UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
UNIDADE ACADÊMICA DE SISTEMAS E COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

SAMARA SONALE SANTOS SAMPAIO

ANALISANDO CONFIGURAÇÕES DE PROVISIONAMENTO AUTOMÁTICO NA NUVEM: UMA ABORDAGEM BASEADA EM SIMULAÇÃO

CAMPINA GRANDE - PB
2024

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Analisando Configurações de Provisionamento Automático na Nuvem: Uma Abordagem Baseada em Simulação

Samara Sonale Santos Sampaio

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação
Linha de Pesquisa: Sistemas de Computação

Fábio Jorge Almeida Morais
Thiago Emmanuel Pereira da Cunha Silva

Campina Grande, Paraíba, Brasil

S192a Sampaio, Samara Sonale Santos.
Analisando configurações de provisionamento automático na nuvem: uma abordagem baseada em simulação / Samara Sonale Santos Sampaio. – Campina Grande, 2024.
86 f. : il. color.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2024.
"Orientação: Prof. Dr. Fábio Jorge Almeida Morais, Prof. Dr. Thiago Emmanuel Pereira da Cunha Silva".
Referências.

1. Computação na Nuvem. 2. Auto-scaling. 3. E-commerce. 4. Infraestrutura como Serviço. 5. Cloud Services. 6. Infrastructure Utilization. 7. Políticas de Provisionamento. I. Morais, Fábio Jorge Almeida. II. Silva, Thiago Emmanuel Pereira da Cunha. III. Título.

CDU 004.738.5.057.2 (043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
POS-GRADUACAO EM CIENCIA DA COMPUTACAO
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124
Site: http://computacao.ufcg.edu.br - E-mail: secretaria-copin@computacao.ufcg.edu.br / copin@copin.ufcg.edu.br

REGISTRO DE PRESENÇA E ASSINATURAS

ATA Nº 004/2024 (DISSERTAÇÃO N° 718)

Aos quinze (15) dias do mês de fevereiro do ano de dois mil e vinte e quatro (2024), às quatorze horas (14:00), no Auditório Hattori CN, da Universidade Federal de Campina Grande - UFCG, nesta cidade, reuniu-se a Comissão Examinadora composta pelos Professores FÁBIO JORGE ALMEIDA MORAIS, Dr., UFCG, Orientador, funcionando neste ato como Presidente, THIAGO EMMANUEL PEREIRA DA CUNHA SILVA, Dr., UFCG, Orientador, REINALDO CÉZAR DE MORAIS GOMES, Dr., UFCG, PAULO DITARSO MACIEL JÚNIOR, Dr., IFPB, este com participação por videoconferência. Constituída a mencionada Comissão Examinadora pela Portaria Nº 002/2024 do Coordenador do Programa de Pós-Graduação em Ciência da Computação, tendo em vista a deliberação do Colegiado do Curso, tomada em reunião de 29 de Janeiro de 2024 e com fundamento no Regulamento Geral dos Cursos de Pós-Graduação da Universidade Federal de Campina Grande - UFCG, juntamente com o Sr(a) SAMARA SONALE SANTOS SAMPAIO, candidato(a) ao grau de MESTRE em Ciência da Computação, presentes ainda professores e alunos do referido centro e demais presentes. Abertos os trabalhos, o(a) Senhor(a) Presidente da Comissão Examinadora anunciou que a reunião tinha por finalidade a apresentação e julgamento da dissertação "ANALISANDO CONFIGURAÇÕES DE PROVISIONAMENTO AUTOMÁTICO NA NUVEM: UMA ABORDAGEM BASEADA EM SIMULAÇÃO", elaborada pelo(a) candidato(a) acima designado, sob a orientação do(s) Professor(es) FÁBIO JORGE ALMEIDA MORAIS e THIAGO EMMANUEL PEREIRA DA CUNHA SILVA, com o objetivo de atender as exigências do Regulamento Geral dos Cursos de Pós-Graduação da Universidade Federal de Campina Grande - UFCG. A seguir, concedeu a palavra ao (a) candidato(a), o qual, após salientar a importância do assunto desenvolvido, defendeu o conteúdo da dissertação. Concluída a exposição e defesa da candidata, passou cada membro da Comissão Examinadora a arguir a mestranda sobre os vários aspectos que constituíram o campo de estudo tratado na referida dissertação. Terminados os trabalhos de arguição, o(a) Senhor(a) Presidente da Comissão Examinadora determinou a suspensão da sessão pelo tempo necessário ao julgamento da dissertação. Reunidos, em caráter secreto, no mesmo recinto, os membros da Comissão Examinadora passaram à apreciação da dissertação. Reaberta a sessão, o(a) Presidente da Comissão Examinadora anunciou o resultado do julgamento, tendo assim, a candidato obtida o Conceito APROVADO. Reaberta a sessão, o(a) Presidente da Comissão Examinadora anunciou o resultado do julgamento, tendo a seguir encerrado a sessão, da qual lavrei a presente ata, que vai assinada por mim, Paloma Nascimento Porto, pelos membros da Comissão Examinadora e pela candidata. Campina Grande, 15 de fevereiro de 2024.

Documento assinado eletronicamente por **PALOMA NASCIMENTO PORTO**, **ASSISTENTE EM ADMINISTRACAO**, em 19/02/2024, às 11:20, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **FABIO JORGE ALMEIDA MORAIS**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 19/02/2024, às 11:33, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Paulo Ditarso Maciel Júnior**, **Usuário Externo**, em 19/02/2024, às 11:46, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **REINALDO CEZAR DE MORAIS GOMES**, **PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 20/02/2024, às 14:32, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **THIAGO EMMANUEL PEREIRA DA CUNHA SILVA**, **PROFESSOR 3 GRAU**, em 21/02/2024, às 08:16, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

Documento assinado eletronicamente por **Samara Sonale Santos Sampaio**, **Usuário Externo**, em 04/03/2024, às 16:04, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da Portaria SEI nº 002, de 25 de outubro de 2018.

A autenticidade deste documento pode ser conferida no site https://sei.ufcg.edu.br/autenticidade, informando o código verificador **4204752** e o código CRC **15944028**.

Referência: Processo nº 23096.007710/2024-20 SEI nº 4204752

Resumo

A demanda oscilante de uma aplicação torna o provisionamento automático de recursos uma atividade complexa, pois é necessário garantir que a infraestrutura suporte a carga de trabalho sem prejudicar o desempenho da aplicação, enquanto se evita custos desnecessários. Para isso, uma política de provisionamento automático deve ser configurada com cuidado, enfrentando a complexidade da grande quantidade de parâmetros envolvidos. Além disso, estratégias eficazes para uma aplicação podem não ser adequadas para outra, e escolhas inadequadas podem causar atrasos na alocação de recursos durante picos de demanda, afetando negativamente a experiência do usuário e elevando custos operacionais. Tais escolhas podem ser ainda mais desafiadoras devido à inexperiência ou falta de conhecimento do profissional responsável. Dessa forma, este estudo aborda a complexidade na escolha de configurações apropriadas para o provisionamento automático, tendo em vista que estudos mostram que essa escolha é feita através de intuição ou de testes limitados na configuração em produção. Nossa proposta é uma metodologia acompanhada de ferramentas para guiar os SREs na escolha e configuração eficiente de políticas de provisionamento automático, a qual permite o usuário simular distintas políticas e/ou configurações e analisar o comportamento da oferta de recursos através de métricas provenientes do estado da arte e outras desenvolvidas neste estudo. A metodologia juntamente com as análises permitem observar o comportamento da demanda e da oferta de recursos mediante modificações na configuração, além de facilitar a replicação dessas análises, seja para fins acadêmicos ou da indústria. Essa abordagem permite a realização de testes antes da implantação em ambiente de produção, proporcionando insights para uma tomada de decisão mais precisa.

