

Diagnóstico do Provisionamento de Recursos para Máquinas Virtuais em Nuvens IaaS

Ricardo J. Pfitscher¹, Maurício A. Pillon¹, Rafael R. Obelheiro¹

¹PPGCA - Programa de Pós-Graduação em Computação Aplicada –
Universidade do Estado de Santa Catarina (UDESC/Joinville)

ricardo.pfitscher@gmail.com, {mpillon, rro}@joinville.udesc.br

Abstract. *Infrastructure-as-a-service (IaaS) clouds enable customers to allocate computing resources in a flexible manner to satisfy their needs, and pay only for the allocated resources. One of the challenges for IaaS customers is the correct provisioning of their resources. Many users end up underprovisioning, hurting application performance, or overprovisioning, paying for resources that are not really necessary. Our work uses monitoring to enable a cloud customer to determine if the resources available to his virtual machines are correctly provisioned, or are under-/overprovisioned. Our focus is on processor and network resources, which can be easily provisioned in current virtualization environments. Experimental results with the Xen platform demonstrate the effectiveness of the proposed approach.*

Resumo. *Nuvens infrastructure-as-a-service (IaaS) permitem que clientes aloquem recursos computacionais de forma flexível para atender suas necessidades, e paguem apenas pelos recursos alocados. Um dos desafios para clientes IaaS é o provisionamento adequado de seus recursos. Muitos usuários incorrem no subprovisionamento, prejudicando o desempenho de suas aplicações, ou no superprovisionamento, pagando por recursos que não são realmente necessários. Este trabalho usa monitoração para permitir que um cliente de nuvem determine se os recursos disponíveis para suas máquinas virtuais estão provisionados corretamente, ou se estão sub- ou superprovisionados. O foco está nos recursos de processador e rede, que podem ser facilmente reservados nos ambientes de virtualização atuais. A eficácia da abordagem proposta é demonstrada por resultados experimentais na plataforma Xen.*

1. Introdução

Um dos desafios da computação em nuvem envolve o provisionamento adequado dos recursos computacionais necessários para atender às demandas dos clientes-usuários. As nuvens trazem como grandes atrativos a elasticidade de recursos e a tarifação por recursos alocados (*pay-per-use*), transferindo os custos fixos de infraestrutura computacional do cliente para um provedor [Armbrust et al. 2010]. Provedores de *infrastructure-as-a-service* (IaaS) oferecem como recursos máquinas virtuais (processador, memória, disco) e largura de banda de rede, que podem ser alocados conforme a necessidade do cliente [Suleiman et al. 2012].

Em uma solução IaaS, usuário e provedor possuem o mesmo objetivo geral, que é o de minimizar custos, mas diferem no modo de atingir esse objetivo. O usuário obtém

um custo mais baixo alocando o mínimo de recursos suficientes para obter um desempenho aceitável. De outro lado, o provedor está interessado em maximizar a ocupação dos seus recursos físicos desde que isso não comprometa a qualidade dos serviços oferecidos. Quando a demanda por recursos é variável, e não constante, o conflito entre essas estratégias se torna evidente: se um recurso físico estiver com toda a sua capacidade alocada para clientes, ele será incapaz de acomodar picos de carga sem prejudicar o desempenho das máquinas virtuais. Em contrapartida, a existência de capacidade ociosa (reservada mas não usada) vai de encontro ao interesse do provedor.

Geralmente, o equilíbrio entre os interesses de clientes e provedores é estabelecido por acordos de nível de serviço (SLAs – *Service Level Agreements*), que definem métricas (SLIs – *Service Level Indicators*) e respectivos valores-objetivo (SLOs – *Service Level Objectives*) para que o desempenho dos recursos alocados seja considerado satisfatório [Sauvé et al. 2005]. Muitos trabalhos de pesquisa que propõem estratégias para melhorar a alocação de recursos em ambientes IaaS, tais como [Buyya et al. 2011, Beloglazov e Buyya 2010, Kundu et al. 2012], tomam a existência de SLAs como uma premissa. Essa premissa embute um outro pressuposto, o de que o cliente é capaz de identificar corretamente as suas necessidades, e expressá-las em termos de SLIs e SLOs. Na prática, porém, fixar SLOs que garantam o desempenho desejado, a um custo aceitável, para as aplicações que serão executadas na nuvem não é trivial, haja vista a falta de diretrizes confiáveis para a especificação dos parâmetros. Logo, muitos usuários acabam sub- ou superdimensionando os seus objetivos, e conseqüentemente os seus recursos.

Quando recursos são subprovisionados, o usuário é capaz de perceber que o desempenho está aquém do esperado, mas muitas vezes não consegue dizer quais recursos precisam de mais capacidade. Por outro lado, o efeito do superprovisionamento de recursos é percebido menos pelo desempenho e mais pelo custo financeiro. A ausência de parâmetros de comparação da relação custo \times aplicação (que permitam constatar que aplicações similares conseguem obter desempenho satisfatório a um custo mais baixo) e a dificuldade em conhecer a capacidade suficiente para cada recurso (quais SLOs podem ser reduzidos?) tornam ainda mais complexa a questão.

Nesse sentido, o objetivo deste trabalho é fornecer ao cliente um diagnóstico sobre o provisionamento dos recursos para máquinas virtuais em nuvens IaaS. A ideia é monitorar métricas de desempenho específicas de cada recurso e identificar, em tempo de execução, se a alocação de recursos para a máquina virtual está adequada, subprovisionada ou superprovisionada. Esse diagnóstico serve a variados propósitos. Por exemplo, o cliente pode usá-lo para ajustar a capacidade de suas máquinas virtuais à demanda, seja redefinindo a alocação de recursos para as MVs existentes ou aumentando/diminuindo o número de MVs instanciadas. Em serviços regidos por SLA, o diagnóstico pode ser usado para refinar os SLOs definidos em contrato. O diagnóstico também pode ser usado como entrada para um mecanismo automatizado de gerência da elasticidade característica das nuvens, permitindo que recursos sejam alocados ou liberados conforme a necessidade.

Os recursos considerados neste trabalho são processador e rede, que podem facilmente ser reservados nos monitores de máquinas virtuais atuais e também podem ser objeto de SLA em alguns provedores IaaS [Suleiman et al. 2012]. As métricas de desempenho são todas coletadas dentro das máquinas virtuais, sendo portanto diretamente acessíveis ao usuário, sem a necessidade de intermediação do provedor. Além disso, elas

refletem o desempenho efetivamente verificado nas máquinas virtuais, considerando a influência tanto da capacidade do *hardware* subjacente como do compartilhamento desse *hardware* com outras MVs. As avaliações experimentais são conduzidas com Linux na plataforma Xen,¹ um dos ambientes de virtualização mais usados por provedores IaaS, sendo a principal solução de código aberto desse mercado [Butler 2012].

