Experimental Performance Evaluation Among Cloud Infrastructure Providers Under Different Load Levels

Denis B. Oliveira
Instituto Federal do Sul de Minas Gerais
IFSULDEMINAS
Poços de Caldas - MG, Brasil
denis.oliveira@alunos.ifsuldeminas.edu.br

Ricardo R. de Oliveira
Instituto Federal do Sul de Minas Gerais
IFSULDEMINAS
Poços de Caldas - MG, Brasil
ricardo.ramos@ifsuldeminas.edu.br

Ricardo F. Vilela
Universidade Federal de Santa Catarina
UFSC
Florianópolis - SC, Brasil
ricardo.vilela@ufsc.br

Victor H. S. C. Pinto
Universidade Federal do Pará
UFPA
Belém - PA, Brasil
victor.santiago@ufpa.br

Roberto N. Ungarelli
Zello
Brasília-DF, Brasil
roberto.ungarelli@gmail.com

***Abstract*—Background: Performance testing can estimate the capacity of a web service under requests. Decision-making on the ideal cloud service infrastructures for deploying specific cloud applications is challenging. Goal: Investigation on the performance of cloud infrastructure providers under different load levels. Method: An experimental study evaluated Amazon, Azure, Google, and IBM cloud infrastructure providers in terms of performance under Infrastructure as a Service (IaaS) perspective. Results: The results indicated satisfactory performance in response time and latency among most providers when subjected to up to 300 simultaneous threads. However, this effect decreases as the number of threads increases, and Apdex index value and level of user´s satisfaction are significantly reduced under different load levels, mainly for the IBM provider. Besides, the error rate rises substantially at 400 threads, with more critical results for IBM and Amazon providers. Conclusions: Providers show distinct differences across metrics, and the data collected during testings reinforced the potential particularities of cloud infrastructure services for the choice of a provider. Although preliminary, such results can support software companies in selecting a proper infrastructure provider according to particular requirements.**

***Index Terms*—Performance Testing, Infrastructure-as-a-Service, IaaS, Experimental Evaluation, Cloud Computing.**

I. Introdução

A computação em nuvem está emergindo rapidamente como uma nova geração de tecnologia da informação, na qual a computação é disponibilizada como um recurso virtualizado, acessível via rede e seu custo é medido pelo uso [13]. A principal característica da computação em nuvem é a mudança na maneira como a computação e os serviços são fornecidos ao cliente.

Atualmente muitos serviços são desenvolvidos e implantados em infraestruturas de computação em nuvem. Essa prática oferece muitas vantagens em termos de escalabilidade e elasticidade, promovendo à aplicação melhor uso de recursos disponíveis enquanto minimiza os custos empregados. Com isso em mente, espera-se que tais aplicações sejam rápidas, confiáveis e que continuem atendendo as expectativas de seus usuários mesmo em circunstâncias desfavoráveis [1].

Entre os tipos de serviços em nuvem existentes, pode-se destacar o modelo IaaS (*Infrastructure as a Service*), que consiste em um ambiente de tempo de execução para aplicações Web, onde consumidores da nuvem podem desenvolver, testar, configurar e implantar aplicativos personalizados, enquanto provedores de nuvem administram a infraestrutura subjacente [9].

Em um ambiente IaaS os recursos são alocados dinamicamente com base nas demandas, mas dependendo da configuração e/ou restrição definida nas máquinas virtuais (*Virtual machines* - VMs), os recursos podem se tornar limitados quando uma carga é atingida e, consequentemente, problemas de indisponibilidade podem ocorrer nessas situações. Desenvolvedores avaliam a qualidade do serviço sob diferentes perspectivas, tais como, tempo de resposta, a taxa de transferência de dados (*throughput*) e a disponibilidade. Assim, testes de desempenho são fundamentais para o diagnóstico, identificação de problemas e pontos de falha de um sistema [14].

Os sistemas *back-end* geralmente utilizam web services para o processamento de informações, tais como, autenticação, pagamentos, transações, entre outros. Um dos principais desafios do teste de desempenho desses sistemas é avaliar a satisfação do usuário ao utilizá-los. Por exemplo, muitos web services são criados e hospedados por terceiros, dessa forma, quando são combinados com uma série de outros web services, como os que envolvem transações financeiras e segurança, podem ser afetados quando existe uma alta carga de requisições, mesmo que

oferecidos no modelo de nuvem.

Nesse contexto, testes de desempenho podem ser empregados para estimar a capacidade de um *web service* atender requisições no ambiente de computação em nuvem. Assim, com a definição de cenários que simulam múltiplas requisições simultâneas é possível identificar se os recursos, previamente alocados no ambiente de nuvem, são suficientes para a execução de um *web service* e em quais casos, notam-se perturbações em suas funcionalidades [4].

Considerando as possíveis adversidades para o provimento de uma aplicação sob o modelo de nuvem, um dos desafios principais nesse processo é a escolha adequada de um provedor IaaS para melhor atender as necessidades de um determinado domínio de aplicação. Como exemplo, nos dias atuais há uma grande demanda por aplicações de *streaming* que necessitam de latência e tempo de resposta adequados, de outra forma, o uso dessas aplicações pode torna-se impraticável [15]. Por sua vez, aplicações de serviços financeiros demandam alta disponibilidade no provimento de seus serviços e, além disso, são intolerantes a erros. Nessa perspectiva, percebe-se a necessidade de investigar os diferentes serviços oferecidos para a escolha adequada de IaaS [8].

Neste contexto, o objetivo deste trabalho é avaliar o desempenho das principais infraestruturas de computação em nuvem, considerando cargas variáveis de requisições de 100, 200, 300, 400 e 500 usuários, sob uma aplicação web denominada SInAU (Sistema de Informação de Ambiente Universitário) que utiliza web services REST (*REpresentational State Transfer*) no atendimento de suas requisições. Na investigação deste estudo, consideram-se o tempo de resposta, latência, *throughput*, taxa de erros e o *Apdex*[1] como métricas de avaliação do teste de desempenho. A principal contribuição deste trabalho centra-se nas observações fornecidas por meio dos resultados obtidos através do estudo experimental. Esta metodologia contrasta o método convencional (*ad-hoc*) utilizado por companhias de software para justificar a escolha do provedor de IaaS mais adequado, trazendo resultados de repetitivos testes realizados sob importantes métricas de desempenho e métodos estatísticos que reforçam a validade do estudo. Embora os resultados sejam preliminares, acredita-se que os indicativos provenientes deste tipo de avaliação auxiliam a tomada de decisão quanto ao provedor de infraestrutura mais adequado com base no contexto da aplicação a ser implantada.

