EMA-Bench: Um Benchmark para a Avaliação de Mecanismos de Alocação Elástica de Memória

Guilherme Galante[1], Cristiane Andrade[1], Luiz Antonio Rodrigues[1], Rodrigo da Rosa Righi[2]

[1]Ciência da Computação – Unioeste
Cascavel – PR – Brasil

[2]Programa Pós-Graduação em Computação Aplicada – Unisinos
São Leopoldo – RS – Brasil

guilherme.galante@unioeste.br

Abstract. *Several works have addressed the development of mechanisms for elastic allocation of memory, creating a need for a precise way to evaluate such solutions. Analyzing the technical literature, we found a gap in the state of the art in the evaluation of vertical elasticity, since the proposals of metrics and methodologies focus on the evaluation of horizontal elasticity. In this sense, this work contributes with the development of the EMA-Bench, a benchmark for evaluating elastic memory allocation mechanisms. The benchmark enables to analyse a mecanism in terms of accuracy and time proportions spent in over-provisioning and overprovisioning states, as well as the financial costs involved. The results show that the proposed tool is able to assist the user to define the best mechanism to be adopted, ensuring cost reduction and the maintenance of quality of service. In addition, the EMA-Bench can be used by other researchers in the comparison and refinement of experiments and elastic solutions.*

Resumo. *Diversos trabalhos têm sido realizados com o objetivo de apresentar mecanismos para alocação de elástica de memória, se fazendo necessário métricas e metodologias capazes de avaliar tais soluções. Analisando a literatura técnica sobre o assunto, observa-se que as propostas de métricas e metodologias se concentram na avaliação de mecanismos de elasticidade horizontal, havendo uma lacuna no estado-da-arte para a avaliação da elasticidade vertical. Nesse sentido, este trabalho tem como principal contribuição o desenvolvimento do EMA-Bench, um benchmark para a avaliação de mecanismos de alocação elástica de memória. O benchmark fornece um conjunto de métricas para a avaliação da precisão e dos períodos dispendidos em períodos de sub-provisionamento e superprovisionamento, bem como o cálculo dos custos envolvidos. Os resultados mostram que a ferramenta proposta é capaz de auxiliar o usuário na definição de qual é o melhor mecanismo a ser adotado, garantindo redução de custo e a manutenção da qualidade do serviço. Além disso, o EMA-Bench pode ser utilizado por outros pesquisadores na comparação e refinamento de experimentos e de soluções elásticas.*

1. Introdução

A elasticidade é uma das principais características da computação em nuvem. Esta característica consiste na capacidade de adicionar ou remover recursos, sem interrupções e em tempo de execução, de acordo com uma demanda específica [Li 2017]. A capacidade de elasticamente expandir e contrair a base de recursos é interessante tanto para o provedor quanto para o usuário final da nuvem. Do ponto de vista do provedor, a elasticidade garante um melhor uso dos recursos de computação, fornecendo economia de escala e permitindo que mais usuários possam ser atendidos simultaneamente, uma vez que os recursos liberados por um usuário podem instantaneamente ser alocados por outro. Da perspectiva do usuário, a elasticidade pode ser usada para diversos fins, tais como manutenção da qualidade de serviço, complementação de recursos e redução de custos.

Dependendo da forma como a nuvem implementa o provisionamento de recursos, pode-se classificar a sua elasticidade em horizontal ou vertical [Galante and Bona 2012]. Uma nuvem com elasticidade horizontal possibilita apenas a adição ou a remoção dinâmica de instâncias alocadas por um usuário. Por sua vez, a elasticidade vertical é caracterizada pela possibilidade de se alterar a capacidade de VMs em execução. Tipicamente, a elasticidade vertical é implementada através da adição ou remoção de CPUs e memória, mas também pode ser utilizada no contexto de redes e armazenamento.

A elasticidade vertical é considerada uma das tecnologias chave para a concretização do conceito de Recurso Como Serviço (RaaS) [Ben-Yehuda et al. 2012] e uma das principais características da segunda geração de Infraestrutura como Serviço (IaaS 2.0), na qual os usuários pagam apenas pelos recursos que eles realmente usam, e os provedores de nuvem podem usar seus recursos de forma mais eficiente e servir mais usuários [Farokhi et al. 2016]. Embora atualmente não esteja implementada na maioria dos provedores de nuvem, a elasticidade vertical já é suportada em hipervisores como Xen e KVM e implementado por diversos mecanismos desenvolvidos pela academia [Galante and Bona 2012, Galante et al. 2016].

Nesse contexto, se fazem necessárias métricas e metodologias capazes de avaliar os mecanismos de elasticidade disponíveis com o intuito de auxiliar o usuário a definir qual é o mais apropriado para suas demandas. Em uma revisão da literatura técnica sobre o assunto, percebe-se que as propostas de métricas e metodologias se concentram na avaliação de mecanismos de elasticidade horizontal, havendo uma lacuna no estado-da-arte para a avaliação da elasticidade vertical. Nesse sentido, este trabalho tem como principal contribuição o desenvolvimento do *Elastic Memory Allocation Benchmark* (EMA-Bench), um *benchmark* para a avaliação de mecanismos de alocação elástica de memória. O *benchmark* implementa um conjunto de métricas para a avaliação da precisão e dos períodos dispendidos em períodos de subprovisionamento e superprovisionamento, bem como o cálculo dos custo envolvidos. Dessa forma, a ferramenta auxilia o usuário a definir qual é o mecanismo mais adequado a sua demanda, garantindo redução de custos e a manutenção da qualidade do serviço.

O restante do trabalho está organizado como segue. A Seção 2 apresenta os trabalhos relacionados. A Seção 3 apresenta uma breve revisão sobre a elasticidade vertical de memória. Na Seção 4, descreve-se o *benchmark* proposto. Na Seção 5, apresenta-se a metodologia de avaliação, os experimentos, bem como a discussão dos resultados. Por fim, a Seção 6 conclui este trabalho.