A principal contribuição do trabalho é a proposta de uma abordagem que permita a clientes IaaS obter um diagnóstico da adequação dos recursos alocados a suas máquinas virtuais em relação à demanda existente. Uma segunda contribuição consiste na introdução do uso do tamanho da fila do adaptador de rede como métrica auxiliar para determinar se a largura de banda disponível a um sistema é ou não suficiente.

O restante do artigo está organizado da seguinte forma. Na seção 2 são descritas as abordagens para diagnosticar o provisionamento de processador e rede, que são validadas pelos resultados experimentais mostrados na seção 3. A seção 4 discute trabalhos relacionados, e a seção 5 traz as considerações finais.

2. Diagnóstico do Provisionamento de Recursos

Para diagnosticar o provisionamento de processador e rede, é necessário definir métricas que indiquem se esse provisionamento está adequado, sub ou superprovisionado. Como o foco está no cliente, são propostas métricas que podem ser obtidas dentro de uma MV.

2.1. Processador

Em ambientes virtualizados, o monitor de máquinas virtuais (*hypervisor* no Xen) disponibiliza unidades de processamento virtuais (VCPUs) para cada domínio virtual (ou MV). Mesmo quando várias VCPUs são multiplexadas sobre o mesmo processador físico, cada VCPU é vista pela MV como um processador real. O *hypervisor* então escala as VCPUs nas CPUs físicas de acordo com o algoritmo de escalonamento do MMV [Wood et al. 2008]. Sendo assim, a monitoração de processador realizada pelo sistema operacional (SO) da MV deve obter valores relacionados à ocupação de sua VCPU, da mesma forma que uma máquina não virtualizada monitora a sua CPU física.

Para diagnosticar a adequação do provisionamento de processador a máquinas virtuais em nuvem, serão avaliadas duas métricas: a utilização de processador (U_{cpu}) e o percentual de “roubo” ($\%Steal$). A utilização U_{cpu} representa quanto tempo o processador estava executando tarefas em um espaço de tempo. No Linux, a contabilização do processador subdivide a utilização em vários componentes; por exemplo, $\%User$ representa o percentual de tempo em que o recurso foi utilizado por processos de usuário, e $\%System$ o tempo usado pelo sistema operacional [Hoch 2010]. Para simplificar, é possível usar o percentual de ociosidade ($\%Idle$) para obter de forma indireta a utilização do processador. Sendo assim, a utilização de processador é dada por $U_{cpu} = 100\% - \%Idle$, sendo sua média em um período denotada por $\overline{U_{cpu}}$.

O tempo de roubo (*steal time*) é uma métrica conhecida em ambientes de *main-frame*, e que representa o percentual de tempo em que uma máquina virtual deseja executar (ou seja, possui processos em estado de pronto) mas sua VCPU não está alocada em uma CPU física, devido à contenção de processador entre MVs [van Riel 2006]. Essa

¹<http://www.xen.org/>

métrica pode auxiliar a identificar a ausência de recursos na máquina física, de forma complementar à utilização de processador: quando $\%Steal$ é maior do que zero, a máquina virtual teve necessidades de execução em processador que não foram supridas, ou seja, existem máquinas concorrentes disputando o recurso.

A taxa de ocupação de processador tende a variar de acordo com a execução das aplicações no período de tempo. Logo, deve-se definir limiares para caracterizar os três níveis relacionados ao provisionamento do recurso. Entretanto, não há um consenso relacionado aos valores admitidos nos limiares; por exemplo, [Hoch 2010] aponta que valores entre 0–5% para $\%Idle$ indicam saturação, enquanto [Padala et al. 2007] adota 80% para utilização (20% de $\%Idle$) como limite de saturação, com a justificativa de manter uma margem de segurança para acomodar o comportamento variável das aplicações. Considerando que a ocupação de CPU pode alternar entre picos de utilização, define-se o recurso como subprovisionado quando $\overline{U_{cpu}}$ mantiver-se menor ou igual a 80%.

Contudo, apesar de serem propostos valores para saturação do recurso, não há regras práticas para apontar a sua subutilização. Considerando que o valor apropriado para utilização de processador para uma MV é o equilíbrio entre o máximo e mínimo disponível, espera-se que o diagnóstico de provisionamento superdimensionado ou adequado considere a proximidade ao equilíbrio (50%), permitindo que a capacidade provisionada suporte oscilações positivas e negativas de carga a um custo aceitável. Sendo assim, cenários onde a média de utilização de processador estiver entre 50% e 80% serão diagnosticados como adequados, pois não há carga suficiente para saturar o recurso e a redução da capacidade pode afetar o desempenho. Nos cenários com utilização inferior a 50%, a média de utilização pode não expressar corretamente um provisionamento superdimensionado, pois a carga pode ter $\overline{U_{cpu}}$ baixa mas apresentar picos de utilização que impeçam uma redução no recurso alocado (caso de C3 na Fig. 1(a)).

Para lidar com esse caso, avalia-se a variabilidade da utilização nas situações em que $\overline{U_{cpu}}$ é menor que 50%, verificando qual a fração de pontos que atingem o pico de utilização do recurso no intervalo monitorado. Seja U_{cpu}^i um ponto de medição de U_{cpu} , θ o limiar de saturação de CPU (i.e., $U_{cpu}^i \geq \theta$ indica CPU saturada), N o número de pontos no intervalo e N' o número de pontos com CPU saturada ($U_{cpu}^i \geq \theta$). Quando $N'/N > \omega$, onde ω é o limiar para o percentual de pontos em saturação, a MV utiliza toda a capacidade de CPU com frequência, e o provisionamento é adequado; caso contrário, ele é superdimensionado. O diagnóstico de provisionamento de processador é expresso pelas equações (1)–(7). Com base em observações experimentais, os limiares adotados foram $\theta = 95\%$ e $\omega = 20\%$, mas esses valores podem ser ajustados.

$$u_i = \begin{cases} 1, & \text{se } U_{cpu}^i \geq \theta \\ 0, & \text{se } U_{cpu}^i < \theta \end{cases} \quad (1) \quad \left| \quad N' = \sum_{i=0}^N u_i \quad (2) \quad \right| \quad \varphi = \frac{N'}{N} \quad (3)$$

$$\overline{U_{cpu}} \geq 80\% \Rightarrow \text{Subprovisionado} \quad (4)$$

$$50\% \leq \overline{U_{cpu}} < 80\% \Rightarrow \text{Adequado} \quad (5)$$

$$\overline{U_{cpu}} < 50\% \wedge \varphi > \omega \Rightarrow \text{Adequado} \quad (6)$$

$$\overline{U_{cpu}} < 50\% \wedge \varphi \leq \omega \Rightarrow \text{Superprovisionado} \quad (7)$$

