



ARTIGO ORIGINAL

## Escalonamento de máquinas virtuais baseado em custo e tolerante a anomalias de tráfego de rede para Dados-como-Serviço

### Cost-based virtual machine scheduling resilient to network traffic anomalies for Data as a Service

Ana Cristina Alves de Oliveira <sup>1</sup> and Marco Aurélio Spohn <sup>2</sup>

<sup>1</sup>Laboratório de Tecnologia da Informação (LATIN). Instituto Federal da Paraíba (IFPB) – Campus Campina Grande, <sup>2</sup>Universidade Federal da Fronteira Sul (UFFS) – Campus Chapecó

\*[ana.oliveira@ifpb.edu.br](mailto:ana.oliveira@ifpb.edu.br); [marco.spohn@uffs.edu.br](mailto:marco.spohn@uffs.edu.br)

Recebido: 23/06/2020. Revisado: 17/09/2020. Aceito: 14/10/2020.

#### Resumo

Os serviços de computação em nuvem são executados em diferentes níveis de abstração, envolvendo mais de um tipo de provedor, que alterna entre os papéis de cliente e provedor. Prejuízos causados por falhas são propagados nos diferentes níveis. Mecanismos para reduzir custos e prejuízos são importantes dada a larga escala. Neste trabalho foi proposto um modelo para escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias de tráfego. O modelo de negócio de Dados-como-Serviço (DaaS) foi utilizado como estudo de caso para avaliar o impacto de anomalias. Os resultados demonstraram melhorias significativas na redução de custos e as perdas financeiras foram reduzidas para entre 15% e 26% dos valores convencionais.

**Palavras-Chave:** Cloud computing; Dados-como-Serviço; detecção de anomalias; escalonamento baseado em custo; análise de tráfego de rede

#### Abstract

Cloud computing services run on top of different abstraction levels, involving many actors playing alternate roles as customers or providers. Financial losses caused by faults are propagated throughout all the systems' levels. Mechanisms to reduce costs and losses are important in face of the large scale of cloud computing systems. In this paper, we propose a cost-based scheduling model integrated into the network traffic anomaly detection engine. The Data-as-a-Service (DaaS) business model was taken as a case study to evaluate the impact resulting from anomalous traffic. The results showed a significant reduction in costs and financial losses, with gains ranging from 15% to 26% of the conventional method's financial losses.

**Keywords:** Cloud computing; data as a service; anomaly detection; cost based scheduling; network traffic analysis

#### 1 Introdução

Clientes de serviços de computação em nuvem não possuem domínio sobre todos os níveis de execução dos

serviços contratados. Existem camadas abstratas para que um serviço seja executado e pode haver mais de um provedor envolvido no fornecimento de um mesmo serviço. Um provedor de dados pode contratar um provedor de infraestrutura para executar VMs que, por sua vez, pode contratar um provedor de serviços de rede.

Os acordos de nível de serviço (*Service Level Agreement – SLA*) entre provedor e cliente são estabelecidos quando o serviço é contratado, assumindo que as metas técnicas definidas para cada serviço devem ser garantidas. O monitoramento das metas técnicas é realizado pelo provedor de serviços. Após a contratação ou execução do serviço de computação em nuvem, o sistema de tarifação do provedor gera uma fatura, a qual é paga pelo cliente de acordo com a política de pagamento em vigor. Tanto o provedor quanto o cliente possuem seus sistemas de contabilidade gerencial, que analisam os custos e as receitas. Caso ocorra alguma anormalidade no provimento de um serviço, mesmo que o cliente observe a falha, os meios para comprovar essa irregularidade e contestar a tarifação de forma legítima não são claros.

Novas estratégias para a implantação de Governança de TI (planejamento, controle, monitoramento e gestão) são necessárias para promover a transparência na prestação de serviços. Os prestadores de serviços devem encontrar uma forma de atender às demandas de seus clientes, garantindo os princípios de confiabilidade, confidencialidade, integridade e disponibilidade (Khatri and Brown, 2010).

Os clientes de serviços de computação em nuvem em geral não tem acesso ao monitoramento de métricas de desempenho dos serviços. A maioria dos SLAs estão relacionados ao nível de rede e apenas em torno de 15% dos provedores fornecem informações sobre métricas de não-conformidade de SLAs (Oliveira, Dora, Spohn and Oliveira, 2015).

O processo de **tarifação confiável** entre as partes cliente e provedora de serviços de computação em nuvem deve envolver um modelo de monitoramento compartilhado de métricas (Oliveira et al., 2014). O acesso padronizado a métricas de desempenho dos serviços contratados permite ao cliente aferir as tarifas cobradas e também fornece subsídios e incentiva o provedor de serviços a melhorar a alocação de seus recursos, bem como fornece indicadores para investimentos em infraestrutura de modo a evitar perdas financeiras pelo pagamento de multas.

Durante a última década, houve um crescimento contínuo de fontes de dados, como aplicativos para telefones inteligentes, sensores e portais. Diante desse contexto, emergiu um modelo de negócio para computação em nuvem chamado *Data-as-a-Service* (DaaS), no qual várias pequenas a médias empresas (PMEs) partilham os altos custos necessários de infraestrutura para rastrear e armazenar uma cópia de “toda” a Web (ou pelo menos uma grande parte dela) e executar consultas específicas sobre negócios em um determinado conjunto de dados (Oliveira, Fetzer, Martin and Spohn, 2015).

Para representar a infraestrutura computacional dos sistemas de DaaS, será adotado um modelo não homo-

gêneo de **micronuvem**, em vez de um único centro de dados dedicado para realizar o rastreamento, armazenamento e tratamento de dados. Micronuvens consistem em pequenas unidades de nós como, por exemplo, 20 máquinas físicas, com diferentes capacidades de armazenamento e processamento.

A arquitetura de monitoramento de computação em nuvem que disponibiliza informações sobre métricas de tráfego de rede tanto para o cliente quanto para o provedor foi proposta por Oliveira et al. (2014). Os autores propuseram um modelo de precificação de máquinas virtuais (VMs) para DaaS e um mecanismo de escalonamento baseado nesses custos; no entanto, a modelagem das VMs foi feita de modo aleatório e o modelo foi validado por meio de simulações.

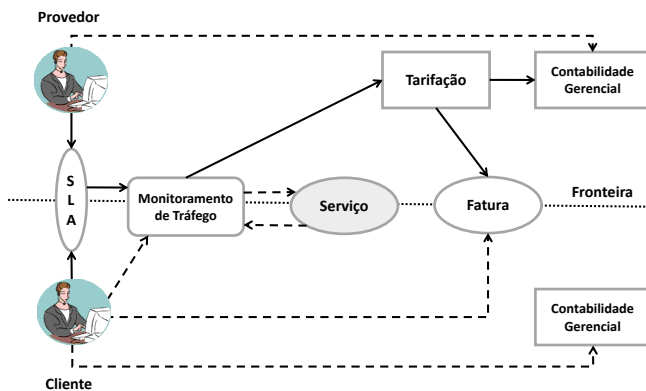
A contribuição deste trabalho é um modelo para *escalonamento de máquinas virtuais baseado em custo integrado ao mecanismo de detecção de anomalias de tráfego de rede*, que é uma extensão do escalonamento baseado em custo proposto por Oliveira, Fetzer, Martin and Spohn (2015). A metodologia empregada na validação deste trabalho envolveu experimentação em ambiente real e simulação. Os experimentos foram conduzidos para caracterizar o comportamento de execução das VMs e o tráfego de rede (*traces*) no processamento de DaaS em uma nuvem privada. As métricas de execução e os (*traces*) do ambiente real foram utilizados para calibrar o simulador que implementou o modelo de custo proposto.