2. Trabalhos Relacionados

Diversas estratégias para a avaliação da elasticidade foram propostas na literatura. De modo geral, os trabalhos propõem metodologias e métricas para mensurar a elasticidade de nuvens de infraestrutura (IaaS).

Islam et al. (2009) define uma métrica para elasticidade baseadas em dois elementos: tempo e custo. A métrica reflete a penalidade financeira para um determinado usuário nos casos de subprovisionamento, que pode resultar em demandas não atendidas ou Acordo de Nível de Serviço (*Service Level Agreement* - SLA) não cumprido, ou de superprovisionamento, onde paga-se mais que o necessário pelos recursos para suportar uma carga de trabalho. No trabalho de Coutinho et al. (2014), os autores apresentam métricas para a medição da elasticidade horizontal baseada em conceitos da física, como estresse e tensão. Faz análise apenas sobre o uso de CPU e número de instâncias alocadas.

Hwang et al. (2016) propõem uma métrica de desempenho da nuvem em termos de eficiência e produtividade. Os experimentos são realizados na nuvem Amazon EC2, na qual os recursos são dimensionados pela quantidade e tipos de instâncias de máquina virtual. Por sua vez, Beltrán (2016) define uma métrica de elasticidade para ambientes de computação em nuvem que considera quatro componentes: escalabilidade, precisão, tempo e custo, além de descrever um procedimento para analisar a elasticidade em contextos de nuvem.

Por fim, Herbst et al. (2015, 2016) propõem um conjunto de métricas para caracterizar a elasticidade de uma plataforma de nuvem, e uma metodologia de *benchmarking*, denominada BUNGEE, para avaliar a elasticidade horizontal de plataformas de nuvem IaaS. As métricas pretendem avaliar os mecanismos de elasticidade em termos de acurácia e tempo dispendido em estados de sub e superprovisionamento.

Considerando os trabalhos apresentados, observa-se que todos se concentram na avaliação da elasticidade horizontal em nuvens IaaS, havendo uma lacuna no estado-da-arte para a avaliação da elasticidade vertical. Nesse sentido, propõem-se um *benchmark* para a avaliação da elasticidade vertical de memória. O *benchmark* se baseia no trabalho de Herbst et al. (2016), porém adaptado para elasticidade vertical de memória.

3. Elasticidade Vertical de Memória

Estimar a quantidade exata de memória necessária por uma determinada aplicação ou serviço não é uma tarefa trivial. Usuários tendem a superestimar seus requisitos de memória baseando-se nos cenários de pior caso, que na prática, pode ser necessário apenas em curtos períodos de tempo. Isso afeta diretamente o aproveitamento da infraestrutura da nuvem, pois resulta em memória não utilizada e que poderia ser dedicada a instâncias adicionais rodando em uma mesma máquina física. Como resultado, técnicas de *overbooking*, nas quais mais recursos são alocados do que fisicamente existem, tem se tornado prática comum [Farokhi et al. 2016]. Embora a abordagem permita aproveitar melhor a capacidade dos recursos, ela também pode causar impacto no desempenho de aplicações e violações de SLA.

Dessa forma, a exploração da elasticidade vertical pode apresentar diversas vantagens. É possível fornecer continuamente a quantidade necessária de memória considerando que aplicações distintas têm necessidades de memória diferentes e que suas car-

gas de trabalho podem modificar-se significativamente ao logo de sua execução. Assim, remove-se o ônus colocado sobre os usuários para que estimem com precisão os recursos a serem utilizados, como também possibilita que os provedores aumentem a eficiência do uso de sua infraestrutura e reduzam o consumo de energia, despesas de capital e custos de administração.

Os hipervisores modernos oferecem suporte à elasticidade de memória usando técnicas como compartilhamento de páginas, *hotplug* virtual e *ballooning* [Zhang et al. 2017] e embora haja esse suporte, a decisão de quando e como as alocações de memória devem ser feitas é tomada por mecanismos de elasticidade com controle mais elaborado. Nesse contexto, diversos trabalhos têm sido realizados com o objetivo de apresentar soluções para alocação de elástica de memória. Basicamente, as propostas se dividem em dois grupos: reativas e proativas [Galante and Bona 2012].

Soluções reativas empregam mecanismos do tipo Regra-Condição-Ação, onde cada condição considera um evento ou uma métrica do sistema que é comparado com um limiar determinado e quando uma condição definida em alguma dessas regras é satisfeita, uma ação é disparada. Por sua vez, as abordagens proativas utilizam técnicas analítico-matemáticas e heurísticas, para obter uma previsão sobre o comportamento das cargas do sistema e, a partir desses resultados, tomar decisões de como e quando escalar os recursos. Essas técnicas incluem análises de séries temporais [Spinner et al. 2015], Transformada de Fourier [Gong et al. 2010], cadeias de Markov e redes Bayesianas [Tan et al. 2012]. A abordagem proativa é mais apropriada para os casos onde a carga de trabalho apresenta padrões bem definidos com periodicidade uniforme, facilitando a previsão de cargas futuras.

4. EMA-Bench

Para possibilitar a avaliação de soluções de elasticidade vertical de memória deve-se haver uma abordagem bem definida, com o objetivo de fornecer suporte às decisões do usuário quanto a eficácia dos método e se satisfazem as suas demandas. É com este objetivo que o EMA-Bench foi proposto.

O EMA-Bench foi implementado em C++ e é executado em linha de comando. Conforme ilustrado na Figura 1, possui como entrada (1) um perfil de consumo de memória, (2) a indicação de qual método implementado no *benchmark* será executado e (3) os seus respectivos parâmetros de configuração. Após o processamento, fornece três resultados: (4) alocações dinâmicas de memória realizadas pelo mecanismo selecionado, fornecido como um arquivo texto, (5) resultado das métricas e (6) informações sobre o custo. Os detalhes das entradas e saídas da ferramenta são descritos na Seção 4.3.