Para exemplificar, seja um cenário hipotético com 10 medições de U_{cpu}^i associadas a quatro cargas de trabalho distintas (C1–C4), mostrado na Fig. 1(a). Observa-se que, para C1, a média de U_{cpu} no período é de 20%, logo o processador está superprovisionado. Para C2, tem-se $\overline{U_{cpu}}$ próxima a 100%, ou seja, processador subprovisionado. A carga de trabalho C3 apresenta média de U_{cpu} igual a 38%, com quatro pontos onde a utilização é próximo ao topo ($\varphi = 40\%$); assim, o provisionamento é adequado, pois o recurso alocado suporta a variação entre picos de utilização e ociosidade. A carga C4 apresenta média de U_{cpu} igual a 75%, e assim o provisionamento é considerado adequado.

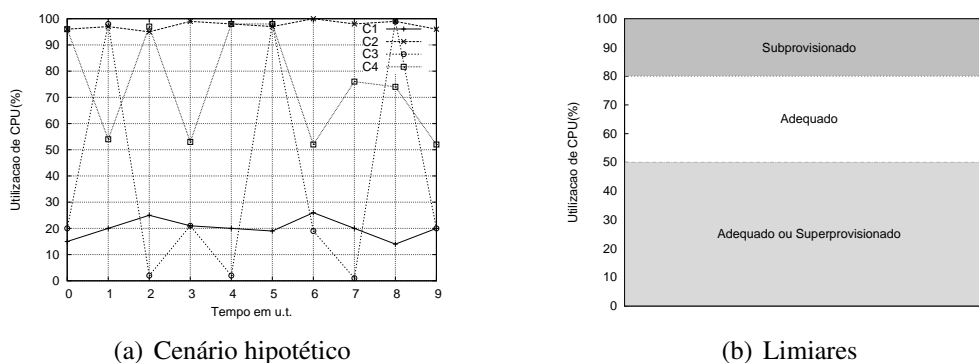


Figura 1. Utilização de CPU em um cenário hipotético 1(a) e Limiares para diagnóstico de provisionamento 1(b). C1 está superprovisionado, C2 está subprovisionado, C3 e C4 têm provisionamento adequado.

A Fig. 1(b) resume os possíveis diagnósticos do provisionamento de processador com base em medições de U_{cpu} . No caso de utilização média situada na região superior ($\overline{U_{cpu}} \geq 80\%$), tem-se subprovisionamento. Para utilização na região intermediária ($50\% \leq \overline{U_{cpu}} < 80\%$), o provisionamento é adequado. Caso a utilização média corresponda à região inferior ($\overline{U_{cpu}} < 50\%$), o recurso pode ser adequado (quando $\varphi > \omega$) ou superprovisionado (quando $\varphi \leq \omega$).

2.2. Rede

Em um ambiente de virtualização, as máquinas virtuais compartilham a largura de banda das interfaces de rede do *hardware* subjacente. Tipicamente, o MMV permite que seja reservada uma fração da largura de banda disponível para cada MV; caso não seja feita uma alocação fixa, ocorre a multiplexação estatística do recurso, com cada MV usando a largura de banda disponível durante um tempo, conforme sua necessidade. Internamente, cada MV possui uma interface de rede virtual, que o seu SO enxerga como uma interface de rede física. O MMV é responsável por transmitir pela interface física os quadros que as MVs enviam por suas interfaces virtuais, e vice-versa [Apparao et al. 2006].

A métrica básica para verificar a adequação da capacidade de rede alocada a uma máquina virtual é a largura de banda. Considera-se apenas a largura de banda de transmissão, pois é a única que pode ser analisada com mais profundidade apenas com dados internos à MV; para a largura de banda de recepção, a análise possível é se a banda consumida se aproxima da banda alocada, não sendo possível inferir se há demanda além da alocação. Do mesmo modo que a utilização de processador, a largura de banda consumida pode variar bastante ao longo do tempo, o que faz com que a média das observações

seja insuficiente para caracterizar esse consumo. [Pras et al. 2009] sugere que se utilize a variância entre os pontos medidos. No nosso trabalho, a variabilidade do consumo de banda é mensurada pelo coeficiente de variação (CV), que é a razão entre o desvio padrão e a média dos pontos medidos: se o CV for de até 5%, considera-se comportamento de consumo constante. Em relação à variância, essa métrica tem a vantagem de ser relativa, e não absoluta. O limiar de 5% foi estabelecido com base em observações experimentais.

Todavia, apenas a análise da largura de banda não permite obter um diagnóstico preciso da adequação do provisionamento da rede. Em particular, quando a utilização de largura de banda se aproxima do máximo disponível, é necessário avaliar se há uma real necessidade de banda adicional (ou seja, se a capacidade disponível está sendo um fator limitante para a MV) ou se a MV está apenas explorando ao máximo a banda disponível, sem precisar de banda extra. Protocolos de rede que implementam controle de congestionamento, como o TCP, adaptam suas taxas de transmissão às condições da rede, tentando encontrar a largura de banda disponível na rede, que é limitada pela menor largura de banda ao longo do caminho de transmissão [Jacobson 1988]. Para fins de diagnóstico da adequação dos recursos, deve-se identificar quando essa limitação está na largura de banda local (seja física ou virtual), ou se ela está ocorrendo em algum ponto no interior da rede. No primeiro caso (que é o único em que se pode tomar alguma providência local), o MMV não consegue transmitir os quadros das máquinas virtuais com a rapidez necessária, e ocorre formação de fila na interface de rede da MV. No segundo caso, não há formação de fila na interface local. Portanto, o comprimento da fila da interface de rede da MV pode ser usado para verificar a adequação da alocação de recursos quando a largura de banda consumida se aproxima do limite.

