ataques à nuvem, inclusive de DoS na rede.

Apesar dessas vulnerabilidades, os algoritmos de alocação de recursos empregados atualmente em nuvens utilizam *round-robin* entre servidores ou entre *racks*, considerando somente recursos computacionais (processamento, memória e armazenamento) [Kitsos et al. 2012]. Contudo, torna-se necessário considerar também os recursos de rede, a fim de mitigar ataques de egoísmo e de comportamento malicioso.

O problema de segurança na rede interna da nuvem não é facilmente resolvível. Nesse contexto, a alocação de recursos em nuvens ciente dos recursos de rede representa um grande desafio de pesquisa. As abordagens presentes na literatura focam em consolidação de VMs ou no compartilhamento da rede por meio de pesos. Abordagens baseadas em consolidação de VMs [Meng et al. 2010, Wang et al. 2011, Breitgand and Epstein 2012] propõem que a alocação seja executada por meio da adaptação do problema de otimização *bin packing* [Goel and Indyk 1999]. Contudo, elas não impedem que o tráfego de aplicações egoístas ou maliciosas interfira no tráfego de aplicações honestas. Já Shieh et al. [Shieh et al. 2011] e Lam e Varghese [Lam and Varghese 2010] propõem um esquema de compartilhamento de largura de banda baseado em pesos. Esse esquema, porém, não impede que aplicações maliciosas realizem ataques de egoísmo e de negação de serviço na rede, visto que a largura de banda utilizada por uma aplicação na rede é dependente das outras aplicações da nuvem.

 ©2012 SBC — Soc. Bras. de Computação

Em outra gama de trabalhos, é explorada a alocação de tarefas em grades computacionais respeitando restrições de confiança entre a tarefa e os recursos da grade computacional [Costa and Carmo 2007]. A estratégia apresentada nesse artigo, em uma perspectiva distinta, explora as relações de confiança mútuas entre os locatários da nuvem.

De forma geral, os recursos da rede interna de *datacenters* frequentemente representam o gargalo quando comparados aos recursos computacionais [Greenberg et al. 2009]. Esse gargalo, aliado à falta de mecanismos para o controle do compartilhamento de recursos na rede, tende a facilitar ataques de egoísmo e comportamento malicioso. Nesse sentido, a estratégia apresentada nesse artigo visa mitigar tais ataques, sendo ciente tanto de recursos de rede quanto de recursos computacionais. Adicionalmente, a utilização de VIs diminui a eficácia dos métodos utilizados por Ristenpart et al. [Ristenpart et al. 2009] para a verificação de co-residência, visto que aplicações mutuamente não confiáveis são alocadas em redes virtuais distintas.

3. Modelo de Ataque

Similarmente à Ristenpart et al. [Ristenpart et al. 2009], considera-se que um locatário egoísta e/ou malicioso possui os mesmos privilégios dos demais locatários da nuvem. Usuários egoístas utilizam versões mais agressivas do TCP (*i.e.*, versões fora dos padrões, cujo código foi modificado para realizar controle de congestionamento mais brando) com o objetivo de aumentar a vazão de rede das suas VMs. Já locatários maliciosos tem por objetivo realizar um ataque de DoS em um alvo previamente definido. Nesse caso, o locatário emprega os métodos descritos por Ristenpart et al. [Ristenpart et al. 2009] para posicionar as suas VMs próximas ao alvo. Consequentemente, as VMs desse locatário podem enviar uma quantidade suficiente de tráfego para sobrecarregar o destinatário (alvo), um enlace pertencente ao caminho ou um gargalo da rede [Jensen et al. 2009]. No presente trabalho, considera-se somente o ataque de um locatário. Entretanto, ataques de conluio são possíveis, através de múltiplas identidades.

Os ataques de egoísmo e de DoS são efetuados através do aumento do número de fluxos de dados, explorando a falta de isolamento entre fluxos devido ao uso do TCP [Shieh et al. 2011], e pelo envio de grandes fluxos por meio do protocolo UDP. Considerando que a rede interna da nuvem é compartilhada por todos os locatários, o ataque pode afetar um grande número de aplicações.

O ataque precisa ser lançado por um "insider", ou seja, um locatário registrado na nuvem, e que pode portanto ser em tese responsabilizado. Entretanto, não há requisitos estritos para criar contas em nuvens e, mesmo que houvesse, o ataque poderia ser efetuado através de uma conta comprometida [Greenberg et al. 2008]. Além disso, a detecção de ataques não é simples. Tráfego malicioso pode simular um tráfego honesto, dificultando a distinção entre os dois tipos, e os esquemas de ofuscação, utilizados pelos provedores para ocultar a localização dos recursos na nuvem, não são capazes de parar um adversário persistente [Shieh et al. 2011].

4. Modelo de Sistema

Com base nas questões de segurança apresentadas anteriormente, é proposta uma estratégia na forma de um modelo de otimização para a alocação de aplicações de

 ©2012 SBC — Soc. Bras. de Computação

locatários que visa mitigar o impacto de ataques de egoísmo e negação de serviço na rede interna da nuvem. A execução desses ataques é facilitada atualmente na nuvem, visto que os provedores não cobram pelo tráfego de dados na rede interna [Guo et al. 2010].