Atualmente, o *benchmark* tem implementado três mecanismos de elasticidade (apresentados em detalhes na Seção 4.1), porém não se limita a esses mecanismos, sendo que novas soluções podem ser adicionadas diretamente no código-fonte, de acordo com um *template* de código pré-definido. As métricas implementadas baseiam-se nos trabalho de Herbst et al. (2015,2016) e fornecem informações sobre a precisão do mecanismo para determinado perfil e sobre os intervalos dispendidos em períodos de subprovisionamento e superprovisionamento. Mais detalhes sobre a metodologia são apresentados na Seção 4.2.

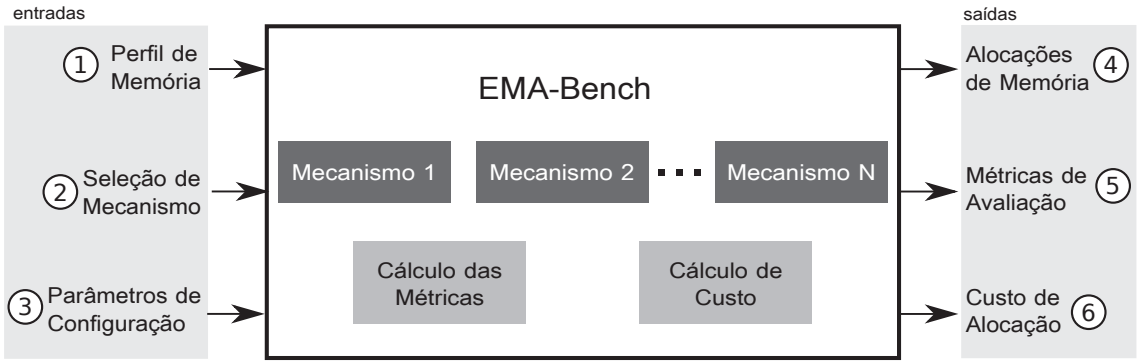

Figura 1. EMA-Bench: estrutura, entradas e resultados fornecidos.

4.1. Mecanismos Elasticidade Vertical de Memória

Três soluções reativas de alocação elástica disponíveis na literatura estão implementadas no EMA-Bench. A primeira solução é a proposta por Heo et al. (2009), na qual a próxima alocação de memória $u_{mem}(k+1)$ é calculada baseando-se na última alocação de memória $u_{mem}(k)$ e na memória consumida $v_{mem}(k)$. Nesse mecanismo há ainda dois parâmetros configuráveis, λ que é a constante de incremento que determina a agressividade do atuador e u_{ref}^{mem} que estabelece a quantidade desejável de memória ocupada na VM.

O cálculo da quantidade de memória a ser alocada no instante $k+1$ é dado por:

$$u_{mem}(k+1) = u_{mem}(k) + \lambda \times v_{mem}(k) \times (u_{ref}^{mem} - r_{mem}(k))/u_{ref}^{mem} \quad (1)$$

onde,

$$r_{mem} = \frac{v_{mem}(k)}{u_{mem}(k)} \quad (2)$$

A segunda solução é descrita por Moltó et al. (2013), na qual os autores apresentam um mecanismo que realiza o monitoramento do uso de memória da VM e caso o percentual de memória livre esteja fora de determinada faixa, pode-se alocar ou liberar memória. As regras de elasticidade são aplicadas quando a porcentagem de memória livre no instante k for menor que 80% ou maior que 120% do *Memory Overprovisioning Percentage* (MOP), que corresponde à porcentagem de memória que deseja-se deixar disponível. Por exemplo, se o usuário escolher um MOP = 10%, o mecanismo de alocação será acionado quando a memória livre da VM for menor ou maior do que 10% do total de memória disponível. Assim, se a quantidade de memória livre for inferior a 8% o mecanismo aloca mais memória. Se superior a 12%, o mecanismo libera memória.

A quantidade de memória a ser alocada no tempo $k+1$ é dada por:

$$u_{mem}(k+1) = u_{mem}(k) \times (1 + MOP) \quad (3)$$

O terceiro mecanismo implementado é descrito no trabalho de Jenitha e Veeramani (2014), que calcula a alocação de memória para o instante $k+1$ usando uma equação baseada em médias móveis exponenciais ponderadas por meio da equação:

$$u_{mem}(k+1) = \alpha \times u_{mem}(k) + (1-\alpha) \times v_{mem}(k) \quad (4)$$

onde α é obtido através de:

$$\alpha = \frac{v_{mem}(k)}{u_{mem}(k)} \quad (5)$$

4.2. Métricas Implementadas

O EMA-Bench utiliza o conjunto de métricas propostas por Herbst et al. (2016), uma vez que se mostra apropriado para a avaliação da alocação elástica de memória. Nesta proposta dois aspectos principais são avaliados: precisão (*accuracy*) e o tempo dispendido nos estados de sub e superprovisionamento. A precisão do dimensionamento de recursos é calculada em relação à quantidade de recursos provisionados e a quantidade de recursos realmente utilizados. O tempo de provisionamento está relacionado à proporção de tempo em cada estado.

Dado o resultado referente às alocações fornecido ao final da execução de um mecanismo de elasticidade, calcula-se:

$\sum A$ - tempo acumulado no estado de subprovisionamento;
$\sum U$ - quantidade acumulada de recursos faltantes;
$\sum B$ - tempo acumulado no estado de superprovisionamento;
$\sum O$ - quantidade acumulada de recursos excedentes.

Os valores para $\sum A$, $\sum U$, $\sum B$ e $\sum O$ são obtidos a partir da comparação do valor de demanda no tempo k com o total de memória alocada pelo método no tempo k. Se o total alocado é menor que o demandado, indicando subprovisionamento, incrementa-se $\sum U$ pelo valor absoluto da diferença entre o alocado e demandado e $\sum A$ é incrementado em 1. Se o total alocado é maior que o demandado, indicando superprovisionamento, incrementa-se $\sum O$ pelo valor absoluto da diferença entre o alocado e demandado e $\sum B$ é incrementado em 1.