Com base nas duas métricas propostas, coeficiente de variação do consumo de banda e fila na interface de rede, definem-se três categorias de provisionamento (subprovisionado, adequado e superprovisionado). Quando existe fila na interface de rede da MV e o consumo de largura de banda é constante, diagnostica-se o recurso como subprovisionado. Quando a banda consumida é constante e não há enfileiramento, o recurso está provisionado de forma adequada. O superprovisionamento ocorre quando o consumo de banda é variável, sem enfileiramento na interface de rede da MV. Essas categorias são descritas pelas Eqs. (8)–(10), onde CV_{lb} é o coeficiente de variação da largura de banda consumida e \overline{fila} é a quantidade média de pacotes na fila da interface.

$$CV_{lb} \leq 5\% \wedge \overline{fila} > 0 \Rightarrow \text{Subprovisionado} \quad (8)$$

$$CV_{lb} \leq 5\% \wedge \overline{fila} = 0 \Rightarrow \text{Adequado} \quad (9)$$

$$CV_{lb} > 5\% \wedge \overline{fila} = 0 \Rightarrow \text{Superprovisionado} \quad (10)$$

3. Análise Experimental

A fim de verificar as propostas definidas na seção 2 para diagnóstico do provisionamento de recursos em nuvem, fez-se uma avaliação experimental. Os recursos foram monitorados a partir da execução de diferentes cargas de trabalho que representam cenários variados de utilização dos recursos.

3.1. Ambiente de Testes

O ambiente de testes é composto por dois computadores com *hardware* idênticos: processador AMD Phenom II X4 B93 2,8 GHz (*quad-core*), 4 GB de memória RAM e disco

500 GB SATA (7200 rpm). A conexão entre as máquinas é realizada por uma rede local de 100 Mbps. As máquinas virtuais criadas dispunham de uma VCPU, 256 MB de memória e 4 GB de espaço em disco montado sobre uma partição LVM.

Em uma das máquinas, denominada cliente, foi instalado o sistema operacional Ubuntu Linux 9.10 com *kernel* 2.6.31-23. Na outra, denominada servidor, foi instalado o sistema operacional Debian Squeeze 64 bits, com *kernel* 2.6.32-5, e sobre este instalou-se o *hypervisor* Xen na versão 4.0.1. As máquinas virtuais utilizam sistema operacional paravirtualizado Debian Squeeze 64 bits, com *kernel* 2.6.32-5.

Para a avaliação de processador, a ferramenta *stress*² foi usada para geração de carga. A monitoração foi efetuada nas próprias MVs com a ferramenta *sar*, do pacote *sysstat*³. Para os experimentos envolvendo a rede, as máquinas virtuais no servidor possuíam um servidor *web* Apache 2.2.16, e o cliente realizava requisições com a ferramenta de *benchmark* *httperf*⁴, versão 0.9.0-1. Para monitorar o consumo de largura de banda, foram extraídas estatísticas de `/proc/net/dev`, que discriminam a banda consumida por interface. Para monitorar a fila da interface de rede, utilizou-se a ferramenta de configuração e gerenciamento de interface de rede *tc*⁵, que fornece, entre outras estatísticas, a quantidade de pacotes na fila da interface.

As métricas foram observadas a cada 1 s, o que oferece uma boa resolução para acompanhar a variabilidade de comportamento das MVs; além disso, as ferramentas *sar* e *stress* têm granularidade mínima de 1 s. Como o *overhead* de monitoração das métricas escolhidas é baixo (*sar* consumiu 0,06% de CPU nos testes), esse mesmo intervalo pode ser usado em ambientes de produção. O diagnóstico foi realizado com base em um período de 60 s, o que é suficiente para nossos propósitos. Em produção, esse período de diagnóstico deve ser compatível com o período de tarifação do provedor IaaS, que tipicamente é de 1 h [Suleiman et al. 2012], possibilitando que se analise o comportamento com base em um número grande de amostras, reduzindo a influência de observações discrepantes. Foram cinco repetições para cada cenário, com $CV \leq 5\%$ em todos os casos.

3.2. Diagnóstico do Provisionamento de Processador

Para avaliar o provisionamento de processador de acordo com o proposto na seção 2.1, foram definidos três cenários de carga sobre o recurso: CPU-Bound, Misto e IO-Bound. No cenário CPU-Bound, a ferramenta *stress* executa carga intensiva de processador sobre cada MV. No cenário IO-Bound, os processos criados por *stress* executam essencialmente operações em disco. No cenário Misto, os processos alternam entre execuções no processador e requisições a disco. O tempo total de execução da ferramenta *stress* foi dividido em intervalos de 10 segundos, sendo que os tempos para execução em processador e disco definidos são 2, 5 e 8 s. Portanto, são avaliados três cenários Mistos: carga de processador maior que a de disco (8 s processador/2 s disco), cargas equivalentes (5 s processador/5 s disco), e carga de disco maior que a de processador (2 s processador/8 s disco).

Em todos os cenários (CPU-Bound, Misto e IO-Bound) variou-se o número de máquinas virtuais com execução da carga e monitorou-se as métricas *%Steal* e *%Idle* (de

²<http://weather.ou.edu/~apw/projects/stress/>

³<http://sebastien.godard.pagesperso-orange.fr/>

⁴<http://www.hpl.hp.com/research/linux/httperf/>

⁵<http://lartc.org/manpages/tc.txt>

onde deriva-se U_{cpu}). Os gráficos da Fig. (2) mostram a média dos pontos observados nos 60 s de monitoração.