## 2 Problematização e Objetivos

Há pouca informação sobre custos em descumprimentos de SLAs e seus decorrentes prejuízos financeiros para o negócio. A Fig. 1 apresenta uma visão geral da fronteira que separa o acesso de clientes e provedores aos dados sobre a execução dos serviços e contempla uma proposta de acesso compartilhado a informações sobre SLAs e monitoramento de tráfego, que é tratada neste trabalho como processo de tarifação confiável.

O objetivo deste trabalho é analisar o impacto do custo de tráfego anômalo quando um provedor de DaaS é também cliente de IaaS. As decisões do mecanismo de escalonamento levam em conta os resultados dessa análise de impacto. As VMs são escalonadas para os centros de dados que minimizem o custo total de prestação do serviço, o que envolve *trade-off* entre o escalonamento baseado em custo (proposto anteriormente) e o prejuízo causado pela perda financeira devido ao não cumprimento dos SLAs.

Este trabalho envolve seis fases. Na fase 1 foi realizado um experimento real com uma aplicação de DaaS para analisar como as VMs se comportam na prática. As variáveis do modelo analítico foram ajustadas conforme essas estimativas. Na fase 2 ocorre o processo de detecção de anomalias. Durante o provimento dos serviços de DaaS é feito o monitoramento do tráfego e quando ocorre uma violação, essas anomalias são identificadas e contabilizadas. A injeção de anomalias é feita na fase 3. Ela segue um modelo analítico proposto por Xiong et al. (2011). Na fase 4 é feita uma predição



**Figura 1:** Problematização: visão geral da fronteira entre cliente e provedor para execução do serviço contratado.

do impacto do custo de uma anomalia para o tipo de serviço a ser escalonado, caso a VM venha a ser executada em um centro de dados (*Data Center*, DC) que está apresentando problemas de disponibilidade. Na fase 5 é feito o escalonamento. O escalonamento baseado nos custos de cada DC é integrado ao módulo de detecção de anomalias. Na fase 6 ocorre a análise dos resultados, onde se verifica o ganho do escalonamento ao se empregar as estimativas de perda financeira em detrimento de apenas se basear nos custos pontuais de cada DC.

### 3 Fundamentação Teórica

Nesta seção será apresentado o modelo de custo para Dados–como–Serviço, o mecanismo de escalonamento baseado nos custos para alocar VMs e uma visão geral sobre técnicas de detecção de anomalias de tráfego.

O mecanismo de escalonamento baseado em custo considera os custos para atender aos requisitos de processamento, armazenamento e comunicação de cada VM, tendo como objetivo minimizar os custos.

#### 3.1 Modelo de Custo para Dados–como–Serviço

Nesta seção serão apresentados o modelo de custo para DaaS e o algoritmo de escalonamento baseado no mesmo, onde as *tarefas* (tratadas por *máquinas virtuais*) como, por exemplo, rastreamento e processamento, são escalonadas em um conjunto de micronuvens (Oliveira, Fetzer, Martin and Spohn, 2015).

Uma *nuvem* pode ser representada como um conjunto de **centros de dados**, cada um possuindo máquinas físicas para hospedar as VMs e para armazenamento de dados. O termo *micronuvem* representa um pequeno conjunto de elementos de processamento localizados na mesma rede local (LAN), sem necessariamente significar a infraestrutura de um centro de dados complexo.

Contudo, sem perda de generalidade, para compatibilizar conceitos, será utilizado o termo *centro de dados* ao longo deste trabalho com o mesmo conceito de *micronuvem*. Desse modo, pode-se representar genericamente diferentes tipos de nuvens; incluindo nuvens privadas, públicas ou federadas.

Um trabalho (*job*) em sistemas de DaaS é genericamente representado por uma consulta (*query*) a grandes quantidades de dados que é realizada de forma distribuída (*Big Data*). As consultas são divididas em tarefas, e cada tarefa é executada por meio de uma **máquina virtual** (VM). O processamento de uma consulta segue o estilo de programação *MapReduce* (Dean and Ghemawat, 2008). Para realizar uma consulta, uma ou mais VMs lêem e processam os dados e, em seguida, enviam os resultados para o destino por meio da rede de comunicação. O processo de executar uma computação sobre um conjunto de dados é a fase de *map* (mapear) e o processo de redução dos dados mapeados para o cálculo do resultado final é chamado de *reduce* (reduzir). O modelo *MapReduce* está sendo amplamente aplicado para processar *Big Data* (Hurwitz et al., 2013).

Para executar uma tarefa, uma VM necessita acessar uma fonte de dados localizada na área de armazenamento. As **fontes de dados** (*data sources*) utilizadas para o processamento das tarefas são replicadas e encontram-se distribuídas pelos centros de dados. Neste modelo, uma tarefa acessa uma cópia dos dados replicados. As fontes de dados são persistidas usando um modelo de armazenamento baseado em **chave/valor** ou *Key/Value (K/V) store*. Exemplos de ferramentas que implementam o modelo de *K/V store* são a *Google BigTable* (Chang et al., 2006) e o arcabouço *Chubby* (Burrows, 2006).

Na visão do cliente de DaaS, os preços do serviço são entendidos como *custos*. Esses custos se baseiam em dois pressupostos:

- i. **VM e fonte de dados estão localizados no mesmo centro de dados:** se as VMs e os dados estiverem localizados no mesmo centro de dados, então não há custos de comunicação;
- ii. **VM e fonte de dados estão localizados em centros de dados diferentes:** os custos de comunicação são aplicáveis somente quando a VM enviar dados para fora do centro de dados em que está localizada (em outra rede).

Os componentes do modelo de precificação para dados como serviço são apresentados a seguir:

- Seja  $D$  o conjunto de centros de dados:
  - $D = \{d_1, d_2, \dots, d_{|D|}\}$ ;
- Seja  $U$  o conjunto de VMs utilizadas para consultas:
  - $U = \{u_1, u_2, \dots, u_{|U|}\}$ ;
- Seja  $S$  o conjunto de fontes de dados:
  - $S = \{s_1, s_2, \dots, s_{|S|}\}$ .

As **variáveis independentes** do modelo são as se-

guintes:

- $R_z$ : número, em bytes, de dados recebidos (lidos) pela VM  $u_z$ ;
- $T_z$ : número, em bytes, de dados transmitidos (escritos) pela VM  $u_z$ ;
- $r_z$ : fator de redução de dados da VM  $u_z$ ; ou seja, a razão entre  $T_z$  e  $R_z$ ;
- $k_z$ : a taxa de tempo para processar 1 byte lido pela VM  $u_z$ ; ou seja, a VM  $u_z$  requer  $k \cdot R$  horas para ser finalizada;
- $q_z$ : taxa de saída de bytes por hora da VM  $u_z$ , onde  $q_z = \frac{r_z}{k_z}$ ;
- $v_{d_i}$  ou  $v_i$ : custo de execução de uma VM durante 1 hora no centro de dados  $d_i$ ;
- $t_{d_i}$  ou  $t_i$ : custo do tráfego de rede no centro de dados  $d_i$ ; ou seja, o valor a ser pago para transmitir 1 byte tendo como origem o centro de dados  $d_i$ .

Diante das variáveis apresentadas, são definidos os seguintes vetores de custos:

- Seja  $\vec{v}$  o vetor de custos de execução de cada centro de dados em  $D$ :
  - $\vec{v} = \langle v_1, v_2, \dots, v_{|D|} \rangle$ ;
- Seja  $\vec{t}$  o vetor de custos de transferência de tráfego para cada centro de dados em  $D$ :
  - $\vec{t} = \langle t_1, t_2, \dots, t_{|D|} \rangle$ .