A partir desses valores é possível definir a precisão, calculada através da média dos desvios absolutos entre os recursos alocados e suas respectivas demandas reais de recursos, normalizadas pelo tempo de avaliação T. Neste caso define-se duas métricas: P_u que corresponde à precisão de alocação e P_d correspondente à precisão de desalocação.

$$P_u = \frac{\sum U}{T} \qquad P_d = \frac{\sum O}{T} \tag{6}$$

Similarmente, pode-se calcular o tempo de provisionamento somando a quantidade total de tempo gasto em um estado de subprovisionamento ($\sum A$) ou superprovisionamento ($\sum B$) normalizado pela duração do período de medição T. Assim, o tempo total gasto em estados sub ou superprovisionados é dado, respectivamente, por:

$$Q_u = \frac{\sum A}{T} \qquad Q_d = \frac{\sum B}{T} \tag{7}$$

Usando as métricas propostas para avaliar os mecanismos de alocação elástica de memória, espera-se que Q_d apresente valores próximo de 1 (consequentemente com Q_u próximo de zero), e P_d apresente valores pequenos, indicando que não há, ou há poucos, pontos de subprovisionamento e que não há alocação excessiva de recursos.

4.3. Parâmetros de Entrada e Saídas

Conforme apresentado na Seção 4 e ilustrado na Figura 1, o EMA-Bench possui um conjunto de entradas e saídas bem definidos. Os parâmetros de entrada são fornecidos por meio de linha de comando na forma:

```
ema-bench perfil_memoria mecanismo [parâmetros]
```

O parâmetro `perfil_memoria` refere-se ao nome do arquivo que contém o perfil de memória obtido a partir de um mecanismo de monitoramento. O arquivo, de texto e sem formatação, consiste de n linhas contendo valores inteiros, cada um associado ao uso de memória no instante k da coleta. O parâmetro `mecanismo` é utilizado para a escolha de uma solução para a avaliação. As atuais opções são: `heo`, `molto` e `jenitha`, correspondendo aos três mecanismos implementados até o momento. Dependendo do mecanismo selecionado tem-se um conjunto de parâmetros de configuração próprios. Por exemplo, optando pelo mecanismo `heo` os parâmetros λ e u_{ref}^{mem} devem ser informados; o parâmetro MOP deve ser informado para `molto`; e nenhum parâmetro adicional é necessário para `jenitha`.

Após a execução o EMA-Bench retorna um arquivo de saida (*output.dat*) com n linhas contendo um número inteiro que corresponde a alocação realizada pelo mecanismo escolhido para o instante k. O arquivo de alocações pode ser usado em uma ferramenta de plotagem, permitindo a visualização gráfica dos resultados. Além disso, são retornadas em tela duas tabelas contendo os valores para as métricas e com os cálculos de custo.

Na tabela de cálculo de custos, apresenta-se um comparativo entre os custos de alocação estática e elástica. Para o cálculo no caso da alocação estática, considera-se a quantidade de memória necessária para satisfazer o maior pico de demanda para todo o período avaliado multiplicado pelo custo por unidade de tempo (dólar por MB/min., por exemplo). Na contabilização dos custos da alocação elástica, considerou-se o custo por unidade de tempo multiplicado pelo montante alocado a cada instante k.

5. Avaliação Experimental

Nessa seção apresenta-se um conjunto de experimentos conduzidos com o *benchmark* proposto. Para a realização dos testes foram utilizados 5 perfis de consumo de memória (nomeados de R1 a R5) obtidos de de máquinas virtuais hospedadas no *data center* de uma empresa de tecnologia localizada na cidade de Cascavel-PR. O monitoramento do consumo de memória foi realizado a cada minuto por aproximadamente 6 dias, totalizando 8.747 coletas para cada um dos 5 perfis.

Os perfis foram selecionados de modo a apresentar comportamentos bastante heterogêneos, fornecendo situações distintas para a avaliação dos mecanismos. O perfil R1 corresponde a um controlador de domínios do *data center*, R2 é um servidor de impressão, R3, R4 e R5 correspondem a servidores de aplicação. O hipervisor utilizado no *data center* para a criação das VMs é o Microsoft Hiper-V.

Ao todo foram realizados 25 testes utilizando o EMA-Bench, nos quais realizou-se cinco testes para cada um dos cinco perfis, com o intuito de se verificar se os métodos apresentados na literatura são apropriados para estes cenários. Utilizou-se nos testes um computador com processador Intel Core i5-2450M @ 2.50 GHz e 6 GB de memória RAM com Sistema Operacional Linux Ubuntu 14.04 LTS o compilador utilizado é o GCC 5.4.0.

Cada um dos perfis foi testado com o mecanismo proposto por Heo (2009) utilizando os parâmetros $u_{ref}^{mem} = 90\%$ e 70% e $\lambda = 1$, empregados no trabalho original. Os resultados obtidos para cada um dos perfis são apresentados graficamente na Figura 2, onde é possível verificar que a alocação de memória realizada satisfaz as demandas du-

rante todo o tempo para todos os perfis. Os valores apresentados nas Tabelas 1 e 2 para a métrica $Q_d = 1$, significando que em 100% do tempo houve alguma sobra de memória, atesta isso.