Esta seção define as formulações básicas utilizadas para a modelagem da estratégia. As notações são representadas por meio das seguintes regras: i) sobrescritos s, v e r indicam entradas relacionadas à rede do substrato físico da nuvem, às infraestruturas virtuais e às requisições dos locatários, respectivamente; ii) subscritos são índices provenientes de atributos, variáveis ou elementos de um conjunto. O sistema de notação utilizado é similar ao empregado por Chowdhury et al. [Chowdhury et al. 2009].

4.1. Requisições de Aplicações

Cada requisição de aplicação $a \in A^r$, proveniente de um locatário, é definida pela tupla $(M_a^r,\ Band_a^r)$. O número de máquinas virtuais requisitadas é representado por M_a^r. Para facilidade de exposição do modelo desenvolvido, os detalhes das VMs são abstraídos e todas elas são consideradas iguais (consomem a mesma quantidade de recursos de CPU, memória e armazenamento). De modo similar a Ballani et al. [Ballani et al. 2011], uma requisição é estendida para especificar, além de recursos computacionais, recursos de rede. A largura de banda disponível para uma VM comunicar-se com outra VM da mesma aplicação é indicada por $Band_a^r$, que é um número real positivo ($Band_a^r \in \mathbb{R}^+$).

Ademais, cada aplicação $a \in A^r$ possui ou não confiança nas outras aplicações da nuvem, de acordo com as relações mútuas entre os locatários das mesmas. A confiança é representada por T_{a_i,a_j}^r, que indica se a aplicação a_i confia na aplicação a_j. Com o objetivo de simplificar a avaliação (mas sem prejuízo ao modelo), no presente trabalho assume-se que as relações de confiança são diretas, binárias e simétricas. Em outras palavras, um locatário confia ou não em um outro com quem ele interage e, se confia, então é recíproco; presume-se que um esquema padrão como PGP [Zimmermann 1995] pode ser usado para o estabelecimento de confiança mútua.

4.2. Infraestruturas Virtuais

O conjunto de infraestruturas virtuais é indicado por I^v. Cada elemento $i \in I^v$ é representado por um grafo bidirecional valorado $G_i^v = (S_i^v, M_i^v, E_i^v, Band^v, Oversub^v)$, sendo seus elementos definidos a seguir. O conjunto de dispositivos de rede (*switches*) de i é indicado por S_i^v. O conjunto de máquinas virtuais de i, tipicamente entre 20 e 40 por *rack* [Greenberg et al. 2009], é indicado por M_i^v e o conjunto de enlaces virtuais por E_i^v. Cada enlace $e^v = (w_i, w_j) \in E_i^v$ conecta os nodos w_i e w_j ($w_i, w_j \in S_i^v \cup M_i^v$). Além disso, cada enlace $e^v \in E_i^v$ possui uma largura de banda bidirecional $Band^v(e^v) \in \mathbb{R}^+$ e um fator de sobrecarga $Oversub^v(e^v) \in \mathbb{Z}^+$ empregado pelo provedor aos recursos de rede. Além disso, o subconjunto de *switches Top-of-Rack* (ToR), retornado pela função $ToR(S_i^v)$, é representado por R_i^v ($R_i^v \subset S_i^v$).

4.3. Infraestrutura Física

De forma similar a Guo et al. [Guo et al. 2010], o substrato físico da nuvem é modelado considerando servidores, dispositivos de rede e enlaces. Ele é representado por

 ©2012 SBC — Soc. Bras. de Computação

um grafo bidirecional valorado $G^s = (S^s\ , M^s\ , E^s\ , Band^s\ , Slots^s\ , Cost^s\ , Cap^s)$, no qual S^s é conjunto de dispositivos de rede (*switches*), M^s é o conjunto de servidores e E^s é o conjunto de enlaces. Cada servidor $m^s \in M^s$ possui um número de *slots*[2] ($Slots^s(m^s) \in \mathbb{Z}^+$). Cada *switch* $s^s \in S^s$ possui um número máximo de *switches* virtuais que ele pode hospedar ($Cap^s(s^s) \in \mathbb{Z}^+$) e possui um custo ($Cost^s(s^s) \in \mathbb{R}^+$) para hospedar um *switch* virtual. O custo é proporcional à importância do *switch* na rede, ou seja, *switches* localizados em níveis mais próximos do núcleo da rede possuem maior custo para hospedar *switches* virtuais. Cada enlace $e^s = (w_i, w_j) \in E^s \mid w_i, w_j \in S^s \cup M^s$ entre os nodos w_i e w_j é associado a uma capacidade (largura de banda) bidirecional $Band^s(e^s) \in \mathbb{R}^+$.

5. Estratégia de Alocação de Recursos

A estratégia apresentada nesse artigo para a alocação de recursos visa mitigar ataques de egoísmo e comportamento malicioso na rede interna da nuvem. Diferentemente de trabalhos anteriores, o objetivo é alcançado por meio do agrupamento de aplicações em infraestruturas virtuais considerando a confiança mútua entre os locatários e a minimização do tráfego de rede gerado pelas interações entre as VMs da mesma aplicação.

Entretanto, existe um desafio fundamental a ser enfrentado: o problema de alocação de recursos em nuvem considerando enlaces de rede com largura de banda limitada é NP-Difícil [Guo et al. 2010]. Por essa razão, dividimos o problema em dois sub-problemas ou etapas menores, propomos uma estratégia de alocação ótima para cada um, e combinamos os resultados. A primeira etapa distribui e mapeia as aplicações em infraestruturas virtuais, enquanto a segunda é responsável por alocar cada infraestrutura virtual no substrato físico da nuvem.