Abstract

The fluctuating demand of an application makes the automatic provisioning of resources a complex activity, as it is necessary to ensure that the infrastructure supports the workload without impairing the application's performance, while avoiding unnecessary costs. For this, an automatic provisioning policy must be carefully configured, facing the complexity of the large number of parameters involved. Moreover, effective strategies for one application may not be suitable for another, and inappropriate choices can cause delays in resource allocation during peak demand, negatively affecting the user experience and increasing operational costs. Such choices can be even more challenging due to the inexperience or lack of knowledge of the professional responsible. Thus, this study addresses the complexity in choosing appropriate settings for automatic provisioning, considering that studies show that this choice is made through intuition or limited tests in the production configuration. Our proposal is a methodology accompanied by tools to guide SREs in the efficient choice and configuration of automatic provisioning policies, which allows the user to simulate different policies and/or configurations and analyze the behavior of resource offering through metrics from the state of the art and others developed in this study. The methodology along with the analyses allow observing the behavior of demand and resource offering through modifications in the configuration, in addition to facilitating the replication of these analyses, whether for academic or industrial purposes. This approach allows for testing before deployment in a production environment, providing insights for more precise decision-making.

Agradecimentos

A Deus, por ter estado comigo durante toda esta jornada, fornecendo a fé e a força necessárias para superar os desafios, e a Nossa Senhora, por sua constante intercessão em minha vida.

Aos meus pais, Graça e Salviano, por tudo que fizeram e fazem por mim. Desde a minha infância, eles ensinaram o valor da educação, sacrificaram-se para prover tudo que foi necessário e nunca deixaram faltar amor e apoio;

Às minhas irmãs, Sabrinne e Sarah que são pilares essenciais em minha vida. O incentivo e apoio delas garantem que nunca estarei sozinha;

Ao meu namorado, Jonas, pela sua incansável paciência, compreensão e incentivo, trazendo apoio e leveza na realização deste mestrado.

Aos meus tios, em especial às minhas tias Fátima, Jacinta (*in memoriam*), Ana e Maria, que ofereceram não apenas suporte material, mas também orações e, acima de tudo, o exemplo de serem professoras, fortificando o ensinamento de que a educação é um caminho transformador;

Aos amigos com quem compartilhei momentos durante a vida acadêmica, na graduação, pós graduação, nos projetos no LACINA (Laboratório de Computação Inteligente Aplicada) e no time Tinyverse no LSD (Laboratório de Sistemas Distribuídos), onde vivenciamos experiências de aprendizado e crescimento juntos. Gostaria de destacar, em particular, Ruan, Caetano, Anna Beatriz e Glauber cujo apoio e incentivo foram fundamentais na realização e conclusão deste trabalho;

A Kilian, por ter ajudado diretamente com o desenvolvimento do simulador utilizado.

Aos meus orientadores, Fábio e Thiago Emmanuel, por todos os ensinamentos, orientações e compreensão, mesmo quando eu não conseguir cumprir com as expectativas;

A todos os professores e demais funcionários das instituições que contribuíram com a minha formação acadêmica e pessoal. Em especial a Cleide, que com cumprimentos calorosos, conselhos e preocupação com o bem-estar dos alunos tornou a caminhada mais agradável;

Enfim, a todos que, de forma direta ou indireta, acrescentaram nesta jornada.

Conteúdo

1 Introdução **1**

1.1 Contextualização 1

1.2 Definição do Problema 4

1.3 Objetivos e Contribuições 4

1.4 Estrutura do Documento 5

2 Fundamentação teórica **6**

2.1 Demanda 6

2.2 Políticas de Provisionamento 6

2.3 Métricas 8

2.3.1 Métricas Operacionais 8

2.3.2 Métricas de Avaliação 9

2.4 Avaliação de Configurações de políticas de provisionamento automático . . 16

3 Trabalhos Relacionados **19**

3.1 Métricas 19

3.2 Políticas de provisionamento automático 20

3.3 Avaliação do provisionamento automático 21

4 Metodologia **24**

4.1 Caracterização do Estudo 24

4.2 Dados 25

4.3 Simulador 27

4.4 Métricas de avaliação das políticas 30

4.4.1 Taxa de Atendimento de Requisições (TAR) 31

4.4.2 Porcentagem de desperdício estimado 33
4.5 Simulação de Configurações e Análises 34

5 Validação do Simulador **36**
5.1 Aplicação A 36
5.2 Aplicação B 37

6 Análise da configuração da empresa **39**
6.1 Aplicação A 39
6.1.1 Parâmetros de configuração de provisionamento 39
6.1.2 Análise da utilização de *cores* 40
6.1.3 Avaliação da alocação de recursos 41
6.1.4 Avaliação dos limiares que acionam ações de provisionamento . . . 43
6.1.5 Análise da capacidade de processamento baseado em requisições . 44
6.1.6 Avaliação quantitativa de desempenho e eficiência de recursos . . . 45
6.1.7 Propostas de melhoria dos parâmetros de provisionamento automático 47
6.2 Aplicação B 47
6.2.1 Parâmetros de configuração de provisionamento 48
6.2.2 Análise da utilização de *cores* 48
6.2.3 Avaliação da alocação de recursos 49
6.2.4 Avaliação dos limiares que acionam ações de provisionamento . . . 50
6.2.5 Análise da capacidade de processamento baseado em requisições . 50
6.2.6 Avaliação Quantitativa de Desempenho e Eficiência de Recursos . . 52
6.2.7 Propostas de Otimização dos Parâmetros de Provisionamento Automático 54

7 Análise das hipóteses sobre como alterar ações de provisionamento **56**
7.1 Aplicação A 56
7.1.1 Avaliação da alocação de recursos 57
7.1.2 Avaliação dos limiares que acionam ações de provisionamento . . . 59
7.1.3 Análise de requisições e capacidade de processamento baseado em requisições 62

7.1.4 Discussão das modificações na configuração da Aplicação A 69
7.2 Aplicação B 71
7.2.1 Avaliação da alocação de recursos 72
7.2.2 Avaliação dos limiares que acionam ações de provisionamento . . . 73
7.2.3 Discussão das modificações na configuração da Aplicação B 77

8 Conclusões **80**

9 Trabalhos Futuros **82**

Lista de Símbolos

IaaS	*Infrastructure as a Service*
VM	*Virtual Machine*
QoS	*Quality of Service*
SLA	*Service Level Agreement*
SRE	*Site Reliability Engineering*
ADI	*Auto-scaling Demand Index*
MAE	Erro Médio Absoluto
TAR	Taxa de Atendimento de Requisições
RMSE	Raiz do Erro Quadrático Médio
R^2	Coeficiente de Determinação

Lista de Figuras

2.1 Representação gráfica da acurácia de superprovisionamento em um instante específico. As barras de cor mais escura e mais clara ilustram a oferta e a demanda de recursos, respectivamente. A seta curva indica a quantidade de recursos oferecidos além da demanda, evidenciando o superprovisionamento. 12

2.2 Gráfico para exemplificar como a métrica ADI é obtida. As áreas sombreadas indicam períodos de desperdício de recursos (em azul) e violações da Qualidade de Serviço (QoS) (em rosa), com linhas tracejadas denotando os limites de utilização aceitáveis de 60% (inferior) e 80% (superior). A acumulação dessas áreas sombreadas ao longo do tempo resulta no ADI. 13

2.3 Ilustração de aplicações com características distintas, mas que apresentam os mesmos valores de acurácia e tempo de provisionamento incorreto, demonstrando a importância da métrica jitter para identificar se ocorreram muitas adaptações. 15

4.1 Parâmetros das políticas de provisionamento automático implementadas no simulador. 28

4.2 Curvas de oferta e demanda, ilustrando situações de sub e superprovisionamento representadas pelas letras U e O, respectivamente. 31

5.1 Variação temporal da diferença entre a utilização de *cores* no dados reais e simulados, utilizando os dados da Aplicação A. 37