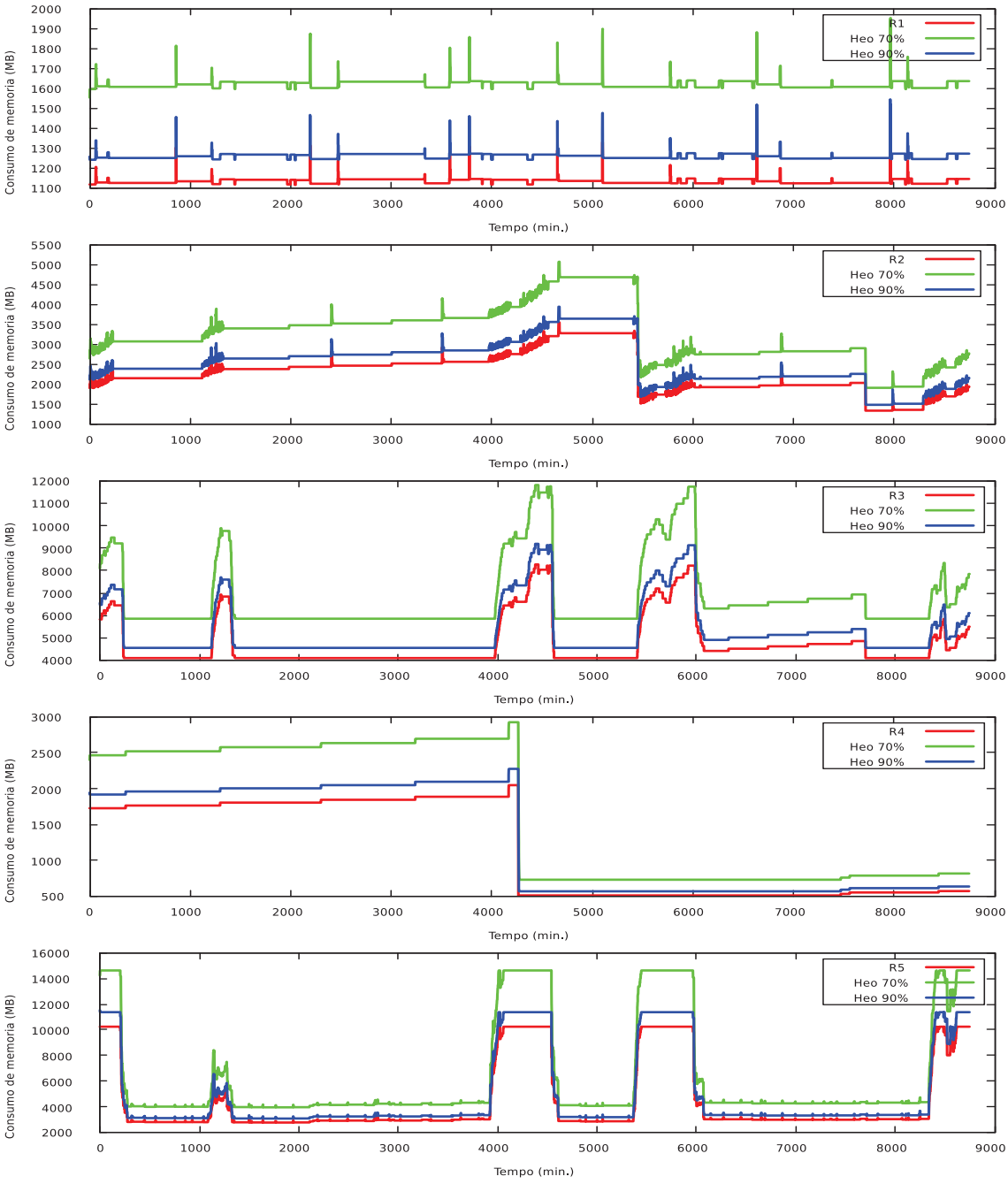

Figura 2. Heo: Alocações realizadas.

Considerando que a solução de Heo determina uma porcentagem alvo para o uso de memória (90 % e 70%, nos testes), é esperado que se tenha superprovisionamento em grande parte das execuções. Isso é evidenciado pela métrica $\sum O$ que mostra o somatório das quantias de memória superprovisionadas ao longo do período considerado. Como esperado, os resultados com $u_{ref}^{mem} = 90\%$ apresentaram valores inferiores para $\sum O$ quando comparados com os testes empregando $u_{ref}^{mem} = 70\%$.

Tabela 1. Heo 90%: Resultado das métricas.

Perfil	$\sum A$	$\sum U$	$\sum B$	$\sum O$	P_u	P_d	Q_u	Q_d
R1	0	0	8.747	1.099.400	0,0000	125,6744	0,0000	1
R2	0	0	8.747	2.216.153	0,0000	253,3325	0,0000	1
R3	0	0	8.747	4.615.865	0,0000	527,6480	0,0000	1
R4	0	0	8.747	1.121.759	0,0000	128,2303	0,0000	1
R5	0	0	8.747	4.323.920	0,0000	494,2753	0,0000	1

Tabela 2. Heo 70%: Resultado das métricas.

Perfil	$\sum A$	$\sum U$	$\sum B$	$\sum O$	P_u	P_d	Q_u	Q_d
R1	0	0	8.747	4.249.098	0,0000	485,7222	0,0000	1
R2	0	0	8.747	8.544.932	0,0000	976,7869	0,0000	1
R3	0	0	8.747	17.801.887	0,0000	2.034,9667	0,0000	1
R4	0	0	8.747	4.337.233	0,0000	495,7971	0,0000	1
R5	0	0	8.747	16.664.225	0,0000	1.904,9182	0,0000	1

Assim, de modo geral, a solução Heo consegue fornecer uma alocação elástica de memória satisfatória e dada sua configurabilidade, pode se adaptar com sucesso a diferentes perfis.

Os experimentos com a solução proposta por de Moltó et al. (2013) foram realizados com o parâmetro MOP assumindo valores 10% e 30%, assim como apresentado no trabalho original. Assim, no primeiro caso espera-se ter aproximadamente 10% memória livre e 30% no segundo. As alocações realizadas usando esta abordagem podem ser observados na Figura 3 e os resultados das métricas são apresentados nas Tabelas 3 e 4.

Tabela 3. Moltó 10%: Resultado das métricas.

Perfil	$\sum A$	$\sum U$	$\sum B$	$\sum O$	P_u	P_d	Q_u	Q_d
R1	7	467	8.740	989.004	0,0534	113,0549	0,0008	0,9991
R2	10	404	8.737	1.990.936	0,0462	227,5876	0,0011	0,9987
R3	0	0	8.747	4.150.330	0,0000	474,4319	0,0000	1
R4	0	0	8.747	883.272	0,0000	100,9685	0,0000	1
R5	7	741	8.740	3.870.026	0,0847	442,3898	0,0008	0,9991

Tabela 4. Moltó 30%: Resultado das métricas.