Neste trabalho, o algoritmo para a formação do conjunto de infraestruturas virtuais (I^v) é abstraído e foca-se na alocação de aplicações nas VIs segundo critérios de segurança e desempenho e no mapeamento das VIs no substrato físico. Dessa forma, o conjunto de VIs é utilizado como entrada para o modelo de otimização. Apesar disso, sugere-se um um critério para formação de VIs, que poderia ser usado pelo provedor. As VIs podem ser formadas com base em informações obtidas a partir de serviços de monitoramento de recursos da nuvem (por exemplo, o Amazon CloudWatch[3]). O uso de informações pertencentes ao histórico da utilização de recursos sobre um grande período de tempo tende a facilitar e a melhorar a qualidade do processo de alocação [Meng et al. 2010]. Além disso, a topologia de rede de cada VI não é restrita pela topologia da infraestrutura física, visto que a heterogeneidade de topologias pode ser tratada no mapeamento das VIs no substrato físico [Chowdhury et al. 2009].

De modo geral, o processo completo é representado na Figura 1, que indica cada etapa por meio de uma função. A primeira etapa é representada por $\mathcal{F} : A^r \rightarrow I^v$, enquanto a segunda é denotada por $\mathcal{G} : I^v \rightarrow G^s$. A função $\mathcal{H} : A^r \rightarrow G^s$ realiza o mapeamento direto de aplicações na infraestrutura física da nuvem, mostrando como esse processo é efetuado atualmente em nuvens. Dessa forma, pode-se dizer que $\mathcal{H}$ é uma composição de $\mathcal{F}$ e $\mathcal{G}$.

[2]Como assume-se que todas as VMs são iguais, ou seja, consomem a mesma quantidade de recursos dos servidores, os *slots* também são considerados iguais.

[3]http://aws.amazon.com/cloudwatch/

 ©2012 SBC — Soc. Bras. de Computação

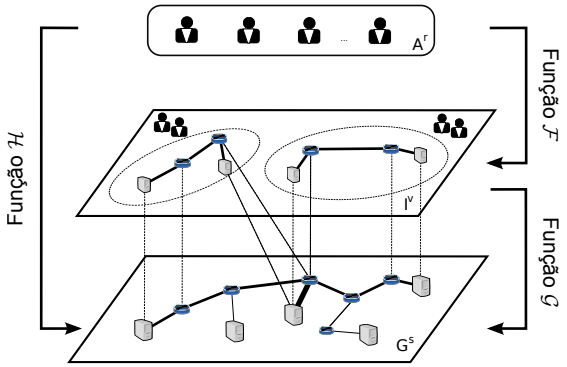


Figura 1. O processo de alocação de recursos na nuvem.

Apesar da estratégia ser composta por duas etapas, ambas podem ser executadas simultaneamente, visto que a saída de uma não interfere na entrada da outra. Isso implica que o tempo necessário para a conclusão do processo é o tempo de execução da etapa mais longa. Nesta seção, primeiramente é apresentada a função $\mathcal{F}$ e, após, a função $\mathcal{G}$.

5.1. Mapeamento de Aplicações em Infraestruturas Virtuais

A função $\mathcal{F}$ tem por objetivo mapear as aplicações em VIs considerando os níveis mútuos de confiança entre os locatários das mesmas. A função de mapeamento utiliza como dados de entrada as requisições de aplicações na nuvem (Seção 4.1) e o conjunto de infraestruturas virtuais (Seção 4.2).

As principais variáveis relacionadas ao modelo de otimização são:

- $F_{a,i,(w_1,w_2),p} \in \mathbb{R}^+$: indica o tamanho do fluxo total de dados para a aplicação $a \in A^r$ no enlace $(w_1, w_2) \in E_i^v \mid w_1, w_2 \in S_i^v$ para a comunicação entre o par de *racks* $p = (r_1, r_2) \mid r_1, r_2 \in R_i^v$ e $r_1 \neq r_2$ da VI $i \in I^v$. O valor desse fluxo, definido com base no número de VMs da aplicação a nos *racks* r_1 e r_2, será explicado posteriormente (Equação 2);
- $g_{a,i,r,m} \in \mathbb{B}$: indica se a VM $m \in M_i^v$ do *rack* $r \in R_i^v$ da VI $i \in I^v$ foi alocada para a aplicação $a \in A^r$;
- $G_{a,i}$: indica se a aplicação $a \in A^r$ está na VI $i \in I^v$;
- $h_{a,i,(w_1,w_2),p} \in \mathbb{B}$: indica se a aplicação $a \in A^r$ utiliza o enlace $(w_1, w_2) \in E_i^v \mid w_1, w_2 \in S_i^v$ para a comunicação entre o par de *racks* $p = (r_1, r_2) \mid r_1, r_2 \in R_i^v$ e $r_1 \neq r_2$ da VI $i \in I^v$;
- H_{i,a_1,a_2}: indica se as aplicações $a_1, a_2 \in A^r$ estão alocadas na VI $i \in I^v$.