5.2 Variação temporal da diferença entre a utilização de *cores* no dados reais e simulados, utilizando os dados da Aplicação B. 38

6.1 Variação diária no uso de *cores* em Julho da Aplicação A, indicando uma oscilação entre aproximadamente 100 e 300. 41
6.2 Distribuição do uso de *cores* durante o mês de Julho na Aplicação A, com os dados organizados por hora, evidenciando o padrão de uso em cada período do dia. 42
6.3 Comparativo diário das instâncias disponíveis em relação às necessárias em Julho da Aplicação A, mostrando que, predominantemente, a oferta de recursos supera a demanda. 43
6.4 Monitoramento da utilização percentual do sistema em Julho. 44
6.5 Análise temporal da capacidade de processamento de requisições. 45
6.6 Variação diária na utilização de *cores* em Dezembro, com os dados da Aplicação B. 49
6.7 Distribuição da utilização de *cores* no mês de Dezembro com os dados agrupados por hora, utilizando os dados da Aplicação B. 50
6.8 Comparativo diário entre instâncias disponíveis e necessárias em Dezembro, utilizando os dados da Aplicação B, mostrando que houveram momentos de sub e superprovisionamento. 51
6.9 Monitoramento da utilização percentual do sistema em Julho, utilizando os dados da Aplicação B com a Configuração Base. 52
6.10 Análise temporal da capacidade de processamento de requisições, utilizando os dados da Aplicação B. 53

7.1 Análise diária do número de instâncias disponíveis em relação às necessárias, sob a política de provisionamento simulada com o limite superior ajustado para 75%. 58
7.2 Análise diária do número de instâncias disponíveis em relação às necessárias, sob a política de provisionamento simulada com os dados da Aplicação A e com a Configuração 3, na qual a quantidade de incremento de recursos igual a 16 *cores*. 58

7.3 Análise temporal do comparativo entre a quantidade de instâncias disponíveis e necessárias, utilizando o limite superior igual a 75% e quantidade de *cores* em ações de expansão de recursos igual a 16, com os dados da Aplicação A. 59

7.4 Esta análise foca no comparativo temporal entre a quantidade de instâncias disponíveis e as que são efetivamente necessárias. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 5. Esta configuração específica difere da Configuração Base apenas no que diz respeito à capacidade máxima de recursos alocados. 60

7.5 Esta análise foca no comparativo temporal entre a quantidade de instâncias disponíveis e as que são efetivamente necessárias. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 6. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao período de avaliação que foi alterado de 2 para 4 61

7.6 Esta análise foca no comparativo temporal entre a quantidade de instâncias disponíveis e as que são efetivamente necessárias. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 7. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao tempo de espera que foi alterado de 3 para 1. 62

7.7 Utilização do sistema ao longo do tempo, destacando os limites que acionam as ações de provisionamento automático. Simulando com a Configuração 2 e os dados da Aplicação A. 63

7.8 Monitoramento de utilização percentual do sistema, destacando os limites superior e inferior. Utilizando os dados da Aplicação A com a Configuração 3, na qual a quantidade de *cores* em ações de expansão de recursos igual a 16. 63

7.9 Monitoramento da utilização de recursos destacando os limites superior e inferior, empregando o limite superior igual a 75% e quantidade de *cores* em ações de expansão de recursos igual a 16. 64

7.10 Esta imagem apresenta a utilização de recursos, mostrando também os limites que foram estabelecidos na configuração para acionar ações de provisionamento automático. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 5. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao a capacidade máxima. 65

7.11 Esta imagem apresenta a utilização de recursos, mostrando também os limites que foram estabelecidos na configuração para acionar ações de provisionamento automático. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 6. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao período de avaliação que foi alterado de 2 para 4. 66

7.12 Esta imagem apresenta a utilização de recursos, mostrando também os limites que foram estabelecidos na configuração para acionar ações de provisionamento automático. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 7. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao tempo de espera que foi alterado de 3 para 1. 67

7.13 Análise temporal do comparativo entre a quantidade de requisições do volume de entrada e a capacidade de atendimento das mesmas. Utilizando os dados da Aplicação A e a Configuração 2, cujo o limite superior é igual a 75% 68

7.14 Análise diária do número de instâncias disponíveis em relação às necessárias, sob a política de provisionamento simulada com o limiar alvo igual a 65%, utilizando os dados referentes a Aplicação B, com a Configuração 2. . 72

7.15 Utilização do sistema ao longo do tempo, destacando o limite que aciona as ações de provisionamento automático. Utilizando a política de provisionamento simulada com o limiar alvo igual a 65%, utilizando os dados referentes a Aplicação B, com a Configuração 2. 74

7.16 Utilização do sistema ao longo do tempo, destacando o limite que aciona as ações de provisionamento automático. Utilizando a política de provisionamento simulada com o limiar alvo igual a 65%, utilizando os dados referentes a Aplicação B, com a Configuração 3. 75
7.17 Análise temporal da capacidade de processamento de requisições, utilizando os dados da Aplicação B e simulando com a Configuração 2. 76
7.18 Análise temporal da capacidade de processamento de requisições, utilizando os dados da Aplicação B e simulando com a Configuração 3. 77

Lista de Tabelas

4.1 Dados utilizados das instâncias de máquinas virtuais. 25
4.2 Dados utilizados dos balanceadores de carga. 26
4.3 Dados utilizados dos grupos de provisionamento automático. 26
4.4 Dados gerados após o processamento das tabelas 4.1, 4.2 e 4.3. Com os quais foi possível calcular métricas e gerar gráficos. 27

6.1 Parâmetros definidos para a Aplicação A durante o mês de análise dos dados. Exceto a capacidade máxima, os demais valores são os mesmos utilizados pela empresa para essa aplicação no mês de Julho. 40
6.2 Métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando a configuração estabelecida pela empresa com os dados da Aplicação A. 46
6.3 Parâmetros definidos para a Aplicação B durante o mês de análise dos dados. 48
6.4 Métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando a configuração estabelecida pela empresa, utilizando os dados da Aplicação B. 54

7.1 Comparativo das métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando diferentes configurações, incluindo a Configuração 4 com limite superior de 75% e alocação de *cores* fixa em ações de expansão de recursos. 69
7.2 Métricas avaliativas alterando a capacidade máxima, período de avaliação e tempo de espera para ações de provisionamento em relação a Configuração Base, com os dados da Aplicação A. 70

7.3 Parâmetros definidos para a Aplicação B durante o mês de análise dos dados, utilizando a Configuração 3 para simular. 71

7.4 Comparativo das métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando os dados da Aplicação B com as Configurações Base, 2 e 3. 78

Capítulo 1

Introdução

1.1 Contextualização

Um dos modelos oferecidos na computação em nuvem é o de Infraestrutura como Serviço (IaaS, do inglês *Infrastructure as a Service*), que disponibiliza uma estrutura flexível, escalável e acessível sob demanda. Essa característica permite que em cenários de variação da demanda da aplicação em execução na infraestrutura, seja possível provisionar dinamicamente a infraestrutura (adicionar ou remover recursos) visando a Qualidade de Serviço (QoS) esperada para a aplicação implantada no ambiente de Nuvem.

Esse serviço oferece recursos computacionais virtuais no formato de Máquinas Virtuais (VMs, do inglês *Virtual Machine*) ou contêineres, que estabelecem uma capacidade de processamento, armazenamento, memória e banda. Desta forma, os usuários adquirem os recursos e são tarifados em função da capacidade e do tempo de uso dos recursos, em um modelo de pague apenas pelo o utilizado (do inglês, *pay-as-you-go*) [7].