Perfil	$\sum A$	$\sum U$	$\sum B$	$\sum O$	P_u	P_d	Q_u	Q_d
R1	0	0	8.747	2.972.113	0,0000	339,7477	0,0000	1
R2	0	0	8.747	5.977.232	0,0000	683,2684	0,0000	1
R3	0	0	8.747	12.455.647	0,0000	1.423,8280	0,0000	1
R4	0	0	8.747	3.030.940	0,0000	346,4724	0,0000	1
R5	0	0	8.747	11.615.460	0,0000	1.327,7847	0,0000	1

Diferentemente dos resultados apresentados pela solução de Heo, o mecanismo de Moltó não obteve resultados totalmente positivos para todos testes. Pode-se observar que para $MOP = 10\%$ há períodos de subprovisionamento para os perfis R1, R2 e R5, mesmo que pouco significativos e breves, como mostram as métricas apresentadas na Tabela 3. O subprovisionamento ocorre nos momentos onde as demandas de memória aumentam

rapidamente e como há pouca memória excedente (cerca de 10%), o mecanismo apresenta um intervalo de tempo entre a detecção da demanda e a alocação efetiva da memória demandada. O mesmo não ocorre com $MOP = 30\%$, pois a quantidade extra de memória alocada mascara o atraso na alocação. Isso fica claro ao comparar a métrica P_d, que mostra a quantidade média de memória excedente, dos testes empregando diferentes valores para MOP.

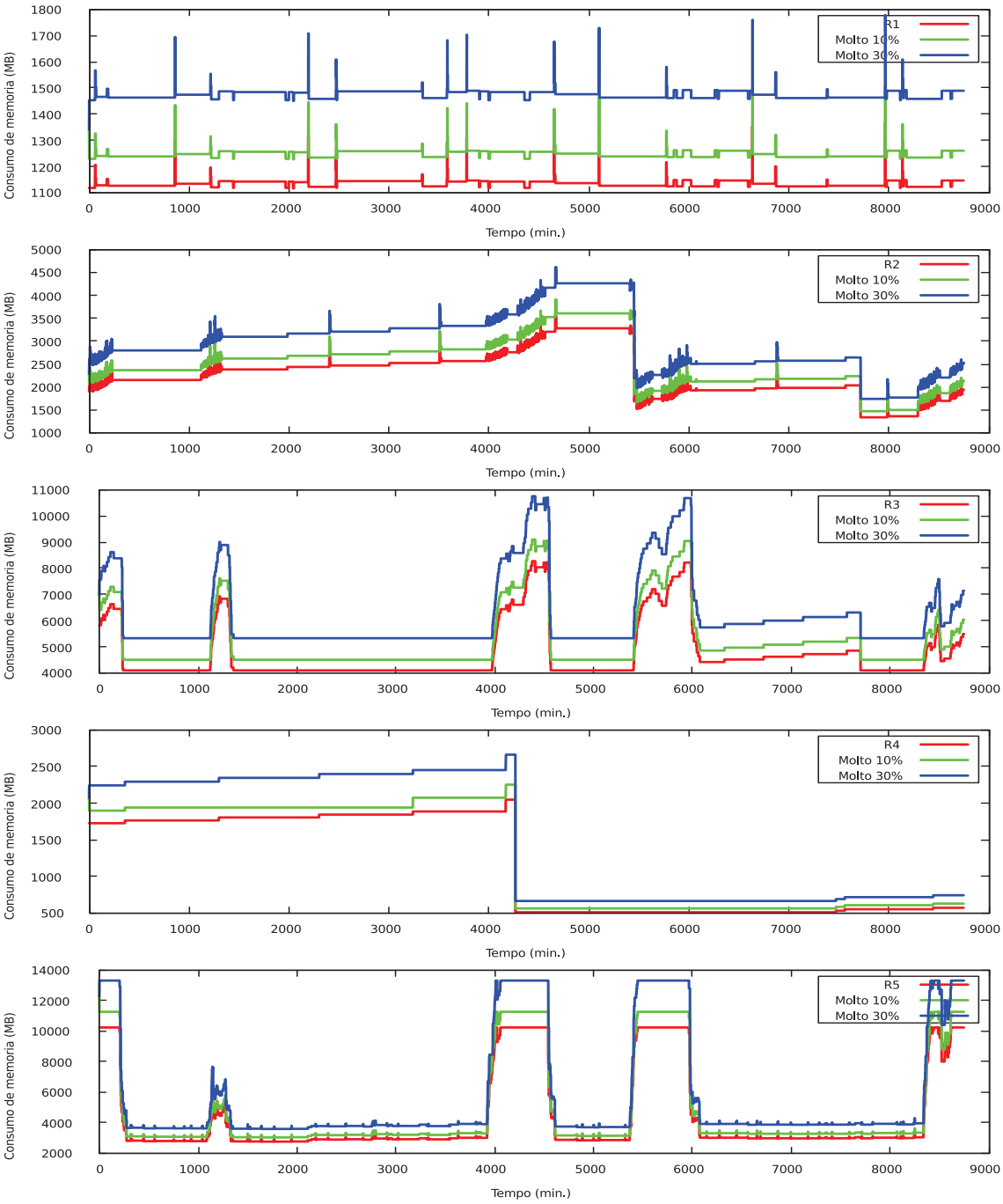

Figura 3. Moltó: Alocações realizadas.

O mecanismo proposto por Jenitha e Veeramani possui um comportamento diverso dos apresentados pelos demais. Por não prever alocação de memória livre excedente, a solução oferece uma alocação muito próxima da demanda, como pode-se observar na Figura 4 e na Tabela 5. Com isso, há períodos de subprovisionameto para todos os perfis ($\sum A \neq 0$), embora a métrica P_u apresente valores baixos. Destaca-se o comportamento desta solução para o perfil R4, onde em 35% $(1 - (Q_u + Q_d))$ do tempo do experimento a quantia de memória foi exatamente o demandado pela aplicação.