A função objetivo do problema de otimização dessa etapa, apresentada na Equação 1, busca minimizar uma penalização imposta por alocar aplicações provenientes de locatários mutuamente não confiáveis na mesma VI. A penalização considera as características das aplicações (número de VMs e largura de banda) mutuamente não confiáveis. Por exemplo, a aplicação a_1 possui 20 VMs, cada uma com largura de banda de 200 Mbps. Caso a aplicação a_2 não confie em a_1, a função objetivo gera uma penalização de a_1 em a_2, cujo valor é $20 \times 200 = 4000$. Essa fórmula também endereça as diferenças entre as aplicações dos locatários. Ou seja, há maior risco de um locatário atacar outro, se não houver confiança entre ambos, caso ele requisite uma aplicação composta por 50 VMs e 100 Mbps de largura de banda para cada VM em detrimento de uma aplicação composta por uma única VM com 100 Mbps.

 ©2012 SBC — Soc. Bras. de Computação

$$Z = \text{Min} \sum_{i \in I^v} \sum_{a_1 \in A^r} \sum_{a_2 \in A^r} ((1 - T^r_{a_1,a_2}) * H_{i,a_1,a_2} * M^r_{a_1} * Band^r_{a_1}) \quad (1)$$

As restrições do problema especificam o mapeamento das aplicações nas VIs, de acordo com os recursos computacionais e de rede disponíveis. Na rede, o tráfego de comunicação entre VMs localizadas no mesmo *rack* não possui custo, visto que ele permanece interno ao *rack* e só utiliza os enlaces que ligam as VMs ao *switch* ToR. Entretanto, o tráfego entre VMs de *racks* distintos possui custo, que é definido pela largura de banda consumida e pelos enlaces utilizados.

Desse modo, as VMs de cada aplicação são alocadas em *grupos*, de modo semelhante a Ballani et al. [Ballani et al. 2011]. Um grupo é composto pelo conjunto de VMs da mesma aplicação localizadas no mesmo *rack*. Por isso, visa-se criar o menor número possível de grupos de VMs, ou seja, alocar as VMs da mesma aplicação próximas umas das outras, o que reduz o tráfego de dados na rede e tende a dificultar ataques de egoísmo e DoS.

Com essa abordagem, é necessário garantir que haja largura de banda disponível para a comunicação entre os grupos de VMs. Por exemplo, considerando a existência de dois grupos $G_{a,1}$ e $G_{a,2}$ de VMs da aplicação $a \in A^r$ e que a comunicação ocorra em um caminho virtual P^v, todos os enlaces desse caminho devem possuir a largura de banda necessária disponível. Uma única VM da aplicação a não pode enviar ou receber dados a uma taxa maior que a requisitada ($Band^r_a$). Dessa forma, o tráfego entre dois grupos é limitado à largura de banda exigida pelo menor dos grupos: $\min(|G_{a,1}|, |G_{a,2}|) * Band^r_a$. A Figura 2 exemplifica uma situação em que duas aplicações possuem dois grupos de VMs cada uma. Nesse caso, é necessário garantir a largura de banda na rede para a comunicação entre a dupla de grupos de VMs de cada aplicação. A restrição para a comunicação de um grupo com os outros grupos da aplicação a é formulada da seguinte maneira:

$$B_{a,g_i} = \min \left(|g_i| * Band^r_a, \sum_{g \in G^r_a, g \neq g_i} |g| * Band^r_a \right) \qquad \forall g_i \in G^r_a \quad (2)$$

onde G^r_a é o conjunto de grupos da aplicação a.

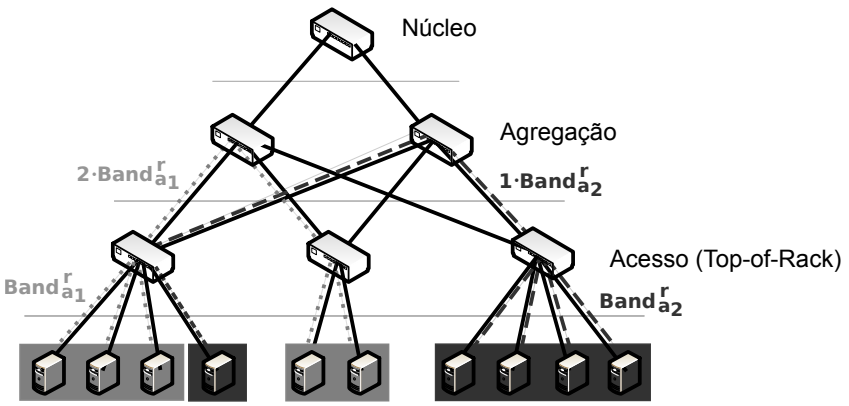


Figura 2. Comunicação em rede entre grupos de VMs.