O restante do trabalho é estruturado da seguinte forma. Na Seção II apresenta-se uma discussão sobre os trabalhos relacionados em relação ao presente trabalho; Na Seção III apresenta-se a metodologia experimental empregada, a execução do estudo experimental e os resultados obtidos; Na Seção V discutem-se as ameaças à validade do estudo. Finalmente, na Seção VI apresentam-se as conclusões e trabalhos futuros.

[1]Apdex: https://www.apdex.org/overview.html

II. Trabalhos relacionados

Segundo Bertolino et al. [3], a maioria das pesquisas envolvendo testes na computação em nuvem concentram-se em validar atributos de desempenho. Entretanto, a avaliação de desempenho exige um planejamento rigoroso e configurações específicas para maximizar a eficácia do teste.

No trabalho de Li et al. [10] sugere-se um modelo de teste de desempenho para aplicações web em ambiente de nuvem. Nesse modelo, características ligadas às expectativas dos usuários são consideradas para o teste. A disponibilidade, consumo de CPU e taxas de transferência do ambiente de nuvem foram avaliadas em relação a dois fatores: *Optimal Users* e o *Maximum Users*. O primeiro refere-se ao número de usuários concorrentes que a aplicação sob teste suporta, sem que haja prejuízos no seu funcionamento, enquanto o segundo, o número máximo desses usuários. Ainda sobre o segundo fator, a carga máxima de acessos não pôde ser estimada de forma precisa. Com base nessa premissa, em nossa investigação atentou-se para a perspectiva de quantidade dos usuários. O Apdex, uma métrica utilizada para estimar a satisfação dos usuários de uma aplicação web, foi usado para medir a proporção de tempos de resposta satisfatórios e insatisfatórios assumindo números variáveis de usuários simultâneos.

O trabalho de Ahmad e Andras [2] propõe métricas de escalabilidade para serviços de software baseados em nuvem. Dois sistemas web, baseados em REST, foram considerados para demonstrar a aplicabilidade das métricas, já o desempenho, em relação a escalabilidade das plataformas, é medido entre a Amazon e Azure. Para a definição e execução dos testes de desempenho admitiu-se a ferramenta *JMeter* [7]. As análises experimentais mostram que as métricas permitem avaliar o impacto das demandas nos sistemas, enquanto quantificam explicitamente o desempenho da escalabilidade. De forma complementar, o presente estudo aborda outras plataformas e métricas, tais como latência, *throughput* e índice de satisfação dos usuários.

III. Estudo experimental

O planejamento deste estudo foi definido seguindo as diretrizes do modelo *Goal Question Metric (GQM)* [16] na elucidação dos objetivos e métodos avaliativos, seguindo também o processo de experimentação de software [17] para investigação de novas teorias, métodos e técnicas. A caracterização do presente estudo é formalmente sumarizada da seguinte forma:

Analisar *provedores de infraestrutura em nuvens consolidados com a* ***finalidade de*** *comparar a qualidade dos serviços prestados,* ***no que diz respeito ao*** *desempenho desses provedores IaaS* ***do ponto de vista do*** *analista de teste de desempenho* ***no contexto*** *da indústria e academia.*

Este estudo foi direcionado por questões chaves que são fortemente associadas ao teste desempenho das amostras (provedores de infraestrutura em nuvem) e características desses serviços. A seguir são descritas em detalhes as questões abordadas, hipóteses formuladas e os objetivos em relação ao estudo experimental.

QP_1**: Como cada provedor de infraestrutura em nuvem avaliado se comporta em relação a diferentes números de usuários realizando requisições simultâneas?** Pretende-se aferir o comportamento dos serviços de infraestrutura em relação ao tempo (Latência/resposta), vazão (*Throughput*) e a taxa de erros (%).

QP_2**: Como os provedores de infraestrutura em nuvem entregam satisfação ao usuário em relação ao desempenho das requisições?** Pretende-se quantificar a satisfação do usuário durante as requisições administradas pelos serviços de infraestrutura aplicando-se a métrica Apdex.

Com objetivo de direcionar as respostas para as questões de pesquisa, as seguintes hipóteses foram formuladas:

- **Hipóteses Nulas**: Não há diferença, entre os provedores de infraestrutura em nuvem, no que se diz respeito ao desempenho sob diferentes níveis de carga para QP_1 (H_0^1) e quanto ao Apdex para a QP_2 (H_0^2).

$$H_0^1 : (\mu Amazon = \mu Azure = \mu Google = \mu IBM) \quad (1)$$

$$H_0^2 : (\mu Amazon = \mu Azure = \mu Google = \mu IBM) \quad (2)$$

- **Hipóteses Alternativas**: Existe diferença, entre os provedores de infraestrutura em nuvem ($\mu PNuvem$), no que se diz respeito ao desempenho sob diferentes níveis de carga para QP_1 (H_1^1) e quanto ao Apdex para a QP_2 (H_1^2).

$$\begin{aligned} H_1^1 : (\mu Amazon \neq \mu Azure) \vee\ (\mu Amazon \neq \mu Google) \vee \\ (\mu Amazon \neq \mu IBM) \vee\ (\mu Azure \neq \mu Google) \vee \\ (\mu Azure \neq \mu IBM) \vee\ (\mu Google \neq \mu IBM) \end{aligned} \quad (3)$$

$$\begin{aligned} H_1^2 : (\mu Amazon \neq \mu Azure) \vee\ (\mu Amazon \neq \mu Google) \vee \\ (\mu Amazon \neq \mu IBM) \vee\ (\mu Azure \neq \mu Google) \vee \\ (\mu Azure \neq \mu IBM) \vee (\mu Google \neq \mu IBM) \end{aligned} \quad (4)$$

A. Variáveis e Métricas

1) Variáveis independentes: Neste estudo três variáveis independentes foram manipuladas, o número de *threads* (usuários), *Ramp-up* e o tempo de execução. Na Tabela I são apresentadas as variações (manipulação) de cada variável independente.

O **Número de *Threads*** representa o número de usuários virtuais do JMeter realizando requisições simultaneamente ao provedor de infraestrutura em nuvem. A manipulação dessa variável permite identificar o comportamento do serviço de infraestrutura sob diferentes variáveis dependentes, sendo possível demonstrar o desempenho e o ponto de *stress* de uma infraestrutura.