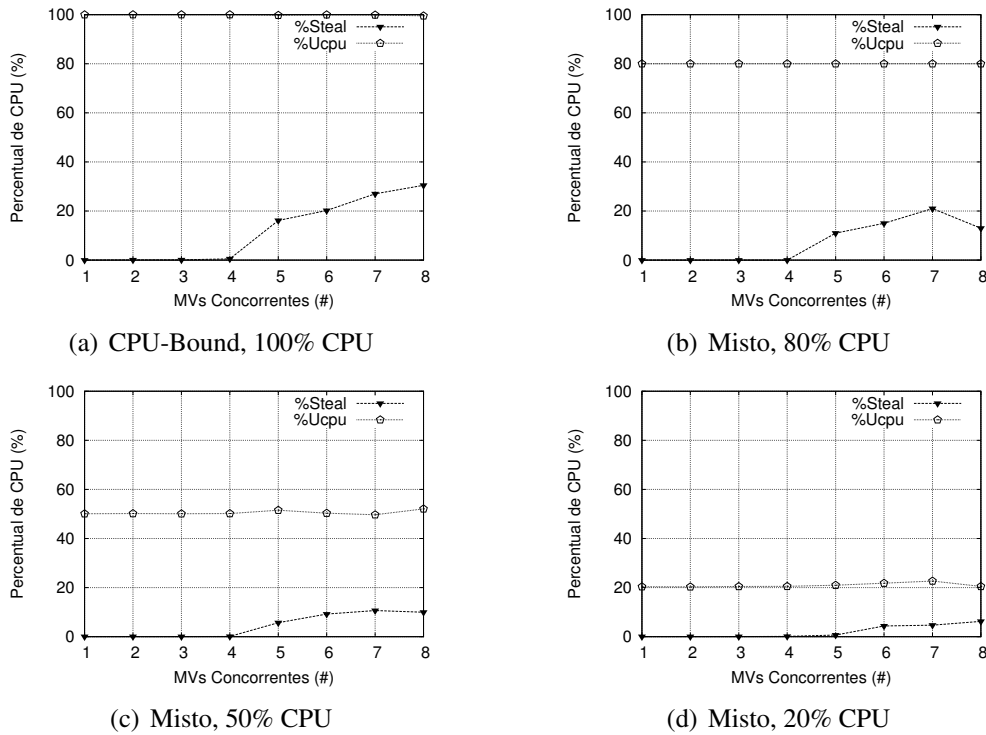


Figura 2. Média de U_{cpu} e %Steal (período de 60 s, 1–8 MVs concorrentes)

Ao observar o gráfico da Fig. 2(a), é possível perceber que, na execução de cargas CPU-Bound, o processador físico se torna saturado a partir de cinco máquinas virtuais, pois possui quatro *cores*. Isso pode ser verificado pelo surgimento de %Steal na monitoração do consumo de processador das MVs. Como $\overline{U_{cpu}}$ é igual a 100%, o processador está subprovisionado. Os dados relacionados à carga IO-Bound (cujo gráfico foi omitido por questão de espaço) permitem concluir que para todos os pontos o percentual de ociosidade manteve-se em 100%, o que indica superprovisionamento.

Na execução de carga mista com maioria do tempo em processador (Fig. 2(b)), a média de U_{cpu} se mantém em 80% para qualquer número de máquinas virtuais, o que indica subprovisionamento. Nos casos de 50% e 20% de utilização de CPU (Figs. 2(c) e 2(d)), o diagnóstico sobre o provisionamento não pode ser obtido pela simples avaliação da média geral, pois a utilização de CPU pode não ser constante. Assim, gerou-se, para estes dois casos, os gráficos das Figs. 3(a) e 3(b), onde todos os pontos medidos para uma MV são apresentados.

O cenário de carga mista equilibrada (Fig. 3(a)), com $\overline{U_{cpu}} = 49,95\%$, apresenta $N' = 26$ (Eq. (2)) e $\varphi = 43\%$ (Eq. (3)); adotando $\omega = 20\%$ na Eq. (6), o provisionamento é adequado. No cenário com maior carga de disco (Fig. 3(b)), $\overline{U_{cpu}} = 20\%$, $N' = 8$ e $\varphi = 13\%$, e o diagnóstico é de superprovisionamento.

A avaliação sobre %Steal nos cenários da Fig. 2 permite concluir que, a partir de cinco máquinas virtuais em execução simultânea, passa a existir contenção de processa-

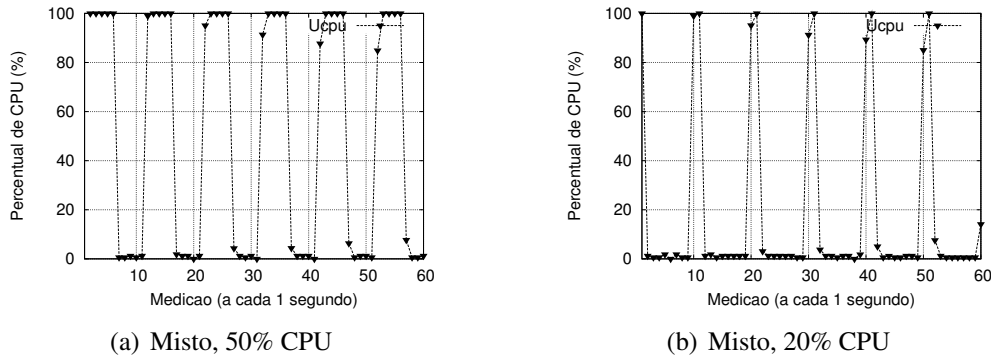


Figura 3. Utilização de CPU no período de 60 s com 1 MV em execução

dor, e ainda que o crescimento de $\%Steal$ está associado ao número de MVs concorrentes e à carga executada pelas MVs. Portanto, nos cenários diagnosticados como subprovisionados em que $\%Steal > 0$, o usuário tem ciência de que, para resolver o problema, não adianta alocar mais VCPU no mesmo *hardware*, sendo necessário migrar a MV para um servidor menos carregado ou instanciar novas máquinas virtuais.

3.3. Diagnóstico do Provisionamento de Rede

Para validar o diagnóstico do provisionamento de rede, foi usada uma máquina virtual com capacidade de largura de banda limitada no MMV através do arquivo de configuração da MV. A banda alocada variou entre 10 Mbps e 100 Mbps, em intervalos de 10 Mbps. Definiram-se três cargas de trabalho, de acordo com o tamanho do arquivo requisitado ao servidor *web* (32 KB, 4 MB, 50 MB). Em todos os casos, havia cinco clientes simultâneos, com intervalo entre requisições entre 1 e 5 s, de acordo com uma distribuição uniforme.

Para representar um cenário que saturasse a rede, independente da largura de banda destinada à MV, os clientes requisitaram ao servidor arquivos de 50 MB. De modo complementar, o arquivo definido para subutilização também deveria manter este comportamento (de subutilização) para qualquer largura de banda provisionada, e assim utilizou-se arquivos de 32 KB. Por fim, em um cenário intermediário, foram usados arquivos de 4 MB, que deveriam saturar a rede quando a banda provisionada fosse pequena, mas não quando ela fosse aumentada além de certo ponto no intervalo entre 10 e 100 Mbps.

As requisições de 32 KB e 50 MB exibem comportamentos semelhantes, independente da largura de banda provisionada; assim, a Fig. 4 mostra os resultados somente para os extremos de cada cenário (10 Mbps e 100 Mbps). Os resultados estão de acordo com o esperado (superprovisionado para 32 KB, subprovisionado para 50 MB), conforme expresso pelas Eqs. (8) e (10): no primeiro caso (Figs. 4(a) e 4(b)) não há enfileiramento na interface de rede e o consumo de largura de banda não alcança o limite estabelecido, enquanto no segundo caso (Figs. 4(c) e 4(d)), verifica-se a ocorrência de fila e o consumo de banda mantém-se constante ($CV_{lb} < 5\%$) sobre a largura de banda provisionada.