Esse modelo vem se consolidando como a opção mais escolhida para o provisionamento de infraestruturas computacionais [4]. Segundo uma pesquisa do *Gartner, Inc.* [8], o mercado mundial de infraestrutura como serviço experimentou um crescimento significativo de 29,7% em 2022, alcançando US$ 120,3 bilhões, um salto considerável em relação aos US$ 92,8 bilhões registrados em 2021. Nesse cenário, a Amazon manteve sua posição dominante como provedora de IaaS, e logo em seguida se destacaram outros provedores como Microsoft, Alibaba, Google e Huawei.

Diante desse crescimento expressivo no mercado de IaaS, é importante entender as téc-

nicas utilizadas para alocar recursos em tais infraestruturas. Essa alocação pode ser feita de duas maneiras principais: verticalmente, ajustando a capacidade de uma Máquina Virtual (VM) em termos de CPU e memória, limitada pelos recursos da máquina física; e horizontalmente, variando a quantidade de VMs para atender à demanda, o que oferece maior flexibilidade e escalabilidade [11].

A decisão de como e quanto provisionar recursos na computação em nuvem afeta a aplicação e a infraestrutura. O superprovisionamento, com recursos além do necessário, mantém a QoS mas leva a custos desnecessários por recursos ociosos. Já o subprovisionamento, com recursos insuficientes, prejudica a QoS e pode violar acordos do Nível de Serviço da Aplicação (SLA, do inglês *Service Level Agreement*), resultando em impactos negativos na experiência do usuário e custos adicionais.

Desta forma, a decisão sobre a realização de operações de provisionamento não é uma tarefa trivial, principalmente em casos em que a demanda da aplicação pode variar de forma imprevisível. Esta variação exige que o provisionamento determine, de forma contínua, quando e em qual quantidade os recursos devem ser alocados ou desalocados. Diante desse cenário, utiliza-se abordagens de provisionamento automático (do inglês, *auto-scaling*), que ajustam dinamicamente a infraestrutura.

Essa abordagem busca garantir que a aplicação mantenha uma QoS consistente, respondendo às variações de carga sem degradação de desempenho. Adicionalmente, é economicamente vantajosa, pois ajusta o consumo de capacidade computacional às reais necessidades da aplicação, evitando gastos desnecessários com infraestrutura subutilizada ou sobrecargas que possam resultar em penalidades financeiras e insatisfação do cliente. Isso é alcançado por meio do monitoramento contínuo de métricas, como a utilização da CPU e a demanda de poder de processamento. Quando o sistema detecta que a demanda está aumentando, ele aloca mais recursos. Por outro lado, quando a demanda diminui, recursos em excesso são reduzidos automaticamente para evitar desperdício.

Para efetivar essa abordagem, é crucial configurar as estratégias de provisionamento de forma precisa, levando em conta as características específicas da aplicação. Além disso, há uma diversidade de políticas de provisionamento, cada uma exigindo a definição de parâmetros distintos para ser eficaz. Essas definem estratégias de como e quando os recursos na nuvem serão provisionados para uma determinada aplicação, por meio de ações de provisi-

onamento disparadas por fatores como carga de trabalho da aplicação, horário, histórico de utilização, entre outros [17]. Além das características particulares de cada aplicação é crucial decidir continuamente sobre as ações de provisionamento a serem realizadas em função de variações na demanda. Devido a isso, inúmeras políticas desse tipo foram propostas nos últimos anos [20].

Dada a importância dessa escolha, para atingir a QoS satisfatória e ao mesmo tempo economizar recursos, é de extrema importância que a política de provisionamento seja escolhida e configurada de forma adequada. Escolhas errôneas podem comprometer tanto o custo, com infraestrutura que não estão sendo utilizados, quanto o desempenho da aplicação.

Devido a isso, configurar adequadamente essas políticas é desafiante, visto à vastidão de parâmetros ajustáveis que interferem diretamente no comportamento e na eficiência da política adotada. Como por exemplo, na AWS, existem três tipos principais de políticas de provisionamento automático e cada uma dessas oferece diferentes opções de configuração (estão disponíveis no Apêndice A), permitindo ajustes finos com base em métricas específicas, valores-alvo, tempos de aquecimento de instâncias e outras configurações. Essa variedade oferece flexibilidade, mas também adiciona complexidade à escolha da política mais adequada para cada aplicação.

Além do mais, cada aplicação possui suas peculiaridades e padrões de uso, e um parâmetro que é bem definido para uma aplicação pode não ser ideal para outra. Esse cenário exige uma análise cuidadosa e muitas vezes experimentação iterativa para se chegar a um ajuste ideal.

Na prática, configurar o provisionamento automático pode ser desafiador e às vezes até inviável ou infactível devido ao risco de comprometer a QoS ou gerar altos custos financeiros. Para medir adequadamente o impacto dessas configurações, é necessário realizar as modificações em ambientes reais ou réplicas fiéis do ambiente de implantação da aplicação. Dada a complexidade desse cenário, surge um desafio adicional: a forma como os sistemas são configurados na prática, muitas vezes de maneira improvisada e em ambientes reais, sem a segurança de conhecer os valores ideais.

O estudo conduzido por Silva [23] revelou que 75% dos participantes já precisaram definir uma ação de provisionamento automático. Desses, 20% basearam suas decisões na intuição, enquanto os restantes optaram por algum tipo de teste. Contudo, dada a natureza

crítica das aplicações em produção, esses testes frequentemente são experimentos rápidos realizados em serviços em ambiente de produção, dificultando a exploração de diferentes combinações de configurações em um curto período.

O estudo de Silva [23] ainda destaca uma lacuna na literatura existente, relacionada à aplicabilidade prática dos cenários utilizados nas pesquisas. Frequentemente, esses estudos empregam cargas de trabalho que podem não refletir adequadamente a realidade, seja por serem desatualizadas, sintéticas, de escala reduzida, ou com um escopo limitado. Além disso, muitas vezes consideram estratégias de provisionamento que, embora teoricamente válidas, apresentam desafios consideráveis para a implementação prática.

Diante disso, identificou-se uma oportunidade para conduzir pesquisas que envolvam a analise do comportamento de diferentes configurações de provisionamento automático.

1.2 Definição do Problema

O estudo propõe auxiliar no desafio de configurar políticas de provisionamento automático em ambiente IaaS, visando equilibrar a manutenção da qualidade de serviço e a minimização de custos, diante da problemática da dificuldade em ajustar essas políticas de forma eficaz, devido à complexidade e à grande quantidade de parâmetros ajustáveis, além da imprevisibilidade das demandas das aplicações.

Isso frequentemente leva a uma abordagem conservadora, resultando em superprovisionamento e, consequentemente, custos desnecessários. A solução proposta visa facilitar a comparação de diferentes configurações de provisionamento, buscando um equilíbrio que evite tanto o comprometimento do desempenho das aplicações quanto o desperdício de recursos.

1.3 Objetivos e Contribuições

Diante do cenário previamente exposto, o objetivo principal deste trabalho é propor um fluxo de estudo que auxilie na análise e avaliação do desempenho de distintas configurações de provisionamento automático em ambientes simulados. Esta análise será conduzida através da comparação e investigação da reação da carga de trabalho sob diferentes configurações,

aplicando métricas avançadas encontradas na literatura e utilizando conhecimentos práticos consolidados na área.

Como contribuição foi desenvolvido um simulador capaz de modelar o comportamento de variadas políticas de provisionamento automático. Essa ferramenta é essencial para avaliar as implicações de diferentes estratégias sem a necessidade de implementação direta em ambientes de produção reais. Além disso, foi elaborado um fluxo de estudo detalhado, que serve como um guia para orientar decisões de provisionamento, visando maximizar a eficácia e a eficiência desses processos.

Complementarmente, foi fornecido um conjunto de métricas selecionadas e exemplos de análises, servindo como um modelo para interpretar os resultados e adaptá-los a outras situações. Por fim, foi demonstrado a aplicação prática dessas métricas de provisionamento automático, analisando como se comportam em ambientes simulados e destacando sua utilidade prática.