TABLE I: Disposição das variáveis independentes

#Threads	#Ramp-up (Thread/s)	Tempo (s)
100	10	50
200	20	100
300	30	150
400	40	200
500	50	250

O período de ***Ramp-up*** determina o tempo necessário para que todas as *threads* fiquem ativas e, consequentemente, enviando requisições ao servidor. Para melhor exemplificar esse cenário, considere um total de 100 *threads*, para as quais o período de *ramp-up* é de 50 segundos. Após a primeira *thread* ser iniciada, cada *thread* seguinte é iniciada em (50/100) segundos após a *thread* anterior. A fila de requisições é realizada em um processo similar ao de um *pipeline* em ciclos de instruções do processador, no qual a *thread* com maior tempo de execução sempre possui maior número de requisições realizadas.

O **tempo de execução** determina o tempo máximo em que as *threads* poderão enviar e receber requisições. Vale ressaltar que esse tempo é compartilhado entre as *threads*, sendo assim, recomenda-se que o tempo de execução seja maior que o tempo de *ramp-up*, pois somente dessa forma há garantias de execução concorrente entre o número total de *threads* [7].

2) Variáveis dependentes: Na condução do estudo, foram medidas cinco variáveis dependentes, divididas em relação ao tempo das requisições (Latência e Tempo), vazão (*Throughput*), taxa de erros (%) e satisfação do usuário (Apdex). A variável **latência** consiste no intervalo de tempo entre o momento exato antes de enviar a requisição até logo após o recebimento da primeira resposta. Portanto, a **latência** inclui o **tempo de conexão** necessário para estabelecer uma conexão TCP entre o cliente e o servidor, o tempo necessário para processar a requisição HTTP até o recebimento da primeira resposta [11].

O **tempo de resposta** consiste no tempo gasto pela aplicação para responder à solicitação do usuário, é medido, usualmente, em segundos ou milissegundos, conforme apropriado para a aplicação [12]. A variável (ou índice) **Apdex** é uma forma numérica de avaliar a satisfação do usuário em relação ao desempenho de um serviço web. Como resposta é obtido um valor em uma escala uniforme de 0 à 1, onde 0 significa que nenhum usuário está satisfeito e 1 significa que todos os usuários estão satisfeitos.

Inicialmente define-se um valor T que representa, em segundos, o tempo aceitável para o usuário obter a resposta de sua requisição. O valor de T é arbitrário e estabelecido pela equipe de garantia de qualidade de software e depende do domínio. Para o presente estudo, o valor de T foi definido como 0,5. Com base nesse tempo é possível representar os intervalos de satisfação para o tempo de espera das requisições do sistema, sendo eles **satisfatório**,

tolerante e **frustrante**, conforme apresentado na Figura 1.

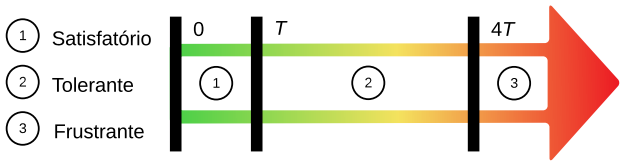

Fig. 1: Níveis de satisfação Apdex.
(Fonte: Adaptado de apdex.org)

- **Satisfatório**. O usuário consegue ser produtivo. Isso representa que a aplicação obteve um tempo de resposta menor ou igual a T.
- **Tolerante**. O usuário percebe um atraso nas respostas maior que T e menor ou igual a $4T$, porém continua o processo.
- **Frustrante**. O tempo de resposta é maior que $4T$, um valor inaceitável que pode fazer com que os usuários abandonem o processo. Este valor também pode ser referenciado como F.

Após a classificação de cada requisição, entre os níveis de satisfação, esses dados são aplicados à Equação 5 para definir um valor de Apdex em uma escala numérica.

$$Apdex_T = \frac{Satisfatórias + \frac{Tolerantes}{2}}{Total\ de\ requisições} \quad (5)$$

Onde os termos $Satisfatórias$, $Tolerantes$ e $Total\ de\ requisições$ correspondem respectivamente ao número de requisições satisfatórias, requisições tolerantes e o número total de requisições.

A variável **taxa de erro** (%) representa a porcentagem das requisições que falharam durante o teste do JMeter [7]. Por sua vez, a variável ***throughput*** ou taxa de transferência é dada pelo número de requisições por segundo em que o servidor consegue processar. Sendo assim, o tempo é calculado desde o início da primeira requisição até o final do processamento da última requisição, vale ressaltar que esse tempo também inclui quaisquer intervalos entre as requisições [7].

B. Seleção dos sujeitos

Conforme descrito por Wohlin et al. [17], a seleção dos sujeitos é um processo importante ao realizar um experimento, pois está intimamente ligada à generalização dos resultados do experimento. Portanto, para generalizar os resultados de uma população desejada, a seleção deve ser representativa para essa população. Neste estudo, a população (conjunto de provedores de infraestrutura em nuvem) foi selecionada de acordo com a representatividade de cada sujeito, em outras palavras, os serviços de infraestrutura em nuvem mais consolidados [6] e com disponibilidade de acesso gratuito temporário foram definidas para o estudo. Dito isso, os ambientes de computação em nuvem referentes à ***Amazon*** *AWS*, ***Google*** *Cloud*, ***IBM*** *Cloud* e *Microsoft* ***Azure*** foram considerados para este estudo.

C. Instrumentação e Execução

O objetivo geral da instrumentação é fornecer meios para executar o experimento e monitorá-lo, sem afetar o controle do experimento [17]. Para este estudo, inicialmente foram definidas as configurações gerais das máquinas virtuais a serem implementadas, da mesma forma, nos diferentes provedores de infraestrutura. Além disso, uma máquina virtual local também foi utilizada para realização das requisições utilizando a ferramenta *JMeter* [7]. As configurações de cada ambiente são descritas na Tabela II.

TABLE II: Configuração das máquinas virtuais.

Configuração	**Local**	**Nuvem**
Sistema Operacional	Ubuntu 18.04 LTS	Ubuntu 16.04 LTS
Localização	Brasil (Sudeste)	EUA (oeste)
vCPU	4	2
Memória (GB)	8GB	4GB
JAVA JRE	1.8	1.8
MySQL	-	5.6
Apache Tomcat	-	7.0.92
Apache JMeter	5.2.1	-

Para fins de replicação do estudo, é importante ressaltar que a máquina da *Azure* correspondente à pré configuração *BS2*, já a *Amazon* refere-se à configuração *t3.medium*, por sua vez, a máquina *IBM* corresponde à configuração *B1 2x4*, por último, a máquina da *Google* é definida de forma personalizada.