A variável dependente é o preço. Seja  $P(u_z, d_i, d_j)$  o preço para executar a VM da consulta  $u_z$  no centro de dados  $d_i$  e acessar os dados do centro de dados  $d_j$ ; onde  $1 \leq i, j \leq |D|$  e  $1 \leq z \leq |U|$ . O preço para executar uma consulta é o custo total relativo à execução da VM ( $Ce_z$ ) mais o custo de comunicação para transferência dos dados ( $Cd_z$ ), tendo em conta a localização da VM e dos dados. Neste sentido, os preços são definidos como se segue:

- Preço para executar uma VM de consulta em um centro de dados que contém os dados de entrada:** o custo relativo à execução de uma VM  $u_z$  no centro de dados  $d_i$  que contém a fonte de dados necessária ao processamento é dado pela Eq. 1.

$$\begin{aligned}
 P(u_z, d_i, d_i) &= P_i^z = Ce_z \\
 &= (k_z \cdot R_z) \cdot v_i + T_z \cdot t_i \\
 &= (k_z \cdot R_z) \cdot v_i + (R_z \cdot r_z) \cdot t_i \\
 &= (k_z \cdot R_z) \cdot v_i + (R_z \cdot q_z \cdot k_z) \cdot t_i \\
 &= k_z \cdot R_z \cdot (v_i + q_z \cdot t_i) \quad (1)
 \end{aligned}$$

- Preço para executar uma VM de consulta em um centro de dados e acessar dados armazenados em outro centro de dados:** o custo relativo à execução de uma VM  $u_z$  no centro de dados  $d_i$  e ler os dados do centro de dados  $d_j$  é dado pela Eq. 2. Esse preço corresponde ao custo para transmissão dos dados entre os centros de dados ( $Cd_z = R_z \cdot t_j$ ) acrescido do

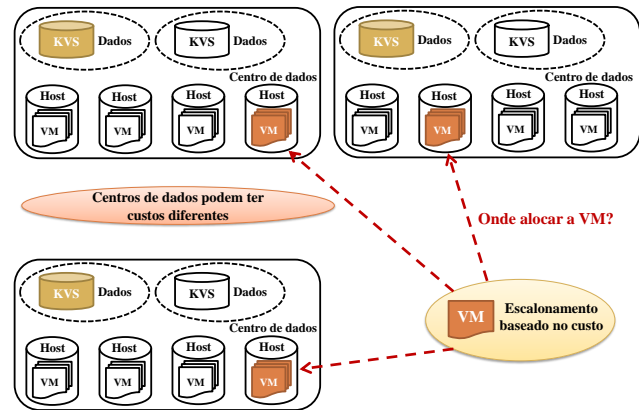


Figura 2: Estratégia de escalonamento baseada em custo.

preço apresentado na Eq.1.

$$\begin{aligned}
 P(u_z, d_i, d_j) &= P_{i,j}^z = Ce_z + Cd_z \\
 &= k_z \cdot R_z \cdot (v_i + q_z \cdot t_i) + R_z \cdot t_j \\
 &= R_z \cdot [t_j + k_z \cdot (v_i + q_z \cdot t_i)] \quad (2)
 \end{aligned}$$

### 3.2 Escalonamento Baseado em Custo

Uma solução para redução de custos na execução das consultas de DaaS é empregar uma **estratégia de escalonamento de máquinas virtuais baseada em custo**. Nem sempre o custo para executar uma VM que realiza o processamento parcial ou integral de uma consulta no centro de dados que armazena a fonte de dados é menor do que se esses dados fossem recuperados via rede.

Se o custo de execução da VM somado ao custo de comunicação for inferior ao custo de execução da VM em um centro de dado que contenha uma réplica da fonte de dados necessária ao processamento da consulta, então essa consulta será alocada no centro de dados de menor custo de execução e os dados serão recebidos via rede. O centro de dados que apresenta o menor custo é o primeiro a receber VMs. Quando sua capacidade de hospedar VMs encontra-se saturada, então o centro de dados de conseguinte menor preço passa a receber as novas VMs a serem escalonadas.

Os principais objetos que compõem a estratégia de escalonamento baseada em custo são mostrados na Fig. 2.

Qualquer decisão é tomada considerando os dados obtidos a partir da fase de análise feita pelo **controlador** (*controller*), ou a partir do **monitor** que verifica os SLAs esperados, bem como outros fatores relacionados à infraestrutura da nuvem, incluindo o tempo de inicialização das VMs e **modelos de precificação e de custo de anomalias**.

O Algoritmo 1 apresenta, de forma genérica, o escalonamento baseado em custo da máquina virtual  $u_z$ , que utiliza a fonte de dados  $s_z$  como entrada para o processamento da consulta. O algoritmo recebe como entrada os vetores contendo todos os custos de execução da VM ( $C\bar{e}_z$ ) e outro vetor contendo os custos de recebimento dos dados ( $C\bar{d}_z$ ).

---

**Algoritmo 1:** Algoritmo de escalonamento baseado em custo para a VM  $u_z$ .

---

Entrada:  $u_z, s_z, C\bar{e}_z, C\bar{d}_z, D$   
Saída:  $\{d_i, d_j\}$

```

1 início
2    $d_i \leftarrow \underset{d_i \in D}{\operatorname{argmin}} C\bar{e}_z;$ 
3   se ( $\operatorname{contains}(d_i, s_z) == 1$ ) então
4      $t_i \leftarrow 0;$ 
5    $d_j \leftarrow \underset{d_j \in D}{\operatorname{argmin}} C\bar{d}_z;$ 

```

Resultado:  $\{d_i, d_j\}$

---

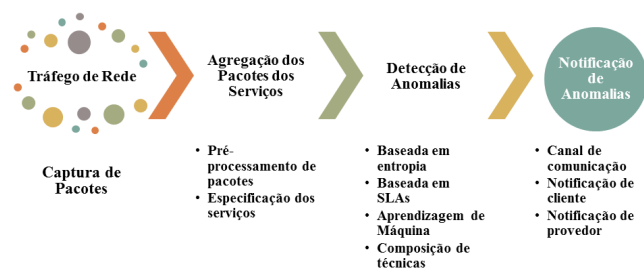
O retorno do processo de escalonamento é um par ordenado, que compreende o centro de dados que irá hospedar a máquina virtual e o centro de dados de onde a fonte de dados será recebida para o processamento da consulta. O centro de dados escolhido para executar a VM é o que minimiza o custo de execução; neste caso, o centro de dados  $d_i$  (Linha 2). Se a fonte de dados requerida para o processamento da VM ( $s_z$ ) estiver replicada no centro de dados escolhido para a execução da VM ( $d_i$ ), então o custo de recebimento dos dados de entrada não se faz necessário. Portanto, esse custo é alterado para zero (Linha 4). Quando não há uma fonte de dados, o custo de transferência é infinito. A cópia da fonte de dados será recebida do centro de dados que minimiza o custo de transferência; neste caso, o centro de dados  $d_j$  (Linha 6). Observa-se que  $i$  pode ser igual a  $j$ , pois há a possibilidade de um mesmo centro de dados minimizar ambos os custos.

### 3.3 Detecção de Anomalias de Tráfego

Com base no estudo do comportamento padrão do sistema, pode-se identificar desvios na operação normal do sistema. Estes podem indicar que aplicações estão gerando tráfego além do esperado; por exemplo, um serviço com erro está enviando pacotes de difusão (*broadcast*) na rede, ou que há ausência de tráfego às 10 horas para um serviço que sempre está operando em horário comercial.

**Anomalia de tráfego** é um termo bastante genérico que abrange ataques, tráfego indesejado devido a aplicações em erro, perda de pacotes e de injeção de tráfego indesejado de aplicativos não permitidos, entre outros comportamentos anormais que possam estar presentes no tráfego.