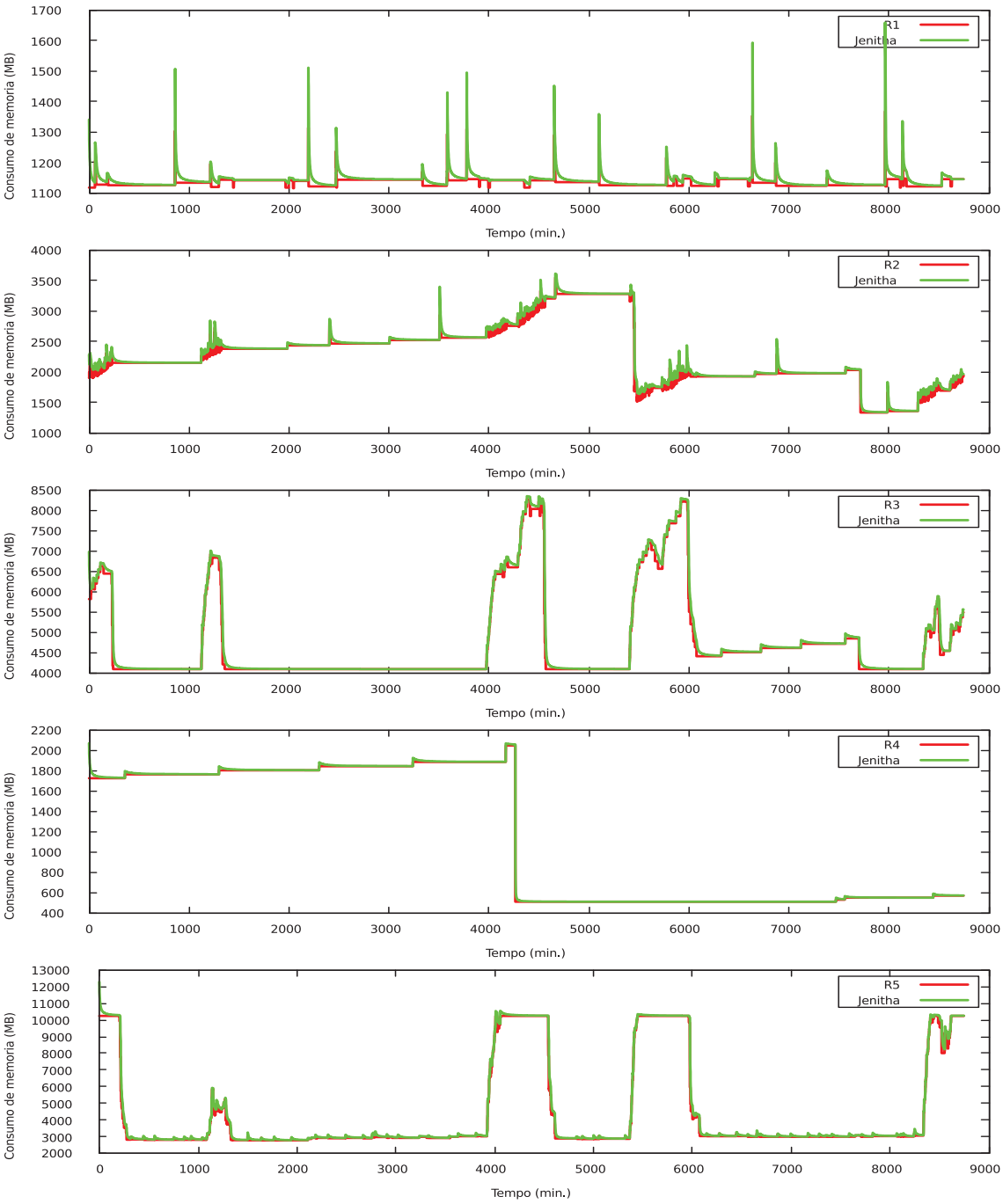

Figura 4. Jenitha: Alocações realizadas.

Tabela 5. Jenitha e Veeramani: Resultado das métricas.

Perfil	$\sum A$	$\sum U$	$\sum B$	$\sum O$	P_u	P_d	Q_u	Q_d
R1	31	1.936	8.597	92.804	0,2213	10,6086	0,0035	0,9827
R2	150	6.584	8.592	266.141	0,7526	30,4231	0,0171	0,9822
R3	119	9.050	8.628	407.279	1,0345	46,5568	0,0136	0,9863
R4	8	298	5678	3.6974	0,0341	4,2266	0,0009	0,6491
R5	163	19.949	8.584	694.881	2,2804	79,4331	0,0186	0,9813

Considerando que sob a óptica do cliente, a principal vantagem da alocação elástica é a manutenção do SLA com redução de custos, apresenta-se na Tabela 6 resultados de custo total para cada um dos perfis apresentados utilizando alocação estática e elástica.

Os valores foram calculados tomando como base os valores cobrados pelo provedor Profitbricks[1], que é de US$ 0,0070 por GB/hora ou US$ 0,000117 GB/minuto. É possível notar na tabela que há apenas dois casos em que o valor total é superior ao alcançado pela alocação estática. Ambos os casos ocorrem para o perfil R1 com mecanismos que preservam o nível de uso de memória em aproximadamente 70% (Heo 70% e Moltó 30%), já que em ambos os casos é previsto superprovisionamento de recursos. Nos demais casos, pode-se observar reduções bastante significativas nos custos, em alguns casos superior a 50%, o que mostra que o uso da elasticidade vertical de memória pode ser realmente efetiva.

Tabela 6. Comparativo de custos: alocação estática versus alocação elástica. Valores em Dólares Americanos.