Após a preparação e configuração dos ambientes nos provedores de infraestrutura em nuvem, a aplicação web SInAU foi implantada em cada uma das infraestruturas de nuvem, conforme ilustrado na Figura 2. A SInAU foi desenvolvida com web services RESTful [5] e utiliza um banco de dados MySQL, onde as universidades, cursos, disciplinas, professores e alunos cadastrados são armazenados no seu respectivo ambiente de nuvem. A aplicação SInAU, desenvolvida em Java WEB, possui web services com 56 *endpoints* que abrangem os principais métodos HTTP (GET, POST, PUT e DELETE).

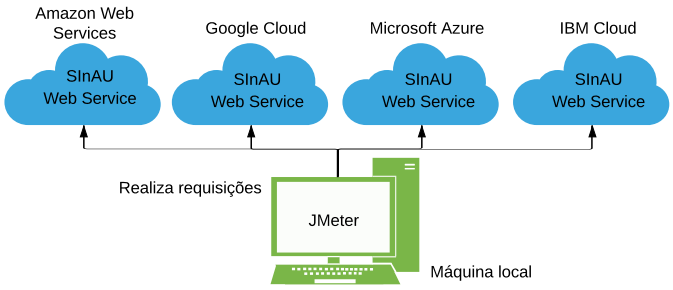

Fig. 2: Design experimental.
(Fonte: Figura do autor)

As requisições aos servidores são realizadas por um *script* de testes desenvolvido na ferramenta *JMeter*. O *script* considera diferentes quantidades de acessos simultâneos, onde cada usuário (*thread*) realiza uma

sequência de requisições lógicas, acessando *endpoints* do serviço localizado no ambiente de computação em nuvem. A ordem das requisições no *script* representa tarefas coerentes do usuário neste sistema, como por exemplo: i) listar disciplina x; e ii) listar alunos da disciplina x.

Vale ressaltar que para cada infraestrutura de nuvem e quantidade de usuários, os testes foram replicados dez vezes com o intuito de alcançar maior confiabilidade nos resultados. Por razões de espaço, as ações presentes nas rotinas de teste são apresentadas nas Tabelas 4 e 5 no relatório técnico disponível no pacote experimental[2].

D. Resultados e Análises

Os resultados provenientes das execuções foram rigorosamente submetidos a testes estatísticos de acordo com a multiplicidade das amostras e a distribuição dos dados. Neste estudo, foi empregado o teste não paramétrico *Kruskal-Wallis* juntamente com a análise de comparações múltiplas de *Dunn*. Os resultados brutos dos testes, também estão disponibilizados no pacote experimental.

As análises são projetadas para investigar como os diferentes níveis do número de usuários (*threads*) afetam o desempenho dos provedores de infraestrutura em nuvem. Em relação a primeira questão de pesquisa (QP_1), a primeira variável dependente investigada é a latência das requisições. Na Figura 3, ilustra-se os resultados obtidos sob os diferentes números de *threads*.

Conforme esperado, percebe-se uma maior consistência dos resultados de latência sob a execução simultânea de 100 *threads* para todos os provedores de infraestrutura em nuvem analisados, uma vez que, não há grande variação de tempo entre as repetições. Isso pode ser observado na distribuição dos quartis e mediana dos *boxplots* ilustrados. Apesar disso, ainda é possível observar uma diferença estatística entre as amostras para esse número de *threads*. Considerando o cenário com 200 *threads* a maioria dos serviços de infraestrutura permanecem consistentes no tempo de latência, exceto o serviço de infraestrutura da IBM, que apresenta maior variação e um tempo de latência distante das demais infraestruturas.

À medida que o número de *threads* aumenta percebe-se que os provedores de infraestrutura demonstram um declínio no desempenho, principalmente entre o intervalo de 400 e 500 *threads*. Essa elevação no valor de latência entre os serviços de infraestrutura pode ser melhor visualizada na Figura 4.

Vale ressaltar que os resultados de latência representam apenas os testes executados com sucesso, ou seja, ainda que tardias, as requisições foram completamente atendidas. Essa estratégia foi adotada para não comprometer os resultados de latência e mitigar requisições sem respostas, para as quais os valores de latência eram próximos de zero e nem sempre computados como *outliers*.

[2] *Pacote Experimental*: http://www.ricardoramosdeoliveira.com.br/wperformance/pacote-experimental.zip

Em adição as análises niveladas de acordo com o número de *threads*, optou-se por demonstrar a diferença entre os provedores de infraestrutura considerando uma análise global, ou seja, representar os diferentes números de *threads* em uma única análise. Para isso, empregou-se o cálculo de área sob a curva que utiliza a integração numérica para representar uma amostra, sob diferentes níveis, em unidades de área (u.a). A Figura 5 ilustra a análise de área sob à curva para os diferentes provedores de infraestrutura investigados.

Similar ao tempo de latência, o tempo de resposta fornece maior abrangência sobre o tempo da requisição, pois considera todo o processo de envio e recebimento de uma requisição. Os resultados obtidos em cada nível do número de *threads* para essa variável são apresentados na Figura 6.

De forma inesperada, as comparações entre os provedores de infraestrutura para o tempo de resposta não foram similares a latência obtida, conforme apresentado na Figura 7, pois percebe-se maior variação no serviço de infraestrutura da Amazon no tempo de resposta a partir de 300 *threads*. A representação em unidades de área dessa análise é ilustrada na Figura 8.

Diferentemente do tempo de resposta e latência, para taxa de transferência (*throughput*) almeja-se que um serviço de infraestrutura aumente gradativamente o *throughput* à medida que novas *threads* são iniciadas. Em geral, os provedores de infraestrutura comportam-se de forma esperada com até 300 *threads*, conforme apresentado na Figura 9.

No entanto, a partir de 400 *threads* percebe-se um declínio na taxa de transferência sugerindo que as requisições não estão sendo atendidas ou estão sendo entregues tardiamente. Apesar disso, de forma inesperada, a taxa de transferência elava-se novamente com um número de 500 *threads*, conforme apresentado nas Figura 10.

Após analisar os resultados brutos dos testes durante o estímulo de 500 *threads*, percebemos que essa elevação é dada, principalmente, pelo número de erros identificados, uma vez que, existem alguns casos em que o servidor deixou de responder as conexões e, consequentemente, a requisição foi finalizada sem que uma completa execução da rotina de teste fosse efetivada. Ainda assim, é possível identificar o ponto de *stress* dos provedores de infraestrutura quando o *throughput* atinge o primeiro pico no gráfico.

Com o objetivo de mensurar estatisticamente os serviços oferecidos pelos provedores de infraestrutura em nuvem ao longo dos diferentes números de *threads*, na Figura 11 é apresentada uma análise de área sob a curva, na qual percebe-se que os melhores resultados foram alcançados pelos provedores Azure e Google.