Para o cenário com arquivos de 4 MB, foram obtidos três diagnósticos diferentes, dependendo da largura de banda alocada: subprovisionamento entre 10 e 50 Mbps, provisionamento adequado com 60 Mbps e superprovisionamento entre 70 e 100 Mbps. Os gráficos da Fig. 5 mostram o intervalo em que há a mudança no diagnóstico. Em termos da análise proposta na seção 2.2, verifica-se que, com largura de banda igual a

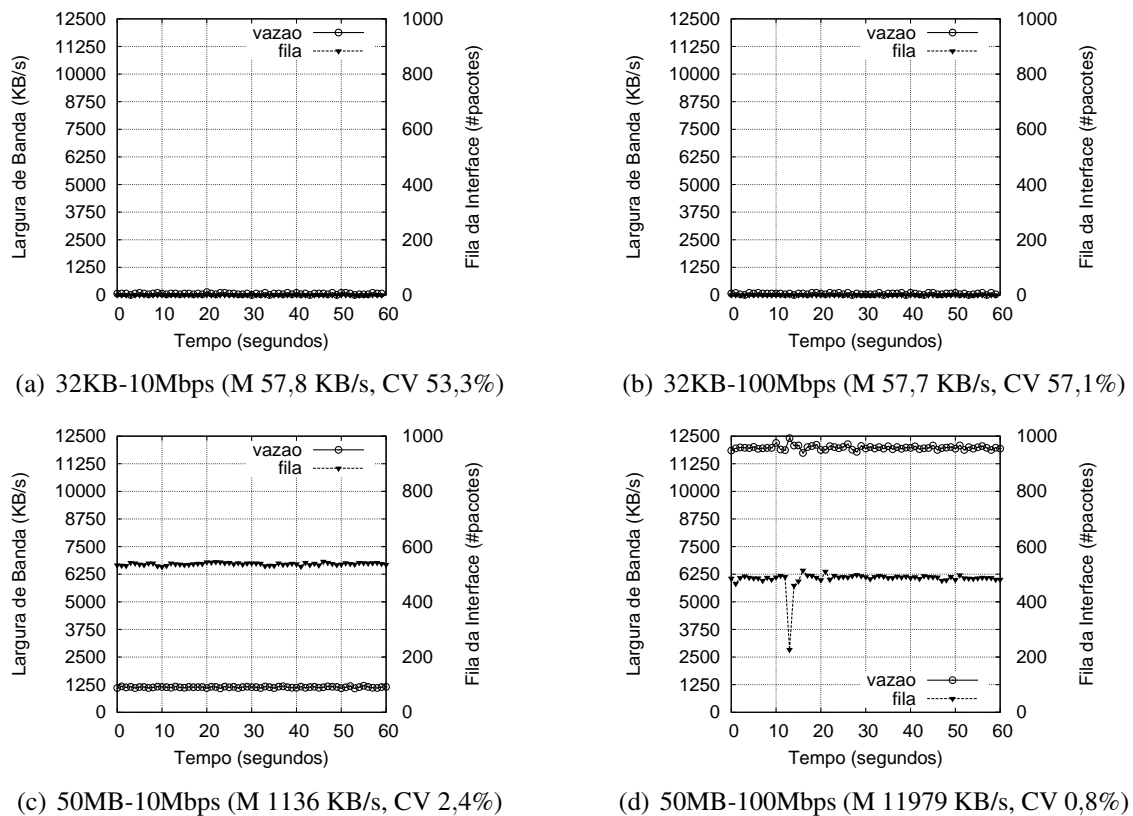


Figura 4. Consumo de largura de banda e enfileiramento nos ambientes saturado (50 MB) e não saturado (32 KB). M é a média do consumo de banda, e CV o coeficiente de variação. Para requisições a arquivos de 32 KB a banda provisionada nunca é alcançada, e para arquivos de 50 MB a banda provisionada é ocupada durante todo o intervalo de medição.

50 Mbps o ambiente está subprovisionado (Fig. 5(a)), ou seja, é necessário mais largura de banda, uma vez que o consumo de banda é constante ($CV_{lb} < 5\%$) e ocorre enfileiramento (Eq. (8)). Quando o limite estabelecido é de 60 Mbps (Fig. 5(b)), a Eq. (9) indica que o provisionamento realizado é adequado, pois o comportamento é constante e não há enfileiramento. A partir de 70 Mbps (Fig. 5(c)), a Eq. (10) indica que o recurso foi superprovisionado e a largura de banda pode ser reduzida, uma vez que o comportamento não é constante ($CV_{lb} > 5\%$). Portanto, pode-se concluir que o raciocínio subjacente às equações da seção 2.2 é consistente com os comportamentos verificados experimentalmente.

Para reforçar essa conclusão, observou-se o tempo de resposta da aplicação (no caso o *httpperf*) em cada cenário (tamanho de arquivo e largura de banda alocada). A Fig. 6 apresenta a média dos tempos de resposta entre os cinco clientes. Para requisições de 32 KB (Fig. 6(a)), o tempo de resposta praticamente não é influenciado pela variação da largura de banda alocada, com ganhos na ordem de milissegundos. No cenário da Fig. 6(b), onde os arquivos requisitados são de 4 MB, verifica-se que o tempo de resposta cai rapidamente à medida em que a largura de banda disponível aumenta de 10 Mbps para 60 Mbps, e que os ganhos a partir de 60 Mbps são sensivelmente menores. Por fim, as requisições de 50 MB obtêm uma redução perceptível no tempo de resposta com o aumento da banda; neste gráfico, quando a largura de banda era inferior a 40 Mbps os

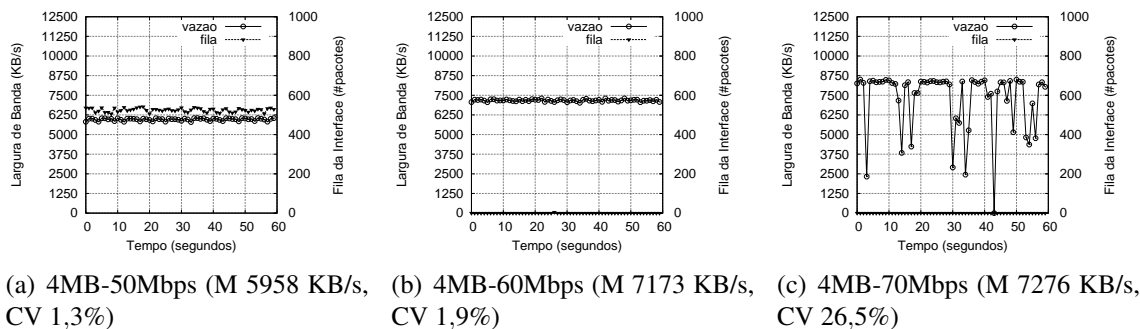


Figura 5. Representação de cenário com busca por largura de banda ideal, com requisições a arquivos de 4 MB. M é a média do consumo de banda, e CV o coeficiente de variação. O consumo é constante com banda de 50 Mbps e 60 Mbps, e inconstante com 70 Mbps. A partir de 60 Mbps, a fila na interface desaparece.

tempos de resposta foram maiores do que o intervalo de monitoração, e por isso os dados correspondentes não estão mostrados. Os resultados da Fig. 6 corroboram o diagnóstico do provisionamento de rede dado pelas Eqs.(8)–(10). Em particular, o gráfico da Fig. 6(b) demonstra que, dependendo do valor tarifado por Mbps, o ganho de desempenho pode não justificar um aumento de banda além de 60 Mbps.