1.4 Estrutura do Documento

No Capítulo 2, apresentamos a fundamentação teórica, onde são discutidas diversas métricas encontradas na literatura, as políticas de provisionamento utilizadas neste estudo, e as abordagens para sua avaliação. Em seguida, no Capítulo 3, exploramos os trabalhos relacionados, focando na avaliação e configuração do provisionamento automático.

A metodologia utilizada é detalhada no Capítulo 4. Detalhamos a caracterização do estudo, os dados utilizados, simulador, métrica e como as análises foram conduzidas. No Capítulo 5, demonstramos como a validação do simulador foi realizada.

Os resultados obtidos e as análises realizadas são compartilhados nos Capítulos 6 e 7. Posteriormente, no Capítulo 8, sintetizamos as principais descobertas e conclusões do trabalho. Por fim, no Capítulo 9, delineamos possíveis direções para futuras pesquisas, sugerindo áreas que podem se beneficiar de nossa pesquisa e explorando possíveis expansões do trabalho realizado.

Capítulo 2

Fundamentação teórica

Neste capítulo, compreendendo que o propósito principal do provisionamento automático é alinhar a oferta de recursos à demanda, focamos nossa atenção às politicas de provisionamento, métricas apresentadas na literatura e na avaliação das configurações das políticas. Estas métricas são fundamentais para avaliar a eficácia com que o provisionamento está sendo executado.

2.1 Demanda

Inicialmente é necessário entender que a demanda, em um contexto de computação em nuvem, pode ser medida por uma variedade de indicadores, como por exemplo a quantidade de vCPUs utilizadas, o número de requisições a um serviço ou aplicação, o uso de memória, largura de banda de rede, e IOPS (do inglês, *Input/Output Operations Per Second*) [28].

Para obtermos a demanda neste estudo, adotamos a quantidade de vCPUs como indicador do número de *cores* em uso, dada a disponibilidade dessa informação nos dados coletados. Consequentemente, procedemos ao cálculo de quantos *cores* uma aplicação específica consumia em um intervalo de tempo definido.

2.2 Políticas de Provisionamento

As políticas de provisionamento automático são essenciais para determinar como a disponibilização de recursos será executada em um sistema. A escolha desta pode variar com base

na abordagem desejada, sendo reativa ou preditiva [18].

A abordagem reativa é direta e age rapidamente ao detectar que certos limites predefinidos foram alcançados. Usando métricas como utilização da CPU, consumo de memória e tráfego de rede, que são configuradas com valores específicos, o sistema se adapta provisionando mais ou menos recursos assim que os limiares são atingidos [6]. No entanto, essa abordagem, apesar de sua simplicidade, pode não ser a mais eficiente em termos de custo. Pode haver situações em que ocorram grandes picos de demanda, causando atrasos até que os recursos adicionais estejam totalmente operacionais, o que pode comprometer a experiência do usuário [22].

A política preditiva emprega algoritmos de aprendizado de máquina. O objetivo é antecipar a demanda futura com base em dados históricos. Quando bem implementada e utilizada com uma demanda com o comportamento previsível, essa abordagem pode assegurar que os recursos estejam prontos bem antes de picos de demanda uma vez que o provisionamento é realizado com base na predição, evitando gargalos potenciais [18]. No entanto, os algoritmos de aprendizado de máquina requerem dados de qualidade e, mesmo assim, previsões sempre carregam um grau de incerteza. Em situações com variações imprevisíveis, isso pode levar a provisionamentos inadequados, com repercussões potencialmente negativas para a aplicação [13].

Neste estudo, as políticas de provisionamento automático adotadas baseiam-se em estratégias reativas. Essa escolha foi diretamente influenciada pelo fato de que a empresa, cujos dados foram utilizados na nossa análise, já emprega essas mesmas políticas em sua infraestrutura. Além disso, a escolha por seguir as diretrizes da Amazon Web Services (AWS) foi reforçada pela própria adesão da empresa a esse provedor de IaaS. As abordagens reativas adotadas foram o Provisionamento Simples (*Simple Scaling*) e o Provisionamento de Rastreamento de Meta (*Target Tracking*).

O provisionamento simples visa manter a porcentagem de utilização dentro de limites definidos. O limite inferior é a porcentagem mínima de utilização permitida, enquanto o limite superior é a máxima. O tipo de recursos pode ser em *cores* ou porcentagem, determinando como ajustar a quantidade de recursos durante o provisionamento. A quantidade de recursos a serem expandido ou reduzidos dependem desse tipo, indicando quantos *cores* adicionar ou remover. O período de avaliação refere-se ao número de períodos consecutivos considerados

para verificar se os limites foram ultrapassados.

O provisionameto com rastreamento de meta dimensiona automaticamente a quantidade de recursos ofertados com base em um valor alvo. Por exemplo, supondo que um aplicativo que seja executado em duas instâncias e a política está configurada pra que a utilização de CPU permaneça em cerca de 50%. Então a quantidade de recursos se expandirá (aumentará a capacidade) quando a CPU exceder 50% para lidar com o aumento da carga. Ele reduzirá a escala horizontalmente (diminuição da capacidade) quando a CPU estiver abaixo de 50% para otimizar os custos durante os períodos de baixa utilização. Isso fornece capacidade extra para lidar com picos de tráfego sem manter um número excessivo de recursos ociosos.

A seção 4.3 detalha como essas políticas foram implementadas neste estudo.

2.3 Métricas

As métricas de provisionamento automático são fundamentais para a gestão dinâmica de recursos, orientando a adequação dos recursos às flutuações na demanda. Elas garantem que o provisionamento esteja em sintonia com as necessidades atuais, atuando em duas categorias principais: operacionais, que iniciam o provisionamento, e de avaliação, que mensuram sua eficácia.

No entanto, além destas, é essencial considerar também as métricas relacionadas ao gerenciamento de recursos e ao custo associado. Em cenários onde o existe superprovisionamento, as métricas operacionais e de avaliação podem indicar uma performance satisfatória, mas isso pode ocultar ineficiências no uso dos recursos e em custos desnecessariamente elevados.

2.3.1 Métricas Operacionais

Métricas operacionais, como utilização de CPU, memória, largura de banda de rede e I/O de disco, atuam diretamente nas decisões de provisionamento automático. Quando certos limiares são atingidos, como por exemplo, a utilização da CPU exceder um certo percentual, uma ação de provisionamento é iniciada automaticamente para ajustar os recursos conforme necessário.

Dessa forma, é evidente que a natureza específica de uma aplicação influencia quais métricas são mais relevantes. Por exemplo, uma aplicação que é intensiva em CPU pode priorizar métricas relacionadas à CPU, enquanto uma plataforma de *streaming* de vídeo pode focar na largura de banda de rede. Esta consideração é essencial para garantir que o provisionamento seja tanto responsivo quanto eficaz, adequando-se às necessidades específicas de cada aplicação.

Entretanto, é importante destacar que a seleção e configuração eficiente de métricas de provisionamento exigem um entendimento profundo da aplicação em questão. O responsável por essa tarefa deve ter um conhecimento detalhado das características específicas e dos requisitos da aplicação, incluindo seu comportamento sob diferentes condições de carga e padrões de uso. Isso é particularmente importante em cenários envolvendo aplicações novas ou aquelas que estão passando por evoluções dinâmicas. Nestes casos, o ambiente e as condições de operação podem mudar rapidamente, tornando essencial um acompanhamento mais frequente e atento das métricas. Monitorar de perto essas e entender como elas se correlacionam com as mudanças na demanda permite ajustes mais ágeis e precisos nas estratégias de provisionamento.

Dessa forma, a seleção e configuração de métricas não são tarefas triviais. Uma escolha inapropriada ou configuração inadequada dos limiares de provisionamento pode resultar em provisionamentos imprecisos, ocasionando custos desnecessários ou incapacidade de atender às demandas. Nesse contexto, em algumas situações, combinar múltiplas métricas em uma estratégia de provisionamento composta pode ser benéfico para evitar respostas prematuras e garantir precisão.