Assim como as demais variáveis, a taxa de erro apresenta menor incidência no intervalo de 100 à 300 *threads*, conforme ilustrado na Figura 12. A partir de 400 *threads* a taxa de erro cresce demasiadamente, de modo que percebe-se uma inconsistência no uso dos serviços de infraestrutura

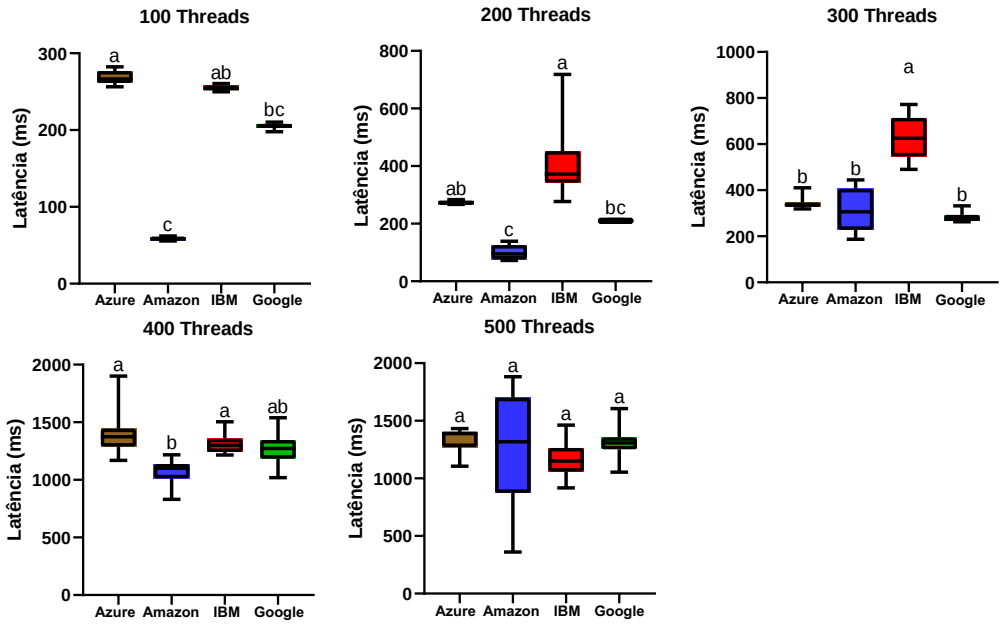

Fig. 3: Análise de latência dos serviços de infraestrutura em nuvem da Azure, Amazon, IBM e Google sob os níveis 100, 200, 300, 400 e 500 *threads*. Letras diferentes indicam uma diferença estatística (p<0.05).

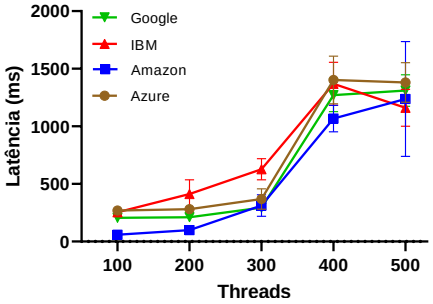

Fig. 4: Latência sob diferentes níveis de *threads*.

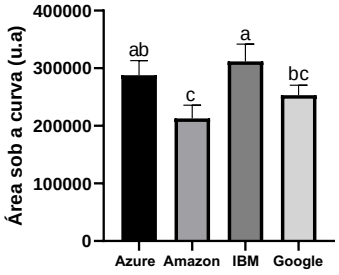

Fig. 5: Área sob a curva de latência dos provedores de infraestrutura em nuvem (p<0.05).

em geral, uma vez que os provedores apresentaram erros em cerca de 70% das requisições

Com o intuito de diferenciar as os provedores de infraestrutura em relação a porcentagem de erros, aplicou-se os devidos testes estatísticos considerando a área sob a curva, conforme apresentado na Figura 13. Nota-se que os serviços de infraestrutura da Amazon e IBM apresentam maior taxa de erro, enquanto os provedores Google e Azure apresentaram os melhores resultados. Esse resultado é fortemente atingido pela taxa de erro sob 400 *threads*, uma vez que não há diferença significativa para 500 *threads*, com 400 *threads* percebe-se que os dois melhores provedores ainda conseguem entregar a maioria de suas requisições com sucesso.

Considerando os resultados apresentados, o teste de hipótese rejeita a hipótese nula, com um nível de confiança de 95%, em todas as variáveis dependentes investigadas na QP_1. De modo geral, percebe-se que os provedores de infraestrutura apresentaram resultados satisfatórios com até 300 *threads*. Apesar disso, ainda é possível distingui-las em relação as variáveis. É importante ressaltar que esses resultados compreendem máquinas virtuais de configuração mínima dos respectivos provedores, desse modo não deve-se generalizar o número de *threads* para diferentes configurações. Ainda assim, acredita-se que a diferença entre os provedores de infraestrutura seja mantida, uma vez que o estudo foi replicado com configurações exatas em todos os serviços de infraestrutura.

Para abordar a questão QP_2, utilizou-se a variável Apdex como métrica de satisfação do usuário, conforme descrito na definição do estudo. Os resultados obtidos são apresentados na Figura 14.

Em desconformidade com as análises da QP_1, o provedor Amazon apresenta resultados superiores para maioria dos níveis de *threads*. A maioria dos provedores apresentam resultados satisfatórios com até 200 *threads* simultâneas, entretanto, o provedor IBM apresenta grande variação de Apdex em todos os níveis de *threads*, conforme apresentado na Figura 15. Esse resultado é crítico, pois ocorre em