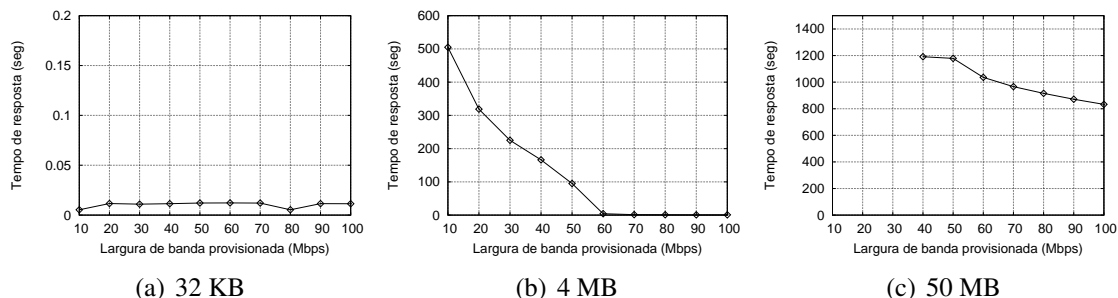


Figura 6. Média do tempo de resposta das requisições aos três tamanhos de arquivo (32 KB, 4 MB e 50 MB) de acordo com a largura de banda provisionada.

4. Trabalhos Relacionados

Os trabalhos relacionados discutidos nesta seção dividem-se em duas categorias: identificação da demanda de recursos de máquinas virtuais e provisionamento dinâmico de recursos em ambientes de nuvem.

Identificação da demanda. Métricas para identificar demandas de recursos de máquinas virtuais executando Linux são propostas em [van Riel 2006]. Para o processador é usada uma análise similar à apresentada na seção 2.1, envolvendo *%Idle* e *%Steal*; no entanto, não são definidos limiares para esses valores, apenas noções subjetivas como “*%Idle* alto” e “*%Steal* baixo”. A identificação da demanda de memória se baseia em estatísticas (*re-faults* e referências a páginas alocadas) que não estão disponíveis a aplicações de usuário nas versões atuais do Linux, e nem em outros SOs. O artigo menciona o uso de largura de banda/vazão como métrica para avaliar as demandas de disco e de rede, mas sem uma proposta concreta. Além disso, ele não apresenta resultados experimentais.

[Pras et al. 2009] propõe avaliar a variabilidade do consumo de largura de banda, associada à utilização média, para dimensionar enlaces de rede. A fila nas interfaces é usada como indicativo da variabilidade do consumo de banda. Em comparação com a proposta da seção 2.2, nota-se que tanto a variabilidade da largura de banda como a fila das interfaces são consideradas. No entanto, a medida da variabilidade é diferente, e em nosso trabalho o enfileiramento é usado para identificar o subprovisionamento da rede.

Alguns trabalhos [Du et al. 2011, Nikolaev e Back 2011] propõem o uso de contadores de desempenho de *hardware* – registradores que contabilizam a ocorrência de eventos de baixo nível, como *cache hits/misses* e instruções executadas – para a monitoração de ambientes virtuais. Embora a monitoração em nível de *hardware* geralmente tenha um *overhead* menor do que em nível de sistema operacional (como a proposta neste artigo), contadores de desempenho são mais apropriados para *profiling* de aplicações individuais, enquanto o nosso foco está na análise de MVs como um todo. Ademais, as métricas usadas em nosso trabalho são bem mais portáteis entre monitores de máquinas virtuais e arquiteturas de *hardware* do que os contadores de desempenho: mesmo existindo no processador, estes nem sempre estão acessíveis às máquinas virtuais, além de serem específicos de cada processador (o conjunto de contadores disponíveis varia mesmo entre diferentes processadores da mesma arquitetura, como Intel x86).

Provisionamento dinâmico de recursos. Uma ferramenta para realizar provisionamento de recursos orientado ao cumprimento de SLA é apresentada em [Buyya et al. 2011]. As requisições de recursos do usuário são alocadas sob a avaliação do cumprimento de SLA e métricas de qualidade de serviço (QoS). Assume-se que o cliente conhece as necessidades de suas aplicações, e que a ferramenta conhece as aplicações em execução. As principais diferenças em relação à proposta deste artigo são que, no nosso caso, as métricas monitoradas não estão associadas a uma aplicação específica, e o cliente não precisa conhecer seus SLOs *a priori*, pois o provisionamento é diagnosticado em tempo de execução.

O uso de técnicas de aprendizado de máquina para provisionamento de recursos em nuvens é proposto por [Rao et al. 2011, Kundu et al. 2012, Vasić et al. 2012]. Esses trabalhos associam a utilização de recursos com medidas de desempenho das aplicações (como tempo de resposta) e usam classificadores estatísticos para determinar a classe à qual pertence uma aplicação e, conseqüentemente, os recursos que devem ser alocados para sua execução. Além do foco em aplicações e não em máquinas virtuais, os resultados demonstram que essa abordagem funciona para cargas de trabalho estáticas, mas tem problemas com cargas variáveis e/ou desconhecidas.

Em [Heo et al. 2009] é proposta uma ferramenta para gerenciar a alocação de processador e memória a MVs sobre a plataforma Xen, possibilitando o *overbooking*. A utilização de ambos recursos é observada periodicamente; a partir dessas observações e de um valor objetivo, calcula-se a quantidade de recursos necessária. No entanto, como é considerada apenas a média de utilização e não a sua variação, o tempo de resposta da aplicação pode ser degradado. Além disso, em processador é necessário observar o tempo de resposta da aplicação, o que difere da proposta deste artigo.