2.3.2 Métricas de Avaliação

As métricas de avaliação têm como objetivo analisar o desempenho e a eficácia dos sistemas de provisionamento automático. Elas podem ser categorizadas de duas formas: orientadas ao sistema ou ao usuário.

Métricas orientadas ao sistema focam na avaliação da demanda e oferta de recursos, estimando a capacidade do sistema de atender às exigências da aplicação. Essas ajudam a entender se os recursos disponíveis estão sendo utilizados de maneira eficiente e se o sistema está conseguindo atender à demanda atual. Neste contexto, as métricas operacionais desem-

penham um papel fundamental. Por exemplo, a utilização da CPU é uma métrica operacional que indica o quanto do poder de processamento do sistema está sendo usado.

Monitorar a utilização da CPU pode revelar se o sistema está sobrecarregado ou subutilizado, permitindo ajustes precisos no provisionamento de recursos. Além dessa, outras métricas operacionais como uso de memória, largura de banda de rede e latência também são cruciais para avaliar o desempenho geral do sistema e garantir que ele tenha uma quantidade adequada de recursos para as demandas atuais e futuras da aplicação.

Por outro lado, as métricas orientadas ao usuário, como tempo de resposta, taxa de erro, tempo de atividade e taxa de solicitações rejeitadas, concentram-se principalmente na experiência e satisfação do cliente. Essas métricas são essenciais para avaliar o impacto das decisões de provisionamento automático no usuário final, assegurando que essas ações estejam de fato assegurando os níveis de qualidade da experiência do usuário.

Neste contexto, as métricas discutidas a seguir se tornam fundamentais, visto que, não apenas complementam as métricas básicas, mas também proporcionam uma visão mais abrangente e eficiente do desempenho do provisionamento automático. Essas ajudam a identificar como os recursos ofertados estão se adaptando a demanda e como essas ações de provisionamento afetam a experiência do usuário final.

Acurácia (Orientada ao sistema):

O estudo [6] descreve essa métrica como a quantidade de recursos que foram sub ou superprovisionados em relação à demanda. Dessa forma, acurácia de sub-provisionamento (θ_U) é a proporção entre o máximo da diferença da quantidade de recursos faltante e a demanda da aplicação e zero, em relação à demanda. De forma semelhante, a acurácia de superprovisionamento (θ_O) é a proporção entre o máximo da diferença da quantidade de recursos fornecidos em excesso e a demanda atual e zero, em relação a demanda. Para normalizar essa métrica pode-se dividir pelo tempo de duração do experimento. Com isso, são apresentadas as fórmulas 2.1 e 2.2.

$$\theta_U = \frac{100}{T} \times \sum_{t=1}^{T} \frac{\max(d_t - s_t, 0)}{d_t} \tag{2.1}$$

$$\theta_O = \frac{100}{T} \times \sum_{t=1}^{T} \frac{\max(s_t - d_t, 0)}{d_t} \tag{2.2}$$

Valores mais próximos de 0 nessas métricas indicam um provisionamento altamente eficiente, refletindo um alinhamento preciso entre a oferta e a necessidade de recursos. Dessa forma, assegura um equilíbrio ótimo entre desempenho e custo.

A Figura 2.1 ilustra de maneira prática como obtemos o cálculo da acurácia de superprovisionamento em um determinado instante. As barras representadas — a mais escura indicando 'Oferta' e a mais clara indicando 'Demanda' — mostram, respectivamente, a quantidade de recursos disponibilizados e a quantidade que seria efetivamente necessária. A seta curva rotulada com 'Quantidade de recursos oferecidos além da demanda' destaca o excesso de recursos provisionados em comparação com a demanda.

Para quantificar a acurácia de superprovisionamento da aplicação em um dado instante, calcula-se a porcentagem pela qual a oferta excedeu a demanda. Para obter essa métrica para todo o período analisado, realiza-se o somatório da quantidade de recursos oferecidos além da necessidade. Essa soma é então normalizada pelo tempo total do experimento, fornecendo uma medida padronizada e proporcional da acurácia do superprovisionamento ao longo do período considerado.

Da mesma forma que a acurácia de superprovisionamento é calculada, a acurácia de subprovisionamento é determinada, porém levando em consideração as situações onde a oferta de recursos é menor que a demanda necessária.

Do ponto de vista de negócios, a acurácia é uma métrica essencial para a auxiliar métricas de custos e eficiência operacional. Visto que, o sub-provisionamento pode levar a perdas de receita e danos à reputação devido à incapacidade de atender às demandas dos clientes. Por outro lado, o superprovisionamento resulta em gastos operacionais desnecessários.

Portanto, manter um equilíbrio entre θ_U e θ_O é crucial para garantir que os recursos sejam alocados de maneira eficiente, sem desperdiçar. Ela ainda proporciona uma visão clara de como os recursos estão sendo utilizados em relação às demandas atuais, permitindo ajustes precisos e estratégicos para melhorar tanto a eficiência operacional quanto a satisfação do cliente.

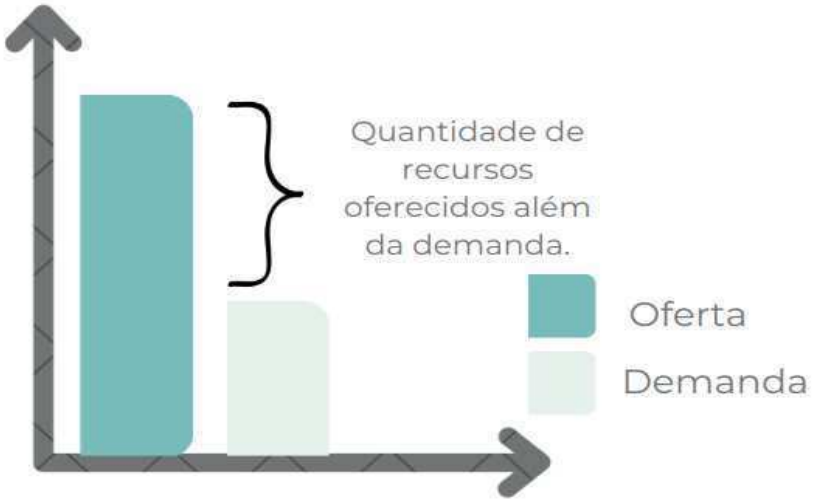

Figura 2.1: Representação gráfica da acurácia de superprovisionamento em um instante específico. As barras de cor mais escura e mais clara ilustram a oferta e a demanda de recursos, respectivamente. A seta curva indica a quantidade de recursos oferecidos além da demanda, evidenciando o superprovisionamento.

***Auto-scaling Demand Index - ADI* (Orientada ao sistema):**

Apresentada no estudo [17], essa métrica é explicada como a soma de todas as distâncias calculadas entre cada nível de utilização informado pelo sistema e o intervalo de utilização alvo definido pelo usuário, isto é, a diferença entre os níveis de utilização de recursos reais e desejados serão somados, em cada instante de tempo. Essa métrica será obtida de acordo a equação 2.3.

$$\sigma_t = \begin{cases} L - u_t, & \text{se } u_t \leq L \\ 0, & \text{se } L < u_t < U \\ u_t - U, & \text{caso contrário} \end{cases} \tag{2.3}$$

No provisionamento baseado em dois limiares, L refere-se ao limite inferior e U ao limite superior. Já para o provisionamento com um limiar alvo, U representa esse limiar e L refere-se ao limiar alvo menos a taxa de tolerância para ações de diminuir o provisionamento. E em ambas, u_t refere-se a utilização da aplicação no instante t.