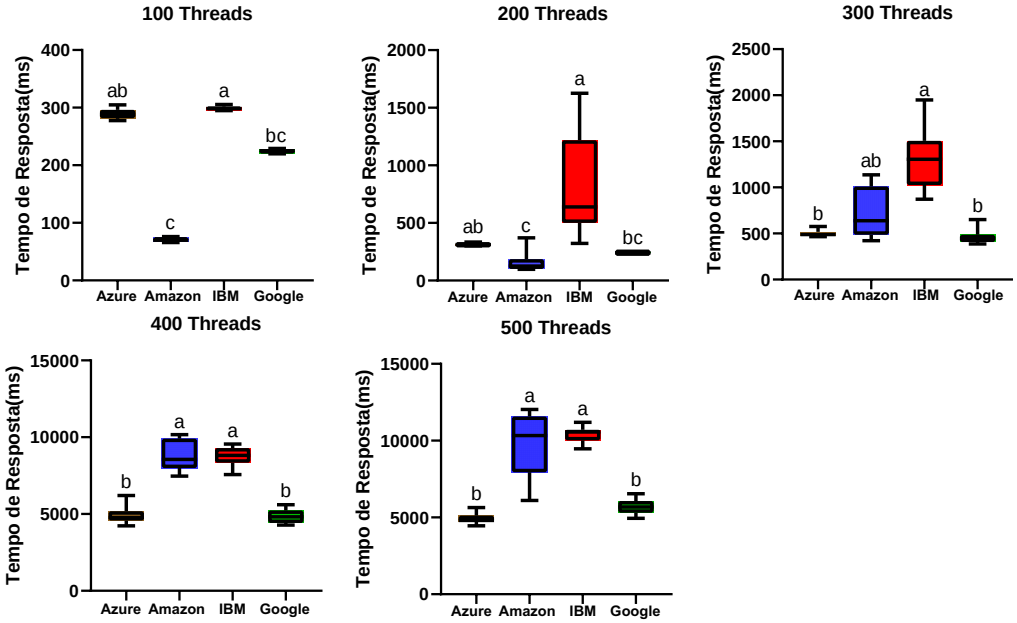

Fig. 6: Tempo de resposta dos serviços de nuvem da Azure, Amazon, IBM e Google sob os níveis 100, 200, 300, 400 e 500 *threads*. Letras diferentes indicam uma diferença estatística ($p<0.05$).

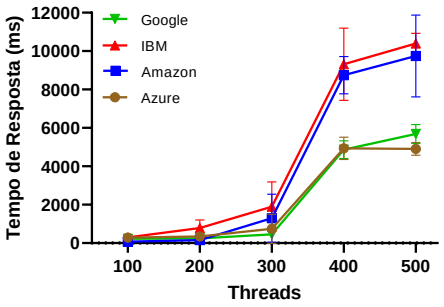

Fig. 7: Tempo de resposta sob diferentes níveis de *threads*.

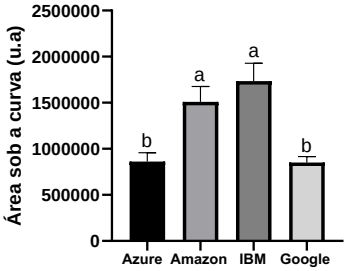

Fig. 8: Área sob a curva do tempo de resposta dos provedores de infraestrutura de nuvem ($p<0.05$).

uma amostra pequena de *threads*, ou seja, ainda com configurações mais avançadas do servidor virtual, acredita-se que esse resultado seria replicado.

Para mensurar o Apdex entre os provedores de infraestrutura em nuvem, sob os diferentes níveis de *threads*, empregou-se o teste de média *Kruskal Wallis* e comparações múltiplas de *Dunn*, a diferença entre os provedores é ilustrada na Figura 16.

Conforme observado na análise de área sob a curva percebe-se que há diferença significativa entre as amostras, desse modo rejeita-se também a hipótese nula H_0^2 sob um nível de confiança de 95%. Os resultados demonstram que os provedores Google e Azure apresentam maior entrega de satisfação ao usuário, já que suportaram melhor os níveis de carga de 300 e 400 usuários. Por outro lado, o provedor IBM apresenta maior inconsistência nos níveis de Apdex, ainda assim, não se difere do provedor Amazon.

Embora o índice de Apdex seja responsável por demonstrar a satisfação do usuário de cada provedor de infraestrutura, existem outros fatores, muitas vezes independentes, que podem influenciar nessa análise. Como exemplo, as rotas de rede tem um importante papel para esse resultado, assim sugere-se que os provedores de serviço em nuvem considerem o uso de estratégias que maximizem o uso da rede ao mesmo tempo que evitem conflitos ocasionados pelas transações de outros clientes do mesmo provedor, ou até mesmo dos demais usuários da Internet.

IV. Discussão

Conforme mencionado anteriormente, a escolha da infraestrutura de nuvem adequada para um projeto de software é um grande desafio, não apenas para novas companhias de software, mas também para empresas já consolidadas que necessitam desenvolver novos projetos ou migrar projetos antigos para infraestruturas mais robustas. Uma solução atual desenvolvida pelas empresas é o compartilhamento de experiências na adoção de novas tecnologias, contudo, esse processo é subjetivo e não fornece garantias de generalização para outros cenários.

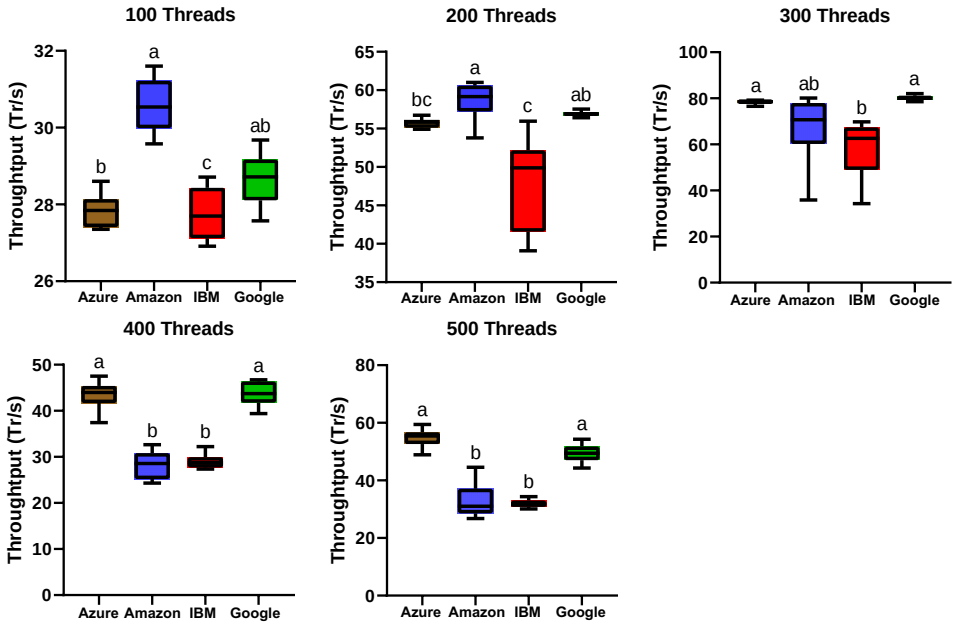

Fig. 9: *Throughput* dos serviços de infraestrutura em nuvem da Azure, Amazon, IBM e Google, dado em transações por segundo, sob os níveis 100, 200, 300, 400 e 500 *threads*. Letras diferentes indicam uma diferença estatística (p<0.05).