A qualidade da comunicação de rede entre provedor e cliente afeta significativamente o desempenho da maioria das aplicações que executam na nuvem (Shetty, 2013). A análise de tráfego não é uma tarefa trivial,



**Figura 3:** Processo de detecção de anomalias.

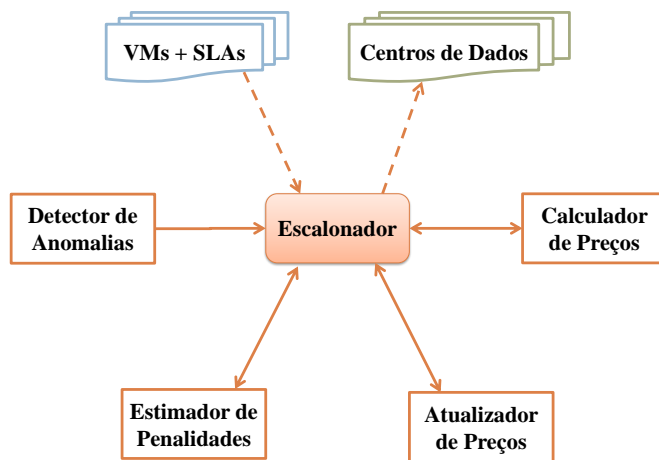
especialmente em sistemas de alta vazão. Há necessidade de mecanismos de monitoramento leves, de alto desempenho e não intrusivos (Callado et al., 2010).

O processo para detecção de anomalias de tráfego envolve a captura do tráfego da rede, passando pelo agrupamento dos pacotes por serviço, a aplicação de métodos para detecção de anomalias e a notificação. Esse processo está ilustrado na Fig. 3. A fase de detecção de anomalias pode envolver diversos tipos de técnicas, como a técnica baseada em monitoramento de SLAs, a técnica baseada em entropia, a baseada em aprendizagem de máquina ou uma técnica híbrida. A fase de notificação de anomalias ilustrada na figura envolve o cliente e a provedora. Como alternativa, pode-se utilizar um canal de publicação de informações para assinantes, também chamado de **Publicador-Assinante**, ou *Publish-Subscriber* (P-S).

## 4 Escalonamento Baseado em Custo Integrado ao Mecanismo de Detecção de Anomalias de Tráfego

Uma implementação prática do modelo de tarifação confiável em serviços de computação em nuvem será apresentada com foco no mecanismo de escalonamento do provedor de DaaS, que também é cliente de IaaS. Portanto, é possível dimensionar custos com a compra de serviços, lucro com a venda de serviços, perda financeira devido à ocorrência de anomalias de tráfego, mas também ganhos advindos do provedor de IaaS, quando essas anomalias ocorrem.

O provedor de DaaS acompanhará os resultados do processo de monitoramento de anomalias e fará o escalonamento dos recursos de modo a reduzir prejuízos financeiros. Para propor quais os centros de dados receberão VMs, ou de qual DC os dados serão recebidos, o escalonador recebe informações de quais VMs devem ser escalonadas e quais seus respectivos SLAs. O detector de anomalias provê dados sobre o estado de cada centro de dados e, caso sejam encontrados desvios, há um módulo que realiza o cálculo das penalidades devidas. Considerando que será aplicado o mecanismo de escalonamento baseado em custo, dois outros módulos ainda são empregados no escalonamento: o calculador de preços e o atualizador de preços. O calculador de preços verifica o preço para executar cada VM, de modo



**Figura 4:** Interação entre o escalonador e os demais módulos do modelo.

que o escalonador priorizará os centros de dados de menor custo. Quando ocorrem anomalias, o centro de dados em falha tem seu preço atualizado de modo a envolver os custos das anomalias. Esse processo está representado na Fig. 4.

O Algoritmo 2 apresenta a estratégia de escalonamento baseado em custo integrado ao mecanismo de detecção de anomalias para alocar a máquina virtual  $u_z$ , a qual utiliza a fonte de dados  $s_z$  como entrada para o processamento de uma consulta. O algoritmo recebe como entrada os vetores contendo todos os custos de execução da VM ( $C\vec{e}_z$ ) e outro vetor contendo os custos de recebimento dos dados ( $C\vec{d}_z$ ).

**Algoritmo 2:** Algoritmo de escalonamento baseado em custo para a VM  $u_z$  integrado ao mecanismo de detecção de anomalias.

```

Entrada:  $u_z, s_z, C\vec{e}_z, C\vec{d}_z, D$ 
Saída:  $\{d_i, d_j\}$ 
1 início
2 /* método chamado quando ocorrem anomalias */
3 atualize_anomalias_encontradas()
4 /* adiciona custos de anomalias para execução de VMs */
5 atualize_custos_de_execucao()
6 /* adiciona custos de anomalias para fontes de dados */
7 atualize_custos_de_fontes_de_dados()
8  $d_i \leftarrow \operatorname{argmin}_{d_i \in D} C\vec{e}_z;$ 
9 se ( $\operatorname{contains}(d_i, s_z) == 1$ ) então
10    $t_i \leftarrow 0;$ 
11  $d_j \leftarrow \operatorname{argmin}_{d_j \in D} C\vec{d}_z;$ 
Resultado:  $\{d_i, d_j\}$ 
  
```

Destaca-se que antes do escalonador realizar a escolha dos centros de dados, ocorrem chamadas a métodos que atualizam a incidência de anomalias e os custos. Na Linha 2 é chamado um método que verifica o detector de anomalias. Há ainda a possibilidade desse método

ser chamado por um módulo externo assim que uma anomalia for detectada. Se forem detectadas anomalias em algum centro de dados, o custo dessas anomalias é calculado. O método chamado na Linha 3 acrescenta o custo estimado das anomalias de tráfego aos custos de execução dos centros de dados que estão em falha.

O método chamado na Linha 4 atualiza os custos de anomalias para o acesso a fontes de dados nos centros de dados em falha. Suponha que o centro de dados  $d_i$  apresente um percentual de anomalias de tráfego para a meta de disponibilidade. Se o custo da perda causada pelas anomalias for inferior aos custos de execução nos demais centros de dados, então mesmo com um acréscimo de custo, a VM  $u_z$  será executada no centro de dados  $d_i$ . Após a atualização dos custos das possíveis anomalias, o mecanismo de escalonamento é o mesmo que foi apresentado no Algoritmo 1, o qual retorna os centros de dados que minimizam os custos de execução e de acesso à fonte de dados, respectivamente.

## 5 Validação

Destaca-se que há o conceito de custo e de preço. O preço é o valor que é cobrado ao cliente por um serviço e o custo é o valor que o provimento do serviço acarreta ao provedor. Em sistemas de computação em nuvem esses valores podem ser alternados para o mesmo ator (provedor ou cliente). Para simplificar, será utilizado apenas o termo **custo** neste trabalho, não tratando de margens de lucro e outras questões.

Um modelo para tarifação confiável em sistemas de computação em nuvem é uma solução que envolve diversos níveis organizacionais, podendo ser avaliado sobre diversas perspectivas.

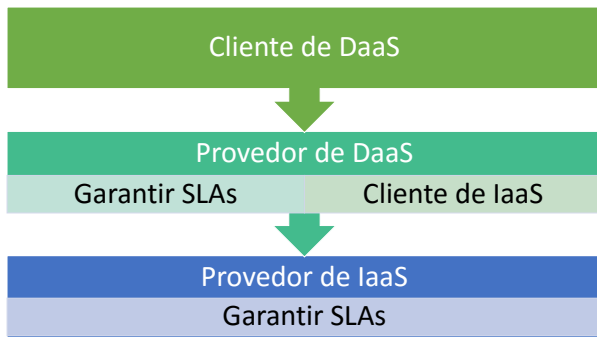
### 5.1 Modelo do Sistema

Neste caso, o provedor de DaaS contrata um provedor para fornecer infraestrutura e vende o serviço de DaaS a outros clientes, como ilustrado na Fig. 5. As anomalias detectadas serão penalizadas de acordo com o prejuízo financeiro que acarretam para o modelo de negócio. Dessa forma, é realizado o cálculo do valor das anomalias no tempo e esse valor é considerado um prejuízo financeiro, retornando ao cliente como crédito.