Perfil	Estática	Heo 90%	Heo 70%	Moltó 10%	Moltó 30%	Jenitha
R1	1,37	1,26 (92%)	*1,61 (118%)*	1,24 (90%)	*1,47 (107%)*	1,14 (83%)
R2	3,55	2,52 (71%)	3,24 (91%)	2,50 (70%)	2,95 (83%)	2,30 (65%)
R3	8,25	5,26 (64%)	6,76 (82%)	5,21 (63%)	6,15 (75%)	4,78 (58%)
R4	2,04	1,28 (63%)	1,65 (81%)	1,25 (61%)	1,50 (74%)	1,16 (57%)
R5	10,22	4,92 (48%)	6,33 (62%)	4,87 (48%)	5,75 (56%)	4,50 (44%)

6. Conclusão

Considerando a falta de métricas e metodologias para a avaliação de mecanismos de elasticidade vertical de memória, a contribuição deste trabalho foi a proposição de um *benchmark* para este fim. Perante os trabalhos relacionados, o diferencial da proposta apresentada foi implementar um conjunto de métricas para a avaliação da eficácia e eficiência dos mecanismos, bem como informações sobre os custo envolvidos na adoção de uma determinada solução.

Os resultados mostram que a ferramenta proposta é capaz de auxiliar o usuário na definição de qual é o melhor mecanismo a ser adotado, garantindo redução de custos e a manutenção da qualidade do serviço. Além disso, o EMA-Bench pode ser utilizado por outros pesquisadores na comparação e refinamento de experimentos e de soluções elásticas.

[1] `https://www.profitbricks.com/`

Os trabalhos futuros incluem a implementação de novos mecanismos de elasticidade de memória, incluindo mecanismos preditivos, que por terem implementação mais complexa ficaram fora do escopo deste trabalho. Pretende-se ainda melhorar a qualidade das informações retornadas ao usuário por meio da geração automática de gráficos de alocação, bem como de relatórios mais completos. Está previsto também a amplição do escopo do trabalho, adicionando também ao *benchmark* a possibilidade de avaliar a elasticidade vertical para outros tipos de recurso, tal como CPU e armazenamento.

Agradecimentos

Os autores agradecem à empresa Constel Tecnologia por fornecer os perfis de utilização de memória dos servidores virtualizados e ao CNPq pelo financiamento parcial desta pesquisa.

Referências

Beltran, M. (2016). Defining an elasticity metric for cloud computing environments. In *Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools*, VALUETOOLS'15, pages 172–179. ICST.

Ben-Yehuda, A. O., Ben-Yehuda, M., Schuster, A., and Tsafrir, D. (2012). The resource-as-a-service (raas) cloud. In *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Computing*, HotCloud'12, pages 1–5. USENIX.

Coutinho, E. F., Rego, P. A. L., Gomes, D. G., and de Souza, J. N. (2014). Métricas para avaliação da elasticidade em computação em nuvem baseadas em conceitos da física. In *Anais do XII Workshop de Computação em Clouds e Aplicações*, WCGA 2014, pages 55–66. SBC.

Farokhi, S., Jamshidi, P., Bayuh Lakew, E., Brandic, I., and Elmroth, E. (2016). A hybrid cloud controller for vertical memory elasticity. *Future Gener. Comput. Syst.*, 65(C):57–72.

Galante, G. and Bona, L. C. E. (2012). A survey on cloud computing elasticity. In *Proceedings of the International Workshop on Clouds and eScience Applications Management*, CloudAM'12, pages 263–270. IEEE.

Galante, G., Erpen De Bona, L. C., Mury, A. R., Schulze, B., and da Rosa Righi, R. (2016). An analysis of public clouds elasticity in the execution of scientific applications: a survey. *Journal of Grid Computing*, 14(2):193–216.

Gong, Z., Gu, X., and Wilkes, J. (2010). Press: Predictive elastic resource scaling for cloud systems. In *Proceedings of the 6th International Conference on Network and Service Management*, CNSM'10, pages 9–16. IEEE.

Heo, J., Zhu, X., Padala, P., and Wang, Z. (2009). Memory overbooking and dynamic control of xen virtual machines in consolidated environments. In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management*, pages 630–637. IEEE.

Herbst, N., Krebs, R., Oikonomou, G., Kousiouris, G., Evangelinou, A., Iosup, A., and Kounev, S. (2016). Ready for rain? A view from SPEC research on the future of cloud metrics. Technical Report SPEC-RG-2016-01, SPEC Research Group - Cloud Working Group.

Herbst, N. R., Kounev, S., Weber, A., and Groenda, H. (2015). Bungee: An elasticity benchmark for self-adaptive iaas cloud environments. In *Proceedings of the 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, SEAMS '15, pages 46–56. IEEE.

Hwang, K., Bai, X., Shi, Y., Li, M., Chen, W., and Wu, Y. (2016). Cloud performance modeling with benchmark evaluation of elastic scaling strategies. *IEEE Trans. Parallel Distrib. Syst.*, 27(1):130–143.

Islam, S., Lee, K., Fekete, A., and Liu, A. (2012). How a consumer can measure elasticity for cloud platforms. In *Proceedings of the 3rd ACM/SPEC International Conference on Performance Engineering*, pages 85–96. ACM.

Jenitha, V. H. A. and Veeramani, R. (2014). Dynamic memory allocation using ballooning and virtualization in cloud computing. *IOSR Journal of Computer Engineering*, 16(2):19–23.

Li, K. (2017). Quantitative modeling and analytical calculation of elasticity in cloud computing. *IEEE Transactions on Cloud Computing*, PP(99):1–14.

Moltó, G., Caballer, M., Romero, E., and de Alfonso, C. (2013). Elastic memory management of virtualized infrastructures for applications with dynamic memory requirements. *Procedia Computer Science*, 18:159–168.

Spinner, S., Herbst, N., Kounev, S., Zhu, X., Lu, L., Uysal, M., and Griffith, R. (2015). Proactive memory scaling of virtualized applications. In *Proceedings of the 8th International Conference on Cloud Computing*, pages 277–284. IEEE.

Tan, Y., Nguyen, H., Shen, Z., Gu, X., Venkatramani, C., and Rajan, D. (2012). Prepare: Predictive performance anomaly prevention for virtualized cloud systems. In *Proceedings of the 32nd International Conference on Distributed Computing Systems*, ICDCS, pages 285–294. IEEE.

Zhang, W.-Z., Xie, H.-C., and Hsu, C.-H. (2017). Automatic memory control of multiple virtual machines on a consolidated server. *IEEE Transactions on Cloud Computing*, 5(1):2–14.