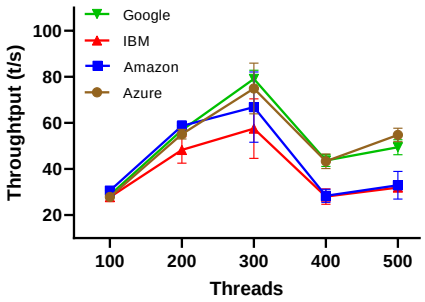

Fig. 10: *Throughput* sob diferentes níveis de *threads*.

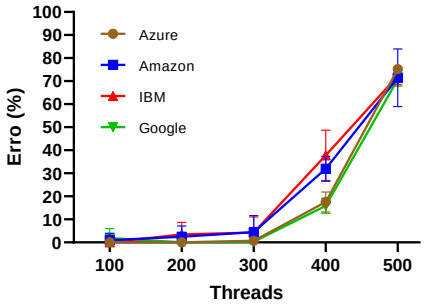

Fig. 12: Taxa de erro sob diferentes níveis de *threads*.

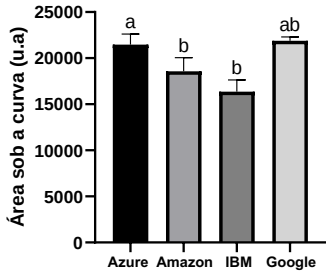

Fig. 11: Área sob a curva de *throughput* dos provedores de infraestrutura em nuvem (p<0.05).

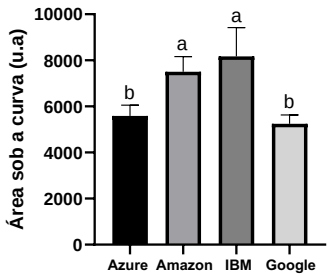

Fig. 13: Área sob a curva de erro dos provedores de infraestrutura em nuvem (p<0.05).

Dessa forma, percebe-se que as empresas perdem um tempo precioso na busca pela infraestrutura mais adequada para o projeto em desenvolvimento. O presente trabalho objetivou contribuir nesse processo de escolha por meio de uma investigação experimental que fornece resultados acerca de métricas fundamentais para o desempenho de infraestruturas em nuvem, contribuindo para que organizações que desejam migrar seus serviços para nuvem possam estabelecer uma solução ideal para um dado conjunto de exigências.

Os resultados obtidos traduzem a experiência obtida durante testes realizados em quatro serviços de infraestrutura em nuvem, os quais demonstraram comportamentos distintos de desempenho sob diferentes níveis de carga.

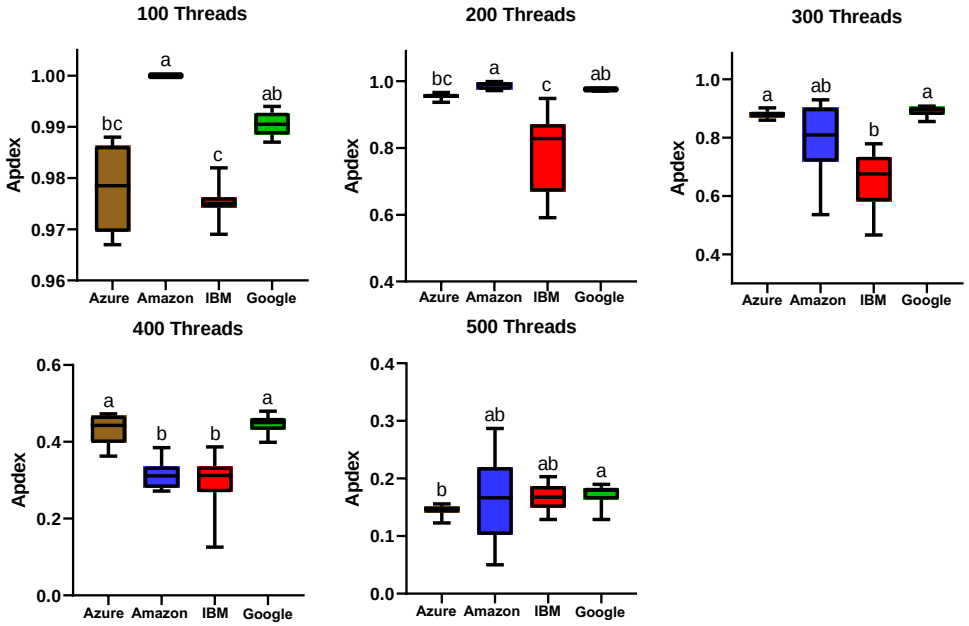

Fig. 14: Análise de Apdex dos provedores de infraestrutura em nuvem da Azure, Amazon, IBM e Google sob os níveis 100, 200, 300, 400 e 500 *threads*. Letras diferentes indicam uma diferença estatística ($p<0.05$).

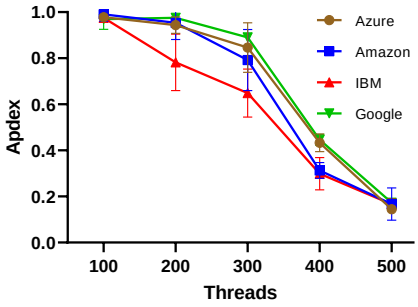

Fig. 15: Apdex sob diferentes níveis de *threads*.

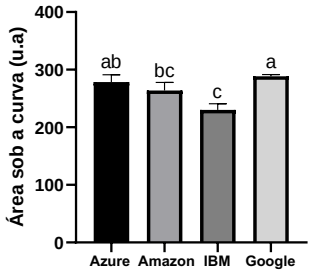

Fig. 16: Área sob a curva de Apdex dos provedores de infraestrutura em nuvem ($p<0.05$).

Diante dos resultados, notou-se que diferentes serviços de infraestrutura podem ser mais adequados de acordo com o contexto aos quais são submetidos. Ainda assim, também foi possível identificar aspectos subjetivos que podem influenciar na escolha do serviço de infraestrutura mais adequado, tais como: i) usabilidade da interface de gerenciamento; ii) plano experimental disponível; iii) planos oferecidos; iv) localização dos servidores; e v) esforço de configuração. Neste caso, sugere-se que a empresa verifique as características citadas antes da definição do serviço de IaaS.