Para o caso do provimento de DaaS, há duas possibilidades. O cliente pode pagar de modo variável, se conseguir a oportunidade de executar seus serviços nos centros de dados de menor custo, ou pode pagar um valor fixo referente aos centros de dados de maior custo e a diferença entre esses custos é revertida ao provedor. Essas são decisões de negócio. Este trabalho flexibiliza esse tipo de decisão, concentrando-se em reduzir os custos e penalidades em sua concepção primária.

#### 5.1.1 Estimativa de Variáveis do Modelo de Custo

As variáveis que compõem o modelo de custo de DaaS são difíceis de serem obtidas. Para avaliar o modelo, realizou-se um experimento com uma aplicação de DaaS que permitiu que essas variáveis pudessem ser estimadas. O experimento, com duração de 24 horas,



**Figura 5:** Exemplo de contratação de serviços de computação em nuvem em diferentes níveis de abstração.

**Tabela 1:** SLAs Contratados.

VM	Restrição (%)	Tipo	Custo Adicional (%)	Máx. Tempo Indispon. (s/h)
1	99,9%	Golden	20%	3,6
2	99,8059%	Silver	10%	6,98
3	99,589%	Bronze	5%	14,796

consistiu em uma aplicação de *crawling* (rastreamento) da *Web* utilizando o modelo de programação *MapReduce* distribuído em três centros de dados (micronuvens) localizados em 3 cidades. Três tipos de trabalhos (consultas) são executados, tendo-se que cada trabalho é dividido em tarefas com duração de 1 hora, as quais são escalonadas como VMs independentes.

Para as VMs da consulta 1, simplesmente chamadas de VM 1, o tipo de SLA estabelecido foi chamado de *Golden*. O tipo de SLA das VMs da consulta 2 é chamado de *Silver*, e da consulta 3 é denominado *Bronze*. Na [Tabela 1](#) estão reunidas as informações sobre os tipos de SLAs e as VMs correspondentes.

As variáveis  $q$ ,  $k$ ,  $R$  e  $T$  das VMs foram estimadas utilizando os dados obtidos pelo processamento do tráfego. Essas variáveis foram contabilizadas para o tempo de execução total do experimento (de aproximadamente 24,79 horas) e também a cada hora de execução. O volume foi medido em GiB (gibibytes),  $2^{30}$  bytes = 1.024 mebibytes, mas ao longo deste trabalho essa medida será tratada simplesmente por GB. As [Figs. 6a a 6d](#) apresentam as estimativas, calculadas por hora, para as variáveis  $q$ ,  $k$ ,  $R$  e  $T$  de cada VM, respectivamente.

### 5.1.2 Injeção de Anomalias e Definição de Penalidades

Neste modelo, uma anomalia é entendida como o período de **1 segundo** considerado anômalo pelo mecanismo de detecção, que é um tempo curto para disponibilidade no nível de aplicação e alto para o nível de rede. O modelo de penalidades para as anomalias detectadas deve ser definido com base no modelo de negócio dos serviços e quanto realmente custa aquele tráfego.

Neste estudo de caso, considerou-se que as penalidades para DaaS seguem o modelo proposto por [Xiong et al. \(2011\)](#), o qual atribui uma penalidade fixa a cada

**Tabela 2:** Modelo de penalidades para anomalias detectadas.

Tipo	Penalidade para DaaS (€/s)
Golden	0,1
Silver	0,05
Bronze	0,025

anomalia detectada. Esse valor fixo é uma função do tipo de SLA da VM que está sujeita à anomalia. Se houver uma anomalia em um centro de dados que deve executar a VM  $u_z$ , o valor de penalidade correspondente ao SLA da VM é  $p_{SLA}(z)$ , sendo este acrescido ao custo de execução daquele centro de dados ( $v_i = v_i + p_{SLA}(z)$ ). Se ocorrer uma anomalia no centro de dados de onde a fonte de dados seria recuperada, então o custo de recuperação da fonte de dados também é acrescido com o valor de penalidade correspondente ao SLA da VM ( $t_j = t_j + p_{SLA}(z)$ ). Os valores das penalidades de cada anomalia estão apresentados na [Tabela 2](#).

Sem perda de generalidade, assume-se que o serviço de IaaS foi contratado com um único SLA, o qual deve atender ao SLA do tipo *Gold* para simplificar o modelo e não haver necessidade do provedor de DaaS renegociar o preço dos serviços a cada vez que vender um serviço de DaaS. As penalidades de IaaS são definidas conforme um valor proporcional ao tempo em que o serviço estiver indisponível.

A probabilidade de violação dos SLAs foi definida com base no estudo realizado por [Goudarzi et al. \(2012\)](#), que tratou as violações de SLA como uma função do tipo de SLA contratado e não como um valor fixo, que traria prejuízos maiores aos SLAs mais restritivos. O percentual de violação de SLA é 50% a mais do que o aceitado para cada tipo de SLA contratado.

## 5.2 Resultados Obtidos

Para o cálculo das penalidades, apenas as anomalias que excedem o número máximo aceitável são contabilizadas. As anomalias foram injetadas no centro de dados de menor custo. Neste estudo, esse centro de dados é o  $d_1$ . [Tabela 3](#) apresenta o número de anomalias que foi injetado no centro de dados  $d_1$ , o número máximo aceitável de anomalias para cada VM e o número de anomalias excedente.

Os custos do processamento das consultas nas VMs dos dois mecanismos de escalonamento ao longo de cada hora são apresentados na [Fig. 7](#). Observa-se que o escalonamento integrado apresenta casos em que o valor excede o praticado pelo escalonamento padrão. Isso ocorre porque o risco de violação de SLAs ao executar uma VM neste centro de dados aumenta com a incidência de anomalias, o que provoca aumento do custo do DC em falha e, portanto, ele pode não se tornar elegível pelo mecanismo de escalonamento. Os custos de IaaS pagos pelo provedor de DaaS são determinados pelo número de horas de utilização da infraestrutura; logo, eles são os mesmos independentemente do mecanismo de escalonamento das VMs. Na [Tabela 4](#) encontram-se os custos totais de processamento de cada VM.