 ©2012 SBC — Soc. Bras. de Computação

Em *datacenters*, o protocolo *Spanning Tree* assegura que o tráfego entre máquinas dentro da mesma camada dois da rede é encaminhado ao longo de uma árvore de expansão. Os dispositivos de rede da nuvem geralmente são interconectados com uma malha de enlaces, cuja utilização é balanceada através do uso de múltiplos caminhos de mesmo custo (*Equal-Cost Multipath* - ECMP) [Abts and Felderman 2012]. Dada a quantidade de multiplexação sobre a malha de enlaces e o número limitado de caminhos, os múltiplos caminhos podem ser considerados como um único caminho agregado para o tratamento de questões de largura de banda.Dessa forma, o modelo matemático proposto considera a utilização de um caminho para a comunicação entre os grupos de VMs da mesma aplicação. As fórmulas para o cálculo dos caminhos são mostradas nas Equações 3 e 4, que determinam o tráfego de saída do *rack* fonte e de entrada no *rack* destino, e nas Equações 5 e 6, utilizadas para garantir que um fluxo possui somente um caminho na rede. Outras restrições do modelo, que buscam garantir que a capacidade dos recursos virtuais não seja excedida, são omitidas devido a questões de espaço.

$$\sum_{\substack{w_2 \in S_i^v \mid \\ (r_1,w_2) \in E_i^v}} F_{a,i,(r_1,w_2),p} = \sum_{\substack{w_1 \in S_i^v \mid \\ (w_1,r_2) \in E_i^v}} F_{a,i,(w_1,r_2),p} = \min\left(\sum_{m \in M_i^v} g_{a,i,r_1,m}, \sum_{m \in M_i^v} g_{a,i,r_2,m} \right) * Band_a^r$$
$$\forall i \in I^v, \quad \forall a \in A^r, \quad \forall r_1, r_2 \in R_i^v \mid r_1 \neq r_2 \text{ and } p = (r_1, r_2) \quad (3)$$

$$\sum_{w_4 \in S_i^v \setminus w_3 \mid (w_3,w_4) \in E_i^v} F_{a,i,(w_3,w_4),p} - \sum_{w_4 \in S_i^v \setminus w_3 \mid (w_4,w_3) \in E_i^v} F_{a,i,(w_4,w_3),p} = 0$$
$$\forall i \in I^v, \quad \forall a \in A^r, \quad \forall p = (r_1, r_2) \mid r_1, r_2 \in R_i^v \text{ and } r_1 \neq r_2, \quad \forall w_3 \in S_i^v \setminus \{r_1, r_2\} \quad (4)$$

$$F_{a,i,(w_1,w_2),p} \leq Band^v(w_1, w_2) * h_{a,i,(w_1,w_2),p}$$
$$\forall a \in A^r, \quad \forall i \in I^v, \quad \forall w_1, w_2 \in S_i^v \mid (w_1, w_2) \in E_i^v, \quad \forall p = (r_1, r_2) \mid r_1, r_2 \in R_i^v \text{ and } r_1 \neq r_2 \quad (5)$$

$$\sum_{w_2 \in S_i^v \mid (w_1,w_2) \in E_i^v} h_{a,i,(w_1,w_2),p} \leq 1$$
$$\forall a \in A^r, \quad \forall i \in I^v, \quad \forall w_1 \in S_i^v, \quad \forall p = (r_1, r_2) \mid r_1, r_2 \in R_i^v \text{ and } r_1 \neq r_2 \quad (6)$$

5.2. Mapeamento de Infraestruturas Virtuais no Substrato Físico

A função $\mathcal{G}$ é responsável pelo mapeamento das VIs no substrato físico da nuvem. Ela aborda um problema similar ao problema de mapeamento de redes virtuais (*virtual network embedding* - VNE), o que possibilita lidar com a heterogeneidade das topologias das VIs em relação ao substrato físico [Chowdhury et al. 2009]. No entanto, além da topologia da rede (composta por dispositivos de rede e enlaces), ela considera também a utilização de nodos computacionais (servidores e máquinas virtuais).

O problema de otimização desenvolvido para essa etapa utiliza como entradas o conjunto de infraestruturas virtuais (Seção 4.2) e a infraestrutura física da nuvem (Seção 4.3). Além disso, são utilizados os parâmetros $\alpha_{(w_1,w_2)}$ e γ, como segue. O fator $\alpha_{(w_1,w_2)}$ é definido de acordo com a importância do enlace (w_1, w_2) e possui

 ©2012 SBC — Soc. Bras. de Computação

valor máximo 1 ($0 < \alpha_{(w_1,w_2)} \leq 1$). Já o parâmetro γ é utilizado para balancear os dois componentes da função objetivo.

As seguintes variáveis são utilizadas no modelo:

- $x_{i,s^v,s^s} \in \mathbb{B}$: indica se o *switch* virtual $s^v \in S_i^v$ da infraestrutura virtual $i \in I^v$ está alocado no *switch* físico $s^s \in S^s$;
- $y_{i,e^v,(w_1,w_2)} \in \mathbb{B}$: indica se o enlace virtual $e^v \in E_i^v$, pertencente à infraestrutura virtual $i \in I^v$, utiliza o enlace físico $(w_1, w_2) \in E^s$;
- $z_{i,m^v,m^s} \in \mathbb{B}$: indica se a VM $m^v \in M_i^v$ da infraestrutura virtual $i \in I^v$ está alocada no servidor $m^s \in M^s$.