V. Ameaças à Validade

A validade de um experimento está diretamente relacionada ao nível de confiança do mesmo [17]. Em relação à **validade externa** considera-se a escolha de uma única configuração (4GB - 2vCPU) de máquinas virtuais para cada provedor de infraestrutura uma ameaça para generalização dos resultados. Apesar disso, acredita-se que mesmo sob configurações distintas as diferenças entre os provedores de infraestrutura, sob as diferentes métricas analisadas, devem permanecer, dada a exata configuração empregada entre as máquinas dos diferentes provedores. A **validade interna** está relacionada à localização entre os provedores de infraestrutura em nuvem. Para contornar essa questão, todas as infraestruturas foram instanciadas a partir de servidores localizados no oeste dos EUA.

Em relação à **validade de construção**, não houve intenção no favorecimento de um dos provedores por parte dos pesquisadores. Todo o pacote experimental é disponibilizado, sendo passível da replicação. A **validade de conclusão** refere-se à escolha do teste estatístico. Utilizou-se o teste de normalidade das amostras *Shapiro-wilk*, no qual verificou-se que as mesmas não são paramétricas, assim optou-se pelo teste não paramétrico *Kruskal-Walis*. Dessa forma, pode-se afirmar que as recomendações estatísticas para utilização foram realizadas criteriosamente com base no número e distribuição das amostras. Por fim, o teste de

Dunn também foi utilizado como pós teste de comparações múltiplas para amostras não paramétricas.

VI. Conclusões

Neste trabalho foram apresentados os resultados de um estudo experimental controlado para avaliar e comparar o desempenho de quatro provedores de infraestrutura em nuvem. A análise empregada demonstrou relações entre métricas de desempenho e satisfação do usuário. Notou-se ainda, que há diferença entre os provedores, nos diferentes níveis de carga, em todas as métricas investigadas. Esse resultado reforça a questão de pesquisa explorada neste trabalho, ou seja, essa diferença demonstra que o desempenho dos provedores é suscetível aos estímulos impostos, assim pode-se reforçar a necessidade de uma investigação prévia para a escolha de um provedor de infraestrutura adequado.

Em geral, todas os provedores apresentaram resultados satisfatórios com até 300 *threads*, entretanto, quando submetidas ao nível de 400, ou mais *threads*, os provedores apresentaram problemas críticos. Em particular, o provedor IBM sofre demasiadamente com o aumento no número de usuários, enquanto o provedor Google oferece maior estabilidade. Observou-se por meio da avaliação uma relação direta entre o número de erros e a taxa de transferência a partir 500 usuários simultâneos. Os provedores de infraestrutura em nuvem aumentam proporcionalmente a taxa de transferência (*throughput*) à medida que a taxa de erros cresce, consequentemente as rotinas de teste não são executadas completamente, devido às requisições não atendidas e as respostas enviadas precocemente. Essa característica pode mascarar falhas nas métricas de vazão, muitas vezes empregadas em tempo real, que podem induzir a tomada de decisões incorretas pelos gestores/ferramentas de nuvem.

Considerando os aspectos levantados, como trabalhos futuros pretende-se alcançar um maior número de provedores de infraestrutura em nuvem considerando aspectos internos das máquinas virtuais, tais como: uso de memória e processamento, taxa interna de transferência de dados e ponto de *stress*. Além disso, pretende-se verificar quais tipos de erros são mais prejudiciais no correto funcionamento dessas infraestruturas.

References

[1] B. K. Aichernig, P. Bauerstätter, E. Jöbstl, S. Kann, R. Korošec, W. Krenn, C. Mateis, R. Schlick, and R. Schumi. Learning and statistical model checking of system response times. *Software Quality Journal*, 27(2):757–795, Jan. 2019.

[2] A. Al-Said Ahmad and P. Andras. Scalability analysis comparisons of cloud-based software services. *Journal of Cloud Computing*, 8(1):10, Jul 2019.

[3] A. Bertolino, G. D. Angelis, M. Gallego, B. García, F. Gortázar, F. Lonetti, and E. Marchetti. A systematic review on cloud testing. *ACM Comput. Surv.*, 52(5), Sept. 2019.

[4] Z. Cai, J. Li, and J. zhang. Research on performance optimization of web application system based on JAVA EE. *Journal of Physics: Conference Series*, 1437:012039, jan 2020.

[5] R. R. de Oliveira. *Avaliação da portabilidade entre fornecedores de teste como serviço na computação em nuvem*. PhD thesis, Universidade de São Paulo, dec 2017.

[6] L. Dignan. Top cloud providers in 2020: Aws, microsoft azure, and google cloud, hybrid, saas players — zdnet. https://www.zdnet.com/article/the-top-cloud-providers-of-2020-aws-microsoft-azure-google-cloud-hybrid-saas/. (Acessado em 13/06/2020).

[7] E. Halili. *Apache JMeter: A Practical Beginner's Guide to Automated Testing and Performance Measurement for Your Websites*. From Technologies to Solutions. Packt Publishing, 2008.

[8] J. R. M. Junior, A. C. Sacilotti, R. Sacilotti, and A. Rodrigues. Promovendo inovação com a atualização de serviços de platform as a service. *Brazilian Journal of Development*, 6(4):18143–18154, 2020.

[9] S. Lehrig, H. Eikerling, and S. Becker. Scalability, elasticity, and efficiency in cloud computing: A systematic literature review of definitions and metrics. In *11th International Conference on Quality of Software Architectures (QoSA)*, pages 83–92, 2015.

[10] H. Li, X. Li, H. Wang, J. Zhang, and Z. Jiang. Research on cloud performance testing model. In *2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE)*, pages 179–183, 2019.

[11] S. Matam and J. Jain. *JMeter Test Plan Components*, pages 35–165. Apress, Berkeley, CA, 2017.

[12] S. Matam and J. Jain. *Performance Testing Primer*, pages 3–12. Apress, Berkeley, CA, 2017.

[13] P. M. Mell and T. Grance. Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, USA, 2011.

[14] D. A. Menasce. Load testing of web sites. *IEEE Internet Computing*, 6(4):70–74, 2002.

[15] R. Ramos-Chavez, T. Karagkioules, and R. Mekuria. A scalable load generation framework for evaluation of video streaming workflows in the cloud. In *Proceedings of the 11th ACM Multimedia Systems Conference*, MMSys '20, page 255–260, New York, NY, USA, 2020. Association for Computing Machinery.

[16] R. van Solingen (Revision), V. Basili (Original article, 1994 ed.), G. Caldiera (Original article, 1994 ed.), and H. D. Rombach (Original article, 1994 ed.). *Goal Question Metric (GQM) Approach*. American Cancer Society, 2002.

[17] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén. *Experimentation in Software Engineering*. Springer Berlin Heidelberg, 2012.