A Figura 2.2 exemplifica como a métrica ADI é obtida. No gráfico, o eixo horizontal representa o tempo e o eixo vertical indica a utilização de recursos em porcentagem. As áreas

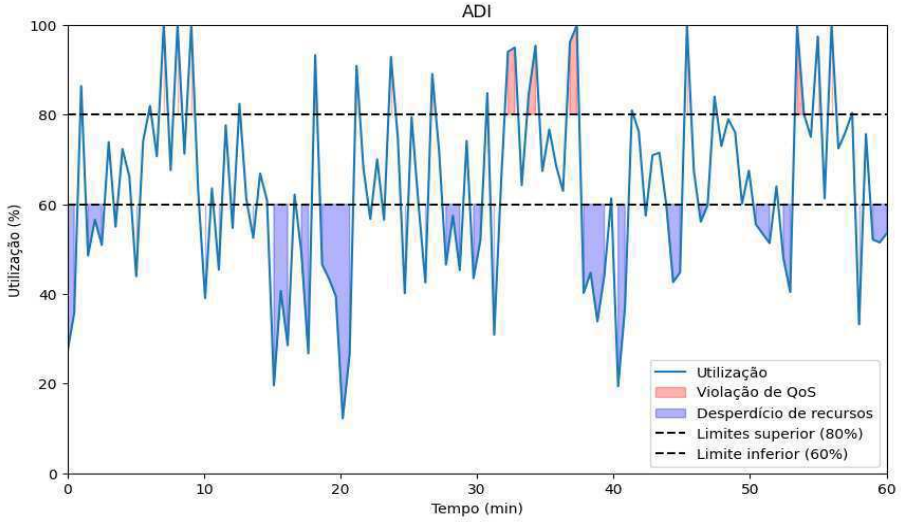

Figura 2.2: Gráfico para exemplificar como a métrica ADI é obtida. As áreas sombreadas indicam períodos de desperdício de recursos (em azul) e violações da Qualidade de Serviço (QoS) (em rosa), com linhas tracejadas denotando os limites de utilização aceitáveis de 60% (inferior) e 80% (superior). A acumulação dessas áreas sombreadas ao longo do tempo resulta no ADI.

sombreadas em azul e rosa refletem os períodos de desperdício de recursos e de violação da Qualidade de Serviço (QoS), respectivamente. O desperdício de recursos é identificado quando a utilização fica abaixo da linha tracejada inferior, definida como o limite de 60%. Por outro lado, a violação da QoS ocorre quando a utilização excede a linha tracejada superior, estabelecida em 80%. A cada ocorrência de utilização fora desses limites, a área correspondente é somada, resultando em um acúmulo ao longo do período analisado. Essa acumulação das áreas permite quantificar o ADI.

Embora as métricas de acurácia e ADI compartilhem um foco comum no provisionamento de recursos, elas se diferenciam em alguns aspectos. O ADI é voltado para a análise da variação na utilização de recursos em relação a um intervalo de utilização desejado. Isso permite identificar como o uso de recursos se ajusta, ou não, aos parâmetros definidos. Em contraste, as métricas de acurácia se concentram mais especificamente na mensuração do sub e superprovisionamento. Ao quantificar o quão alinhada a oferta de recursos está com a demanda real, elas revelam desajustes precisos e momentos críticos de ineficiência. A combinação dessas métricas oferece uma visão mais completa da gestão de recursos. Pro-

porcionando percepções sobre tendências e padrões gerais no uso de recursos ao longo do tempo e problemas pontuais e específicos. Isso não apenas facilita a identificação de áreas que precisam de ajustes imediatos, mas também auxilia na elaboração de estratégias de longo prazo para melhoria na disponibilização de recursos.

Tempo de provisionamento incorreto (Orientada ao sistema):

Esta métrica reflete a confiabilidade da configuração, pois nos ajuda a entender se houve desvios na quantidade de recursos desejada por curtos ou longos períodos. Ela indica a porcentagem de tempo em que o provisionamento foi insuficiente ou excessivo, representando assim os momentos de quebra do SLO [9; 6; 5].

O trabalho [9] apresenta as seguintes fórmulas para o cálculo das métricas de tempo de provisionamento incorreto de sub (Equação 2.4) e superprovisionamento (Equação 2.5).

$$\Gamma_U[\%] = \frac{\sum U}{T} \tag{2.4}$$

$$\Gamma_O[\%] = \frac{\sum O}{T} \tag{2.5}$$

Nestas equações, a variável T representa o período total do experimento. Por outro lado, U e O são acumuladores temporais que refletem, respectivamente, a duração total em que o sistema experimentou sub-provisionamento e superprovisionamento. T refere-se ao tempo de duração do experimento, U e O representam o somatório do tempo que o sistema ficou sub ou superprovisionado, respectivamente.

***Jitter* (Orientada ao sistema)**

Em estudos [9; 12] sobre a elasticidade dos sistemas, foi identificado que métricas convencionais, como acurácia e tempo de provisionamento incorreto, podem não capturar completamente o comportamento das aplicações, especialmente em cenários onde duas aplicações com métricas semelhantes exibem comportamentos operacionais muito diferentes.

Como ilustrado na Figura 2.3, embora as aplicações A e B possam exibir os mesmos valores para as métricas de acurácia e tempo de provisionamento incorreto, a Aplicação B desencadeia adaptações desnecessárias de fornecimento de recursos, enquanto a Aplicação

A aciona apenas algumas. Esse estudo introduziu essa métrica para capturar esses nuances, especialmente útil para modelos de precificação baseados em horas de instância e para visões operacionais que buscam minimizar sobrecargas de adaptação.

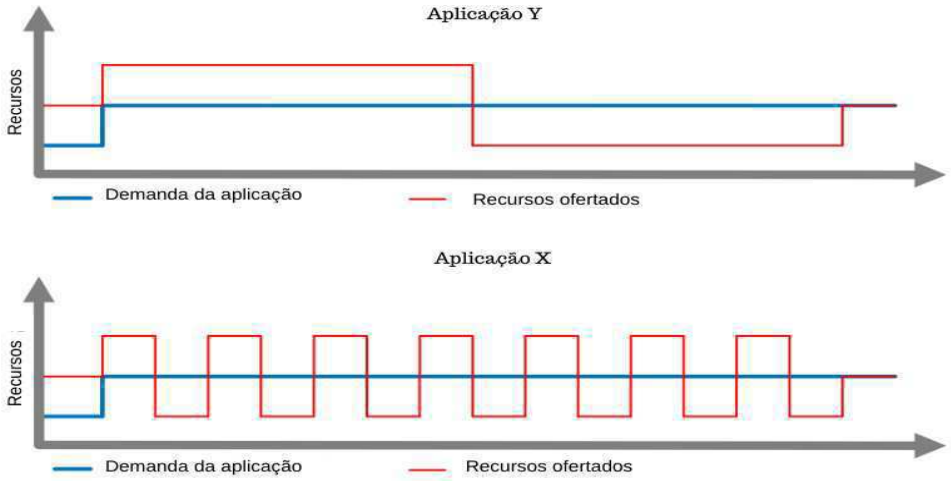

Figura 2.3: Ilustração de aplicações com características distintas, mas que apresentam os mesmos valores de acurácia e tempo de provisionamento incorreto, demonstrando a importância da métrica jitter para identificar se ocorreram muitas adaptações.

Na prática, essa métrica compara a quantidade de adaptações na curva de oferta (E_S) com o número de adaptações na curva de demanda (E_D). Se uma plataforma aloca ou desaloca mais de um recurso por vez, essas adaptações são contadas individualmente por unidade de recurso. A diferença entre essas adaptações é normalizada pela duração do período de medição (T). A equação 2.6 apresenta como obter essa métrica.

$$\rho = \frac{E_S - E_D}{T} \tag{2.6}$$

Velocidade Elástica (Orientada ao sistema)

Esta métrica avalia a eficácia do provisionamento automático, comparando o comportamento do sistema quando ele é aplicado versus quando não é. Ela é essencial para entender a rapidez e eficiência com que o sistema pode se adaptar às demandas variáveis, o que pode levar a economias de custo e melhor desempenho da aplicação. A velocidade elástica é

frequentemente usada para justificar a implementação do provisionamento automático em ambientes de nuvem [6; 1].