A função objetivo (7) busca minimizar os recursos utilizados para alocar o conjunto de infraestruturas virtuais. O primeiro componente está relacionado aos enlaces da rede. Ao dividir-se $\alpha_{(w_1,w_2)}$ pela largura de banda do enlace, garante-se que os enlaces com menor prioridade e maior capacidade são preferidos em detrimento dos enlaces com maior importância. O segundo componente quantifica o custo de alocar *switches* virtuais em *switches* do substrato físico. Apesar da função não considerar requisitos de segurança, planeja-se incluir critérios de resiliência/segurança em trabalhos futuros, a fim de permitir maior isolamento entre as VIs no substrato físico e alocar enlaces virtuais em múltiplos caminhos, mitigando possíveis ataques de DoS.

$$Z = \text{Min} \sum_{(w_1,w_2)\in E^s} \frac{\alpha_{(w_1,w_2)}}{Band^s(w_1,w_2)} * \sum_{i\in I^v} \sum_{e^v \in E_i^v} y_{i,e^v,(w_1,w_2)} * Band^v(e^v) + \gamma * \sum_{s^s \in S^s} \sum_{i\in I^v} \sum_{s^v \in S_i^v} x_{i,s^v,s^s} * Cost^s(s^s) \tag{7}$$

Os valores das variáveis pertencentes à função objetivo são definidos com base nas restrições do modelo. Inicialmente, é necessário assegurar a capacidade dos servidores (Equação 8), dos *switches* físicos (Equação 9) e dos enlaces do substrato (Equação 10) em suportar os respectivos elementos virtuais.

$$\sum_{i\in I^v} \sum_{m^v \in M_i^v} z_{i,m^v,m^s} \leq Slots^s(m^s) \qquad \forall m^s \in M^s \tag{8}$$

$$\sum_{i\in I^v} \sum_{s^v \in S_i^v} x_{i,s^v,s^s} \leq Cap^s(s^s) \qquad \forall s^s \in S^s \tag{9}$$

$$\sum_{i\in I^v} \sum_{e^v \in E_i^v} (y_{i,e^v,(w_1,w_2)} * Band^v(e^v)) \leq Band^s(e^s) \qquad \forall e^s = (w_1,w_2) \in E^s \tag{10}$$

Também deve-se garantir que os enlaces virtuais são mapeados em caminhos válidos do substrato físico (Equação 11). As demais restrições do modelo, responsáveis pelo mapeamento dos recursos virtuais no substrato físico, são omitidas devido a questões de espaço.

 ©2012 SBC — Soc. Bras. de Computação

$$\sum_{w_4 \in S^s} y_{i,e^v,(w_3,w_4)} - \sum_{w_4 \in S^s} y_{i,e^v,(w_4,w_3)} = x_{i,w_1,w_3} - x_{i,w_2,w_3}$$
$$\forall i \in I^v, \quad \forall w_1, w_2 \in S_i^v \mid e^v = (w_1, w_2) \in E_i^v, \quad \forall w_3 \in S^s \quad (11)$$

6. Avaliação

Nesta seção, são avaliados dois aspectos da estratégia de alocação de recursos proposta. Primeiramente, a qualidade da solução é quantificada de duas formas: o benefício propiciado, com o aumento do nível de segurança às aplicações, e o custo disso, decorrente das restrições impostas na alocação pelas relações de confiança entre locatários. Em segundo lugar, busca-se verificar o tempo de processamento necessário à alocação ótima de recursos, o que afeta a viabilidade da solução.

6.1. Ambiente de Avaliação

Os experimentos foram implementados e executados no IBM ILOG CPLEX *Optimization Studio*[4]. Todos os experimentos foram executados em um computador Intel Core i3-2120 de 3.30 GHz, com 8 GB de memória RAM e sistema operacional GNU/Linux Debian x86_64. Cada experimento é limitado a um tempo máximo de duração de 1 hora e com uma árvore de busca de no máximo 4 GB, devido à complexidade do problema. Dessa forma, os resultados mostrados são os melhores resultados factíveis alcançados pelo CPLEX no período estipulado.

De modo similar a trabalhos relacionados [Ballani et al. 2011, Guo et al. 2010, Shieh et al. 2011], o substrato físico da nuvem foi definido como uma topologia em árvore. O substrato físico é composto por 80 servidores, cada um com 4 *slots*, divididos igualmente em 10 *racks*. Os *racks* são conectados entre si por *switches* das camadas superiores com diversidade de caminho.

A carga de trabalho a ser alocada são conjuntos de 12, 15, 20 ou 25 requisições para instanciar aplicações na nuvem, cada uma pertencendo a um locatário distinto. O número de VMs e a largura de banda especificados em cada requisição são uniformemente distribuídos, respectivamente, nos intervalos [2, 10] e [100,450] Mbps. A confiança mútua entre locatários foi gerada através das relações diretas entre os mesmos em um grafo aleatório, com grau de cada vértice (locatário) seguindo uma distribuição $P(k) \propto \frac{1}{k}$.