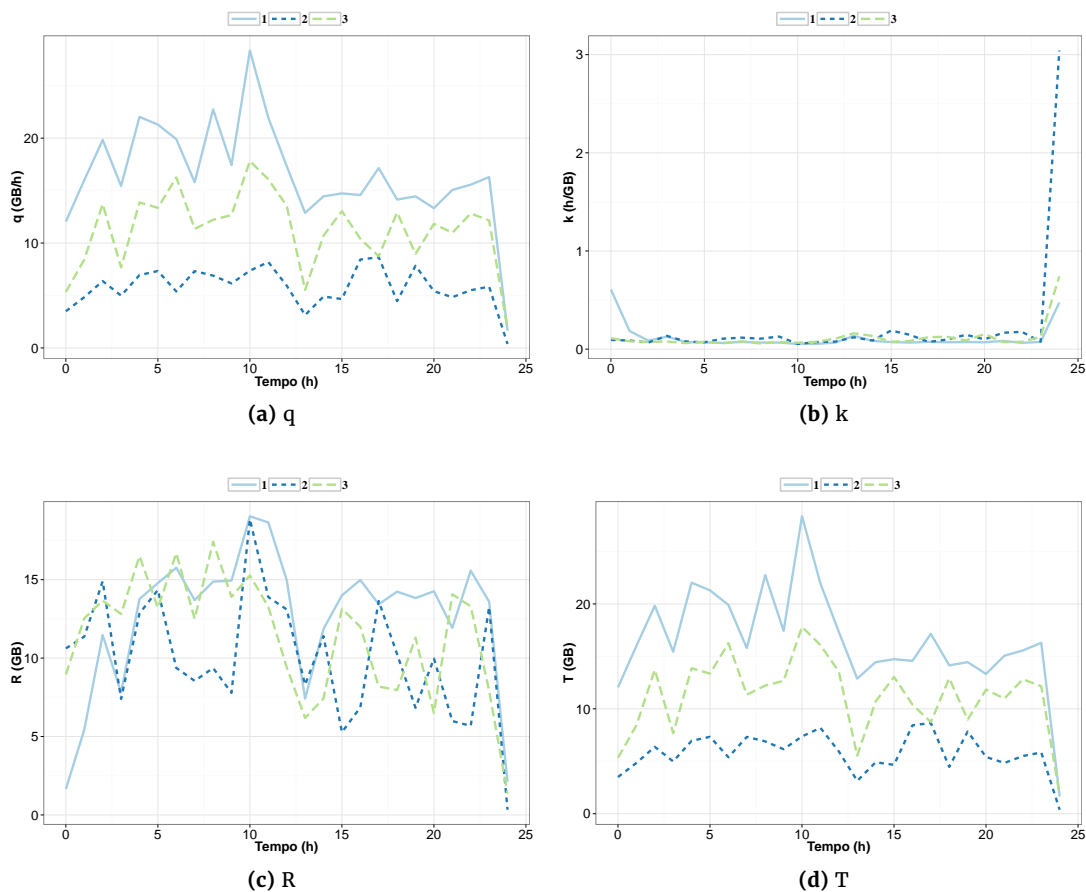


Figura 6: Estimativa das variáveis de custo  $q$  e  $k$  e das variáveis de volume  $R$  e  $T$  calculadas por hora para cada VM.

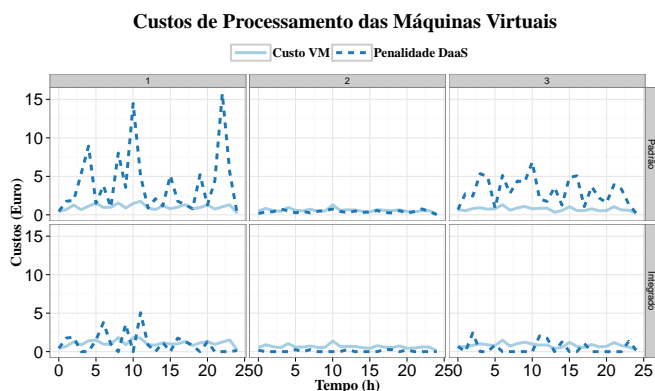


Figura 7: Custos das 3 VMs utilizando os dois mecanismos de escalonamento.

Nem sempre é vantajoso financeiramente escolher o centro de dados com menor custo. Quando o valor da penalidade no centro de dados de menor custo é superior à diferença de custo entre esse centro de dados e outro, então a alocação da VM é feita em um centro de dados que não minimiza os custos.

A perda acumulada para DaaS e IaaS é apresentada na Fig. 8. Por meio desses gráficos é possível notar que o prejuízo causado com anomalias de tráfego é reduzido significativamente ao longo de todo o processo de execução, quando aplicado o mecanismo de escalonamento integrado.

O escalonamento integrado também reduz perdas no provedor de IaaS, pois a indisponibilidade com anomalias independe do nível de abstração do serviço. Logo, quando o serviço é mudado para um centro de dados livre de falhas, a perda no nível de IaaS também é reduzida.

O escalonamento integrado reduziu as perdas a aproximadamente 26%, 16,8% e 15,3% do valor que foi obtido pelo escalonamento padrão para as VMs 1, 2 e 3, respectivamente. Houve uma redução entre 2,34% a 15% para a perda no serviço de IaaS também, pois as VMs foram alocadas a centros de dados livres de falhas de rede.



**Tabela 3:** Número de anomalias injetadas, número máximo aceitável de anomalias e número de anomalias excedentes.

Tempo	VM 1 (Máx. 3,6)		VM 2 (Máx. 6,98)		VM 3 (Máx. 14,796)	
	#Anom.	#Anom. Excedentes	#Anom.	# Anom. Excedentes	#Anom.	#Anom. Excedentes
0	3	0	7	0,02	9	0
1	5	1,4	10	3,02	21	6,204
2	4	0,4	10	3,02	19	4,204
3	8	4,4	17	10,02	27	12,204
4	8	4,4	12	5,02	24	9,204
5	2	0	7	0,02	15	0,204
6	5	1,4	11	4,02	23	8,204
7	3	0	6	0	21	6,204
8	7	3,4	11	4,02	22	7,204
9	5	1,4	13	6,02	23	8,204
10	9	5,4	16	9,02	29	14,204
11	5	1,4	10	3,02	18	3,204
12	2	0	8	1,02	18	3,204
13	5	1,4	13	6,02	32	17,204
14	2	0	9	2,02	18	3,204
15	6	2,4	10	3,02	25	10,204
16	4	0,4	12	5,02	28	13,204
17	3	0	7	0,02	17	2,204
18	3	0	8	1,02	27	12,204
19	6	2,4	12	5,02	20	5,204
20	2	0	5	0	20	5,204
21	6	2,4	11	4,02	22	7,204
22	11	7,4	17	10,02	22	7,204
23	6	2,4	11	4,02	18	3,204
24	3	0	6	0	13	0
<b>Total:</b>	<b>123</b>	<b>42,4</b>	<b>259</b>	<b>88,44</b>	<b>531</b>	<b>168,692</b>

**Tabela 4:** Custos totais das VMs para DaaS utilizando o escalonamento integrado e o escalonamento padrão,

VM	Escalonador	Custo Total	Custo Médio por Hora	$\sigma$ por Hora
1	Padrão	25,493689	1,019747542	0,364567546
1	Integrado	27,744306	1,109772238	0,401932738
2	Padrão	14,551529	0,58206118	0,218658274
2	Integrado	15,863584	0,634543367	0,23292921
3	Padrão	18,29941	0,731976402	0,249611926
3	Integrado	20,428799	0,81715197	0,284202724

Os custos de IaaS foram definidos de forma fixa, portanto já era esperado que não houvesse modificação no valor dos custos de utilização dos serviços. Os custos da utilização dos serviços de DaaS são mais altos pois, em alguns casos, as VMs não foram alocadas aos centros de dados de menor custo. Contudo, mesmo com essa mudança, o acréscimo de custo ficou em torno de 10%. A Tabela 5 apresenta um sumário com as perdas e os ganhos relativos entre o valor obtido pelo escalonamento baseado em custo integrado ao mecanismo de anomalias de tráfego e o escalonamento baseado em custo padrão.

### 5.3 Reprodutibilidade

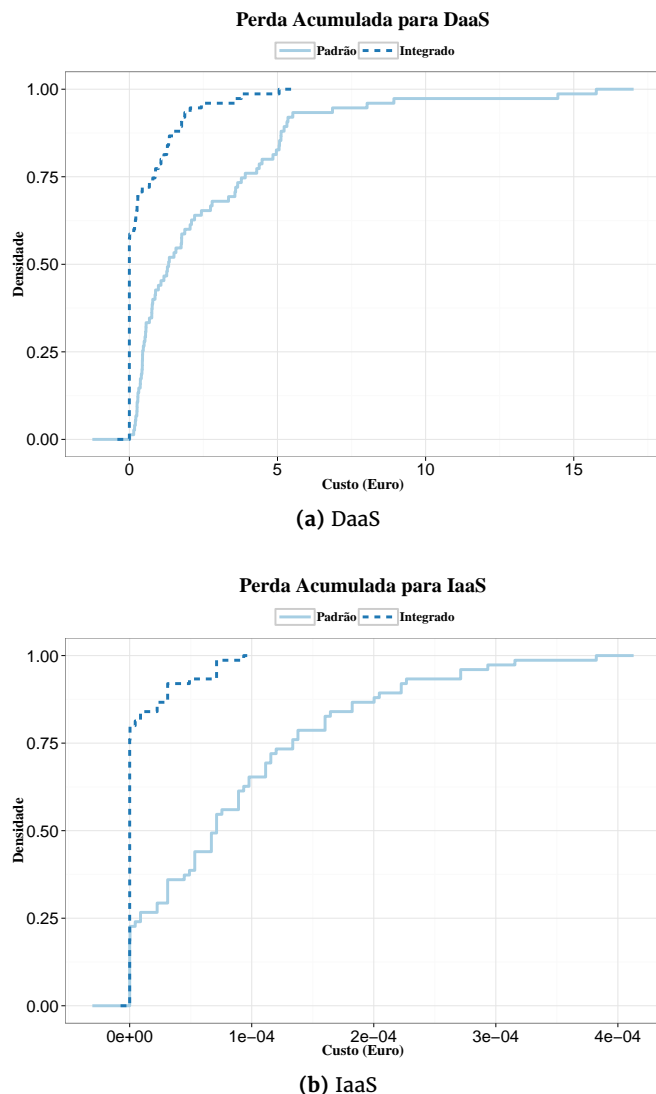
Os códigos fontes, materiais e informações suplementares referentes a esta pesquisa estão disponíveis em um repositório público no seguinte endereço: <https://anacristina@bitbucket.org/anacristina/thesis.git>.