[Sudevalayam e Kulkarni 2011] usam monitoração para avaliar o impacto no consumo de processador quando duas máquinas virtuais que se comunicam entre si são alocadas no mesmo *hardware*, e propõe um modelo analítico para estimar a utilização de

CPU quando as MVs são alocadas na mesma máquina física ou em máquinas separadas. A análise se restringe à plataforma Xen, e o único recurso considerado é o processador.

Um algoritmo para provisionar máquinas virtuais levando em conta a correlação na demanda por recursos entre as MVs é proposto em [Halder et al. 2012]. MVs correlacionadas, que são aquelas com picos de demanda nos mesmos intervalos de tempo, são alocadas em máquinas físicas distintas, enquanto MVs não correlacionadas podem ser alocadas na mesma máquina. Além de considerar apenas processador, o algoritmo pressupõe cargas bem conhecidas, que se repetem sazonalmente, uma premissa que não é necessária em nossa abordagem.

5. Conclusão

Para que clientes de provedores IaaS obtenham um bom custo-benefício de suas soluções em nuvem, é necessário equilibrar desempenho e custo, e para isso é fundamental conhecer suas necessidades de recursos computacionais. Este trabalho apresenta uma importante contribuição nesse sentido, fornecendo a clientes IaaS um diagnóstico sobre o provisionamento de processador e rede de suas máquinas virtuais. A abordagem proposta se baseia na análise de métricas disponíveis nas máquinas virtuais para determinar se esses recursos estão adequados, sub- ou superprovisionados diante da demanda existente. A avaliação de processador leva em conta a média e a variação da utilização do recurso, enquanto a avaliação de rede se baseia na variação do consumo de largura de banda e no enfileiramento observado nas interfaces virtuais. Resultados experimentais com Linux e Xen demonstram que a abordagem proposta é capaz de diagnosticar corretamente o provisionamento em diferentes cenários de carga.

O diagnóstico fornecido permite que um cliente IaaS adeque sua alocação de recursos e/ou seus acordos e objetivos de nível de serviço (SLAs/SLOs) estabelecidos com o provedor. Uma extensão natural deste trabalho seria um mecanismo automatizado de gerência da elasticidade de recursos na nuvem que incorpore esse diagnóstico em suas decisões. Outra perspectiva futura é investigar como oferecer um diagnóstico para o provedor IaaS sobre o provisionamento dos recursos aos seus diferentes clientes e MVs; para isso, será necessário definir métricas adequadas e que sejam visíveis ao provedor, estabelecendo a relação entre essas métricas, o provisionamento dos recursos e o desempenho resultante.

Referências

- Apparao, P., Makineni, S., e Newell, D. (2006). Characterization of network processing overheads in Xen. In *Proc. VTDC'06*, p. 2.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., e Zaharia, M. (2010). A view of cloud computing. *Commun. ACM*, 53(4):50–58.
- Beloglazov, A. e Buyya, R. (2010). Energy efficient resource management in virtualized cloud data centers. In *Proc. IEEE/ACM CCGRID'10*, pp. 826–831.
- Butler, B. (2012). Gartner's IaaS magic quadrant: A who's who of cloud market. *Network World*. <http://bit.ly/U1My9s>. Acessado em 29/nov/2012.

- Buyya, R., Garg, S., e Calheiros, R. (2011). SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. In *Proc. Intl. Conf. on Cloud and Service Computing (CSC)*, pp. 1–10.
- Du, J., Sehrawat, N., e Zwaenepoel, W. (2011). Performance profiling of virtual machines. *SIGPLAN Not.*, 46(7):3–14.
- Halder, K., Bellur, U., e Kulkarni, P. (2012). Risk aware provisioning and resource aggregation based consolidation of virtual machines. In *Proc. IEEE CLOUD'12*, pp. 598–605.
- Heo, J. Zhu, X. Padala, P. e Wang, Z. (2009). Memory overbooking and dynamic control of Xen virtual machines in consolidated environments. In *Proc. IFIP/IEEE IM'09*, pp. 630–637.
- Hoch, D. (2010). *Linux System and Performance Monitoring*. In *LinuxCon 2010*, <http://ufsdump.org/papers/linuxcon2010-linux-monitoring.pdf>.
- Jacobson, V. (1988). Congestion avoidance and control. In *Proc. SIGCOMM'88*, pp. 314–329.
- Kundu, S., Rangaswami, R., Gulati, A., Zhao, M., e Dutta, K. (2012). Modeling virtualized applications using machine learning techniques. *SIGPLAN Not.*, 47(7):3–14.
- Nikolaev, R. e Back, G. (2011). Perfctr-Xen: a framework for performance counter virtualization. In *Proc. ACM VEE'11*, pp. 15–26.
- Padala, P., Shin, K. G., Zhu, X., Uysal, M., Wang, Z., Singhal, S., Merchant, A., e Salem, K. (2007). Adaptive control of virtualized resources in utility computing environments. In *Proc. ACM EuroSys'07*, pp. 289–302.
- Pras, A., Nieuwenhuis, L., van de Meent, R., e Mandjes, M. (2009). Dimensioning network links: a new look at equivalent bandwidth. *IEEE Network*, 23(2):5–10.
- Rao, J., Bu, X., Xu, C.-Z., e Wang, K. (2011). A distributed self-learning approach for elastic provisioning of virtualized cloud resources. In *Proc. IEEE MASCOTS'11*, pp. 45–54.
- Sauvé, J., Marques, F., Moura, A., Sampaio, M., Jornada, J., e Radziuk, E. (2005). SLA design from a business perspective. *Ambient Networks*, LNCS 3775, pp. 72–83.
- Sudevalayam, S. e Kulkarni, P. (2011). Affinity-aware modeling of CPU usage for provisioning virtualized applications. In *Proc. IEEE CLOUD'11*, pp. 139–146.
- Suleiman, B., Sakr, S., Jeffery, R., e Liu, A. (2012). On understanding the economics and elasticity challenges of deploying business applications on public cloud infrastructure. *Journal of Internet Services and Applications*, 3:173–193.
- van Riel, R. (2006). Measuring resource demand in Linux. In *Proc. Ottawa Linux Symposium*.
- Vasić, N., Novaković, D., Miučin, S., Kostić, D., e Bianchini, R. (2012). Dejavu: accelerating resource allocation in virtualized environments. *SIGARCH Comp. Arch. News*, 40(1):423–436.
- Wood, T., Cherkasova, L., Ozonat, K., e Shenoy, P. (2008). Profiling and modeling resource usage of virtualized applications. In *Proc. ACM/IFIP/USENIX Middleware'08*, pp. 366–387.