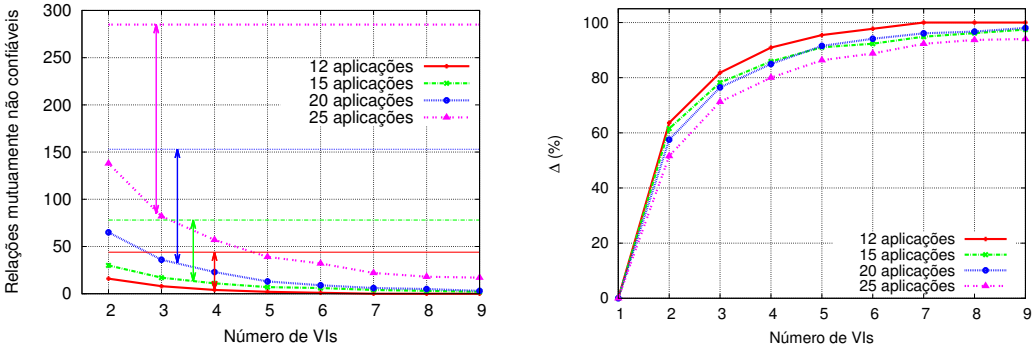
6.2. Qualidade da Solução

A qualidade da solução é quantificada de acordo com o critério de confiança mútua entre os locatários das aplicações da nuvem. Ou seja, busca-se avaliar a solução com base na segurança oferecida pelo agrupamento de aplicações de locatários em comparação com o cenário atual, no qual não é efetuado nenhum tipo de agrupamento. Refletindo isso, a métrica empregada é o número de relações entre aplicações de locatários que não possuem uma relação de confiança entre si e que foram alocadas no mesmo agrupamento. Deseja-se que esse valor seja o menor possível, pois representa uma certa exposição da aplicação a ataques.

[4] `http://www-01.ibm.com/software/integration/optimization/cplexoptimization-studio/`

 ©2012 SBC — Soc. Bras. de Computação

A Figura 3(a) mostra a variação do número de relações mutuamente não confiáveis de acordo com a quantidade de VIs oferecidas pelo provedor para os diferentes conjuntos de requisições de aplicações. Com base nesses dados, define-se diferença média do nível de confiança no cenário com agrupamento: $\Delta = 1 - \frac{\Phi}{\Gamma}$, onde Φ representa o número de relações não confiáveis entre aplicações dentro do agrupamento e Γ denota o número de relações não confiáveis sem o agrupamento. O valor de Δ é apresentado na Figura 3(b), que mostra que o número de aplicações não é o fator principal para o aumento da segurança, mas sim o número de VIs oferecidas pelo provedor. De modo geral, é possível verificar que o ganho de segurança tem um comportamento logarítmico de acordo com o número de VIs.



(a) Relações não confiáveis.

(b) Diferença do nível de segurança com o agrupamento.

Figura 3. Segurança com o agrupamento de aplicações na nuvem.

O agrupamento pode ser benéfico também em relação ao consumo de largura de banda e ao isolamento entre o tráfego de aplicações com requisitos de banda distintos, como segue. As VMs pertencentes a uma aplicação são posicionadas, por definição, dentro de uma VI, enquanto os recursos de uma VI tendem a ser alocados próximos uns dos outros na infraestrutura física. Portanto, quanto menor a VI, maior é a chance de VMs comunicantes estarem próximas e, consequentemente, consumirem menos recursos de rede. O isolamento entre o tráfego de aplicações com requisitos de banda distintos, por sua vez, minimiza a interferência nociva que pode ser causada ao misturar tráfego com diferentes características [Abts and Felderman 2012].

Por outro lado, a criação de VIs e o agrupamento de locatários nas mesmas pode ter dois efeitos negativos. Primeiro, a criação e a manutenção das VIs introduz uma sobrecarga de gerência e comunicação em relação a uma rede convencional. Em relação a isso, segundo trabalhos anteriores, essa sobrecarga é proporcionalmente pequena e plenamente justificável considerando os benefícios [Ballani et al. 2011].

Segundo, o agrupamento tende a restringir a alocação de aplicações na nuvem, por causa da fragmentação dos recursos. Ou seja, quanto menor as VIs criadas, maior a fragmentação (interna). Com isso, maior será o número de requisições que não poderão ser atendidas apesar da capacidade agregada disponível. No presente trabalho, não quantificamos esse efeito porque modelamos o problema de otimização considerando recursos suficientes para atender todas as requisições, similarmente à Guo et al. 2010. Pretendemos analisar mais a fundo essa questão, em cenários com carência de recursos, em trabalhos futuros.

 ©2012 SBC — Soc. Bras. de Computação

6.3. Tempo de Processamento para Alocação

Outra métrica analisada na avaliação da estratégia de alocação de recursos é o tempo de processamento necessário à alocação de um conjunto de aplicações a um conjunto de VIs, considerando o tratamento *offline* neste artigo. A complexidade da função $\mathcal{F}$ é combinatória de acordo com o número de aplicações, o número de VIs e os seus respectivos elementos virtuais. A complexidade da função $\mathcal{G}$ é combinatória de acordo com as VIs oferecidas e o substrato físico. Por isso, o espaço de busca por soluções de ambos os problemas de otimização é grande, sendo necessária uma grande quantidade de tempo de processamento e memória para encontrar a solução ótima. Apesar disso, soluções factíveis, mas não ótimas, podem ser encontradas com menor dificuldade.

Nesse contexto, a complexidade dos dois problemas serve como grande motivação para o desenvolvimento, em trabalhos futuros, de meta-heurísticas (p.ex., *Simmulated Annealing*) e heurísticas construtivas considerando a chegada dinâmica (*online*) de requisições de aplicações.

7. Considerações Finais

As economias de escala estão impulsionando aplicações distribuídas a coexistir em infraestruturas compartilhadas. Entretanto, a falta de mecanismos para controlar o compartilhamento de recursos da rede interna da nuvem possibilita a ocorrência de ataques de egoísmo e DoS, prejudicando tanto provedores quanto locatários. Desse modo, a alocação de aplicações considerando recursos de rede e critérios de segurança representa um grande desafio de pesquisa.