**Tabela 5:** Perdas e custos relativos entre os valores obtidos pelo escalonamento integrado e o escalonamento padrão baseado em custo.

VM	Integrado / Padrão			
	Perda DaaS	Perda IaaS	Custo IaaS	Custo DaaS
1	0,259935866	0,1509434	1	1,0882814
2	0,168254209	2,34E-02	1	1,0901661
3	0,152637409	0,1151685	1	1,1163638

## 6 Trabalhos Relacionados

Os trabalhos relacionados geralmente abordam estratégias de monitoramento de SLAs, realocação de recursos com base nos SLAs e modelos de precificação de serviços de computação em nuvem em diferentes níveis. No entanto, não apresentam um estudo consolidando a definição do modelo de precificação de serviços, a modelagem de máquinas virtuais e os custos dos servi-



**Figura 8:** Perda acumulada para os serviços de DaaS e IaaS em cada VM.

ços e das perdas com violações nos acordos de forma integrada, como tratado neste artigo.

Diversos trabalhos tratam de realocação de recursos com base no monitoramento e na conformidade dos objetivos para os níveis de serviço em computação em nuvem, como os de [Pranav et al. \(2020\)](#), [Hussain et al. \(2020\)](#), [Pietri and Sakellariou \(2016\)](#), [Zaman and Grosu \(2013\)](#), [Justafort and Pierre \(2012\)](#), [Patel and Sarje \(2012\)](#) e [Kantarci et al. \(2012\)](#). As abordagens apresentam diferentes objetivos, com foco em redução de perdas financeiras e leilões, mas também em eficiência energética, tempo de execução, roteamento, entre outros.

Alguns trabalhos abordam preços em diferentes níveis de serviços de computação em nuvem. [Dimitri \(2020\)](#) abordou um modelo de precificação para IaaS. [Mahmud et al. \(2020\)](#) desenvolveram um sistema integrado de qualidade de serviço, lucro e alocação de recur-

sos, contudo não aplicado diretamente a computação em nuvem, mas a *Fog-Cloud*, utilizando Programação Linear Inteira.

[Kohne et al. \(2017\)](#) trabalharam em um simulador chamado *FederatedCloudSim* para simulação e avaliação de nuvens federadas, incluindo simulação de SLAs, estratégias de escalonamento e intermediação em vários níveis, carregamento de cargas de trabalho de nuvens reais e avaliação financeira.

[Aldossary et al. \(2019\)](#) apresentam uma arquitetura para sistemas de computação em nuvem com foco na avaliação da eficiência energética e computacional e um arcabouço para estimação de custo com base na utilização dos recursos. [Kantarci et al. \(2012\)](#) propuseram também um esquema para alocação de VMs com foco em minimizar o consumo de energia. Os centros de dados estão entre os principais contribuintes para a emissão do gás GHC (*Green-House Gas*) dentre os serviços de TI. Os centros de dados mais eficientes energeticamente são escolhidos para hospedar as VMs. Essa abordagem aplica Programação Linear Inteira Mista.

[Zhu et al. \(2018\)](#) desenvolveram um método de detecção de anomalias baseado em modelos de *deep learning*, mais especificamente os modelos de rede neural *feed-forward* (FNN) e o modelo de rede neural convolucional (CNN). [Al-Musawi et al. \(2017\)](#) classificaram anomalias de tráfego baseadas nos recursos do protocolo de roteamento BGP (*Border Gateway Protocol*). [Li et al. \(2012\)](#) desenvolveram um algoritmo adaptativo para encontrar o melhor plano de alocação com objetivo de maximizar a disponibilidade de recursos de IaaS, evitando superutilização de recursos, considerando que essas práticas são as principais responsáveis para o aumento do lucro.

[Taghavi et al. \(2020\)](#) propuseram um modelo para monitoramento da qualidade de serviço em sistemas de computação em nuvem baseados em *blockchain*, que têm barreiras de proteção para acesso a dados da rede externa, por meio da implementação de uma entidade confiável chamada de oráculo.

[Islam et al. \(2018\)](#) propuseram sistemas de detecção de anomalias usando uma combinação booleana baseada em poda ponderada para selecionar e combinar detectores de anomalias precisos e diversos. [Garg et al. \(2017\)](#) propuseram um esquema de detecção de anomalias baseado em grupos (*clusters*).

[Schwarzkopf et al. \(2013\)](#) propuseram uma nova arquitetura de escalonamento chamada de escalonador *Omega*, o qual emprega compartilhamento de estado e controle de concorrência otimista e livre de retenção para proporcionar extensibilidade e escalabilidade.

## 7 Conclusões

O modelo de tarifação confiável para serviços de computação em nuvem consiste em disponibilizar e utilizar as informações sobre anomalias de tráfego em serviços de computação em nuvem para a parte provedora e o cliente. Ele foi validado em um mecanismo de escalonamento baseado em custo, tendo como estudo de caso um provedor de DaaS que também era um cliente de IaaS.

O modelo demonstrou melhorias significativas na redução de custos e na redução de perdas financeiras provenientes de violações de acordos de nível de serviço. A integração do escalonamento baseado em custo ao processo de detecção de anomalias de tráfego reduziu as perdas financeiras para valores entre 15% e 26% dos valores obtidos para as perdas nos métodos convencionais.

Quanto a balanceamento de carga, o esquema de custos pode ser ajustado de modo que DCs subutilizados tenham seus custos diminuídos e os com maior utilização se tornem mais altos. No entanto, essa questão está fora do escopo deste trabalho.

## Agradecimentos

Este trabalho foi desenvolvido no Laboratório de Tecnologia da Informação (LaTIn/IFPB) com equipamentos adquiridos por meio de editais de pesquisa com recursos do IFPB Campus Campina Grande e do Projeto Jit Clouds financiado pelo Centro de Pesquisa e Desenvolvimento em Tecnologias Digitais para Informação e Comunicação (CTIC) em parceria com a Rede Nacional de Ensino e Pesquisa (RNP).