Este artigo apresentou uma estratégia de alocação de recursos que visa mitigar ataques de egoísmo e negação de serviço na rede interna da nuvem. Este é o primeiro trabalho na literatura a explorar o agrupamento de aplicações em infraestruturas virtuais considerando a confiança mútua entre os locatários. Dessa forma, o sistema torna-se mais resiliente contra aplicações que usariam uma fatia desproporcional de recursos, com ou sem o intuito de prejudicar outras aplicações. A estratégia foi avaliada analiticamente perante diferentes níveis de agrupamento e comparada com um esquema padrão, sem agrupamento.

Como trabalhos futuros, pretende-se desenvolver meta-heurísticas e heurísticas construtivas, que considerem alta taxa de chegada e saída de locatários (*churn*) e elasticidade horizontal, para o modelo proposto. Também pretende-se acrescentar questões referentes à segurança na função $\mathcal{G}$ e explorar o isolamento/agrupamento propiciado pela estratégia proposta para mitigar outros tipos de ataques (p.ex., privacidade).

Referências

Abts, D. and Felderman, B. (2012). A guided tour of data-center networking. *Commun. ACM*, 55(6):44–51.

Al-Fares, M., Loukissas, A., and Vahdat, A. (2008). A scalable, commodity data center network architecture. In *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, SIGCOMM '08, pages 63–74, New York, NY, USA. ACM.

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

 ©2012 SBC — Soc. Bras. de Computação

Ballani, H., Costa, P., Karagiannis, T., and Rowstron, A. (2011). Towards predictable datacenter networks. In *Proceedings of the ACM SIGCOMM 2011 conference on SIGCOMM*, SIGCOMM '11, pages 242–253, New York, NY, USA. ACM.

Breitgand, D. and Epstein, A. (2012). Improving Consolidation of Virtual Machines with Risk-aware Bandwidth Oversubscription in Compute Clouds. In *Proceedings of the 31th conference on Information communications*, INFOCOM'12, Piscataway, NJ, USA. IEEE Press.

Chowdhury, N., Rahman, M., and Boutaba, R. (2009). Virtual network embedding with coordinated node and link mapping. In *INFOCOM 2009, IEEE*, pages 783 –791.

Costa, R. B. and Carmo, L. F. R. C. (2007). Avaliação de Confiança Contextual em Grades Computacionais Multímodo usando Plataformas Seguras. In *VII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, SBSeg 2007*, pages 173–185.

Goel, A. and Indyk, P. (1999). Stochastic load balancing and related problems. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 579 –586.

Greenberg, A., Hamilton, J., Maltz, D. A., and Patel, P. (2008). The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73.

Greenberg, A., Hamilton, J. R., Jain, N., Kandula, S., Kim, C., Lahiri, P., Maltz, D. A., Patel, P., and Sengupta, S. (2009). Vl2: a scalable and flexible data center network. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*, SIGCOMM '09, pages 51–62, New York, NY, USA. ACM.

Grobauer, B., Walloschek, T., and Stocker, E. (2011). Understanding Cloud Computing Vulnerabilities. *Security Privacy, IEEE*, 9(2):50–57.

Guo, C., Lu, G., Wang, H. J., Yang, S., Kong, C., Sun, P., Wu, W., and Zhang, Y. (2010). Secondnet: a data center network virtualization architecture with bandwidth guarantees. In *Proceedings of the 6th International Conference*, Co-NEXT '10, pages 15:1–15:12, New York, NY, USA. ACM.

Jensen, M., Schwenk, J., Gruschka, N., and Iacono, L. L. (2009). On Technical Security Issues in Cloud Computing. In *Cloud Computing, 2009. CLOUD '09. IEEE International Conference on*, pages 109–116.

Kitsos, I., Papaioannou, A., Tsikoudis, N., and Magoutis, K. (2012). Adapting data-intensive workloads to generic allocation policies in cloud infrastructures. In *Network Operations and Management Symposium (NOMS), 2012 IEEE*, pages 25 –33.

Lam, T. and Varghese, G. (2010). NetShare: Virtualizing Bandwidth within the Cloud. Technical report cs2010-0957, University of California.

Liu, H. (2010). A new form of DOS attack in a cloud and its avoidance mechanism. In *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, CCSW '10, pages 65–76, New York, NY, USA. ACM.

Meng, X., Pappas, V., and Zhang, L. (2010). Improving the scalability of data center networks with traffic-aware virtual machine placement. In *Proceedings of the 29th conference on Information communications*, INFOCOM'10, pages 1154–1162, Piscataway, NJ, USA. IEEE Press.

Ristenpart, T., Tromer, E., Shacham, H., and Savage, S. (2009). Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM conference on Computer and communications security*, CCS '09, pages 199–212, New York, NY, USA. ACM.

Shieh, A., Kandula, S., Greenberg, A., Kim, C., and Saha, B. (2011). Sharing the data center network. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 23–23, Berkeley, CA, USA. USENIX Association.

Wang, M., Meng, X., and Zhang, L. (2011). Consolidating virtual machines with dynamic bandwidth demand in data centers. In *INFOCOM, 2011 Proceedings IEEE*, pages 71 –75.

Zimmermann, P. R. (1995). *The official PGP user's guide.* MIT Press, Cambridge, MA, USA.

 ©2012 SBC — Soc. Bras. de Computação