## Referências

- Al-Musawi, B., Branch, P. and Armitage, G. (2017). Bgp anomaly detection techniques: A survey, *IEEE Communications Surveys Tutorials* **19**(1): 377–396. <https://doi.org/10.1109/COMST.2016.2622240>.
- Aldossary, M., Djemame, K., Alzamil, I., Kostopoulos, A., Dimakis, A. and Agiatzidou, E. (2019). Energy-aware cost prediction and pricing of virtual machines in cloud computing environments, *Future Generation Computer Systems* **93**: 442 – 459. <https://doi.org/10.1016/j.future.2018.10.027>.
- Burrows, M. (2006). The chubby lock service for loosely-coupled distributed systems, *Proceedings of the 7th Symposium on Operating Systems Design and Implementation, OSDI '06*, USENIX Association, Berkeley, CA, USA, pp. 335–350.
- Callado, A., Kelner, J., Sadok, D., Kamienski, C. A. and Fernandes, S. (2010). Better network traffic identification through the independent combination of techniques, *J. Netw. Comput. Appl.* **33**(4): 433–446. <https://doi.org/10.1016/j.jnca.2010.02.002>.
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A. and Gruber, R. E. (2006). Bigtable: A distributed storage system for structured data, *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation - Volume 7, OSDI '06*, USENIX Association, Berkeley, CA, USA, pp. 15–15. <https://doi.org/10.1145/1365815.1365816>.
- Dean, J. and Ghemawat, S. (2008). MapReduce : Simplified Data Processing on Large Clusters, *Communications of the ACM* **51**(1): 1–13. <https://doi.org/10.1145/1327452.1327492>.
- Dimitri, N. (2020). *Journal of Cloud Computing: Advances, Systems and Applications* **9**(1): 9–14. <https://doi.org/10.1186/s13677-020-00161-2>.
- Garg, S., Singh, A., Batra, S., Kumar, N. and Obaidat, M. S. (2017). Enclass: Ensemble-based classification model for network anomaly detection in massive datasets, *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–7. <https://doi.org/10.1109/GLOCOM.2017.8255025>.
- Goudarzi, H., Ghasemazar, M. and Pedram, M. (2012). SLA-based Optimization of Power and Migration Cost in Cloud Computing, *Ieee*, pp. 172–179. <https://doi.org/10.1109/CCGrid.2012.112>.
- Hurwitz, J., Nugent, A., Halper, F. and Kaufman, M. (2013). *Big Data For Dummies*, 1st edn, For Dummies.
- Hussain, W., Sohaib, O., Naderpour, M. and Gao, H. (2020). Cloud marginal resource allocation: A decision support model, *Mobile Networks and Applications* (25): 1418–1433. <https://doi.org/10.1007/s11036-019-01457-7>.
- Islam, M. S., Khreich, W. and Hamou-Lhadj, A. (2018). Anomaly detection techniques based on kappa-pruned ensembles, *IEEE Transactions on Reliability* **67**(1): 212–229. <https://doi.org/10.1109/TR.2017.2787138>.
- Justafort, V. and Pierre, S. (2012). Performance-aware virtual machine allocation approach in an intercloud environment, *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, pp. 1–4. <https://doi.org/10.1109/CCECE.2012.6334927>.
- Kantarci, B., Foschini, L., Corradi, A. and Mouftah, H. (2012). Inter-and-intra data center vm-placement for energy-efficient large-scale cloud systems, *Globecom Workshops (GC Wkshps), 2012 IEEE*, pp. 708–713. <https://doi.org/10.1109/GLOCOMW.2012.6477661>.
- Khatri, V. and Brown, C. V. (2010). Designing data governance, *Commun. ACM* **53**(1): 148–152. <https://doi.org/10.1145/1629175.1629210>.
- Kohne, A., Krüger, M., Pfahlberg, M., Spinczyk, O. and Nagel, L. (2017). Financial evaluation of sla-based vm scheduling strategies for cloud federations, *Proceedings of the 4th Workshop on CrossCloud Infrastructures & Platforms, Crosscloud'17*, ACM, New York, NY, USA, pp. 1:1–1:7. <https://doi.org/10.1145/3069383.3069384>.
- Li, J., Wang, Q., Jayasinghe, D., Malkowski, S., Xi-ong, P., Pu, C., Kanemasa, Y. and Kawaba, M. (2012). Profit-based experimental analysis of iaas cloud performance: Impact of software resource allocation, *Services Computing (SCC), 2012 IEEE Ninth International Conference on*, pp. 344–351. <https://doi.org/10.1109/SCC.2012.85>.

- Mahmud, R., Srirama, S. N., Ramamohanarao, K. and Buyya, R. (2020). Profit-aware application placement for integrated fog-cloud computing environments, *Journal of Parallel and Distributed Computing* **135**: 177 – 190. <https://doi.org/10.1016/j.jpdc.2019.10.001>.
- Oliveira, A. C., Chagas, H., Spohn, M., Gomes, R. and Duarte, B. J. (2014). Efficient network service level agreement monitoring for cloud computing systems, *Computers and Communication (ISCC), 2014 IEEE Symposium on*, pp. 1–6. <https://doi.org/10.1109/ISCC.2014.6912551>.
- Oliveira, A. C., Dora, P., Spohn, M. and Oliveira, K. (2015). From the dark net to the cloudy data: Cloud network governance guidelines, *XXXIV International Conference of the Chilean Society of Computer Science (SCCC)*, Santiago, Chile.
- Oliveira, A. C., Fetzer, C., Martin, A. and Spohn, M. (2015). Optimizing query prices for data-as-a-service, *Proceedings of the 2015 IEEE International Congress on Big Data, BIGDATA CONGRESS '15*, IEEE Computer Society, USA, p. 289–296. <https://doi.org/10.1109/BigDataCongress.2015.48>.
- Patel, K. S. and Sarje, A. (2012). Vm provisioning method to improve the profit and sla violation of cloud service providers, *Cloud Computing in Emerging Markets (CCEM), 2012 IEEE International Conference on*, pp. 1–5. <https://doi.org/10.1109/CCEM.2012.6354623>.
- Pietri, I. and Sakellariou, R. (2016). Mapping virtual machines onto physical machines in cloud computing: A survey, *ACM Comput. Surv.* **49**(3): 49:1–49:30. <https://doi.org/10.1145/2983575>.
- Pranav, P., Rizvi, N., Khatoon, N. and Roy, S. (2020). A review on sla-based resource provisioning in cloud, *Machine Learning and Information Processing. Advances in Intelligent Systems and Computing* **1101**: 449–463. [https://doi.org/10.1007/978-981-15-1884-3\\_41](https://doi.org/10.1007/978-981-15-1884-3_41).
- Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M. and Wilkes, J. (2013). Omega: Flexible, scalable schedulers for large compute clusters, *Proceedings of the 8th ACM European Conference on Computer Systems, EuroSys'13*, ACM, New York, NY, USA, pp. 351–364. <https://doi.org/10.1145/2465351.2465386>.
- Shetty, S. (2013). Auditing and analysis of network traffic in cloud environment, *Proceedings of the 2013 IEEE Ninth World Congress on Services, SERVICES '13*, IEEE Computer Society, Washington, DC, USA, pp. 260–267. <https://doi.org/10.1109/SERVICES.2013.42>.
- Taghavi, M., Bentahar, J., Otrók, H. and Bakhtiyari, K. (2020). A blockchain-based model for cloud service quality monitoring, *IEEE Transactions on Services Computing* **13**(2): 276–288. <https://doi.org/10.1109/TSC.2019.2948010>.
- Xiong, P., Chi, Y., Zhu, S., Tatemura, J., Pu, C. and Hacigümüş, H. (2011). Activesla: A profit-oriented admission control framework for database-as-a-service providers, *Proceedings of the 2Nd ACM Symposium on Cloud Computing, SOCC '11*, ACM, New York, NY, USA, pp. 15:1–15:14. <https://doi.org/10.1145/2038916.2038931>.
- Zaman, S. and Grosu, D. (2013). A combinatorial auction-based mechanism for dynamic vm provisioning and allocation in clouds, *Cloud Computing, IEEE Transactions on PP*(99): 1–1. <https://doi.org/10.1109/TCC.2013.9>.
- Zhu, M., Ye, K. and Xu, C.-Z. (2018). Network anomaly detection and identification based on deep learning methods, in M. Luo and L.-J. Zhang (eds), *Cloud Computing – CLOUD 2018*, Springer International Publishing, Cham, pp. 219–234. [https://doi.org/10.1007/978-3-319-94295-7\\_15](https://doi.org/10.1007/978-3-319-94295-7_15).