Essa métrica é obtida através das métricas já apresentandas nesse estudo: acurácia e tempo de provisionamento incorreto de sub e superprovisionamento e *jitter*, respectivamente, θ_U, θ_O, τ_U, τ_O e ρ. Para isso, é calculada a média geométrica da razão entre cada par de métricas. Matematicamente, a velocidade elastica para uma aplicação que utiliza provisionamento automático (a), baseada no cenário sem provisionamento automático (b), pode ser formulada pela Equação 2.7.

$$\epsilon_b = \left(\frac{\theta_{U,b}}{\theta_{U,a}} \cdot \frac{\theta_{O,b}}{\theta_{O,a}} \cdot \frac{\tau_{U,b}}{\tau_{U,a}} \cdot \frac{\tau_{O,b}}{\tau_{O,a}} \cdot \frac{\rho b}{\rho_a} \right)^{\frac{1}{5}}. \tag{2.7}$$

Tempo de resposta (Orientada ao usuário)

Mede a latência entre o momento em que uma solicitação é feita pelo usuário (ou sistema) e o momento em que a resposta é recebida. É uma métrica crítica para avaliar a performance de uma aplicação, já que tem um impacto direto na experiência do usuário. Em ambientes de alta demanda, um tempo de resposta elevado pode indicar gargalos no sistema ou falta de recursos apropriados [1].

Dentre as métricas apresentadas anteriormente, acurácia, ADI e tempo de provisionamento incorreto foram selecionados para uso neste trabalho.

2.4 Avaliação de Configurações de políticas de provisionamento automático

A avaliação da configuração da política de provisionamento automático é essencial, independentemente de ser uma abordagem reativa, proativa ou preditiva. O objetivo primordial desta avaliação é verificar a eficácia da configuração fornecida. Esta análise não é apenas uma etapa técnica, mas uma atividade fundamental para garantir o bom funcionamento do sistema. Uma avaliação precisa e cuidadosa é crucial para o sucesso da política de provisionamento [26].

Configurações inadequadas na política de provisionamento podem ter consequências sig-

nificativas. Por um lado, a alocação desnecessária de recursos leva ao desperdício de dinheiro. Por outro lado, a falta de recursos necessários pode comprometer severamente o desempenho da aplicação. Portanto, uma configuração precisa é necessária para equilibrar eficientemente o uso de recursos e a performance do sistema.

Como mencionado na seção 2.3, é crucial utilizar métricas específicas para avaliar a política de provisionamento automático. Estas métricas revelam o desempenho do sistema e sua resposta às demandas, permitindo verificar se a configuração atual atende aos objetivos de eficiência operacional, desempenho ideal, uso eficiente dos recursos e confiabilidade do sistema.

No entanto, encontrar uma configuração adequada é um desafio, visto que a demanda de uma aplicação é variável e as configurações ideais podem variar de acordo com demandas específicas de cada aplicação. Além do que, é importante verificar se a configuração atual permite que o sistema se expanda de forma eficiente, acompanhando o aumento da demanda. Isso pode incluir a adição de novas instâncias para distribuir a carga, o tempo requerido pela infraestrutura de nuvem para alocar e o balanceamento de carga para garantir que cada instância esteja sendo utilizada de maneira adequada[17].

Os experimentos comparativos são essenciais para avaliar e identificar a configuração mais adequada para a aplicação. Nesses experimentos, diferentes configurações são testadas e comparadas em termos de desempenho e eficiência. Essa abordagem ajuda a determinar a configuração que melhor se adapta ao resultado desejado.

Também é importante destacar a importância do acompanhamento contínuo da configuração do sistema. As condições e demandas podem mudar ao longo do tempo, tornando necessária um monitoramento regular das métricas de desempenho e avaliações periódicas da configuração. Isso permite identificar eventuais gargalos e ajustar parâmetros. Um acompanhamento contínuo garante que a configuração permaneça adequada às necessidades do sistema e dos usuários, garantindo sua eficiência e eficácia ao longo do tempo.

Dessa forma, é crucial que os profissionais que trabalham com alocação de recursos para as aplicações obtenham informações confiáveis e explicáveis sobre uma configuração de provisionamento automático [24]. Isso auxilia na tomada de decisão, aumentando a possibilidade de que as configurações escolhidas atendam às necessidades da aplicação e alcancem os SLOs desejados. Além disso, informações fundamentadas auxiliam na melhor compreensão

e transparência do processo de alocação de recursos, permitindo ao operador do provisionamento evoluir e adaptar as política de provisionamento com base nas métricas observadas.

Capítulo 3

Trabalhos Relacionados

Neste capítulo apresentamos o estado da arte relacionado a avaliação de políticas de provisionamento automático. Nesse contexto, foram selecionados trabalhos que definem métricas avaliativas, propõem estratégias de provisionamento e avaliações dessas. Nas seções subsequentes, destacamos as semelhanças e diferenças entre os estudos analisados e a presente pesquisa.

3.1 Métricas

Nas ações de provisionamento automático as métricas desempenham um papel fundamental, visto que são utilizadas tanto para guiar como para avaliar o emprego das ações de provisionamento. Diante disso, diversos autores abordam em seus trabalhos a criação de novas métricas, para que possam ser utilizadas nesse contexto de provisionamento automático.

No trabalho de Netto et al. [17], de forma semelhante a este estudo, os autores ressaltam a importância de configurar adequadamente as políticas de provisionamento automático, com base nas características da carga de trabalho do usuário, para evitar a degradação da Qualidade de Serviço (QoS) e o desperdício de recursos. Tendo como base essa problemática, a metodologia adotada por esses autores, que utiliza dados reais de um cluster do Google, proporciona uma abordagem única. Eles definem a métrica avaliativa denominada ADI (do inglês, *Auto-scaling Demand Index*), a qual consiste na soma das distâncias entre o limiar de utilização real e o desejado. Adcionalmente também discuti sobre como diferentes configurações de alocação ou desalocação de recursos e momentos de ativação do provisionamento

automático impactam diferentes tipos de cargas de trabalho. O trabalho de Netto et al. serve como uma referência fundamental para a nossa pesquisa, ajudando a moldar nossa abordagem e aprofundar nossa análise sobre configurações eficazes de provisionamento automático.

Herbst et al. [9] e Bauer et al. [6] argumentam a necessidade de repensar as métricas tradicionais de avaliação de ações de provisionamento automático e propor novas métricas específicas para nuvens. Então, oferecem contribuições importantes para o campo do provisionamento automático em nuvens, destacando a importância de métricas como acurácia, tempo de provisionamento e jitter. Abrindo caminho para uma compreensão mais profunda de como avaliar e otimizar estratégias de provisionamento

Essas métricas são utilizadas em diversos trabalhos que possuem o objetivo de avaliar políticas de provisionamento automático [24; 10; 21]. Assim como os estudos apresentados nessa seção, o presente trabalho utiliza as métricas já apresentadas na literatura e apresenta também uma nova métrica, discutida na seção 4.4.1, a qual têm o objetivo de reportar a porcentagem de requisições que são atendidas pelo sistema em relação a capacidade e para isso a capacidade foi calculada através de uma regressão linear múltipla. Em todos os estudos citados nessa seção, a métrica que relata a utilização de CPU é a responsável por disparar as ações de provisionamento automático, uma característica também observada neste estudo atual. No entanto outras métricas poderiam ter sido utilizadas, como a utilização de memória, largura de banda de rede, tempo de resposta do sistema, taxa de transferência de I/O de disco, quantidade de solicitações simultâneas, entre outros.

3.2 Políticas de provisionamento automático

As políticas de provisionamento automático, também conhecidas como políticas, desempenham um papel crucial no planejamento da quantidade de recursos que serão alocados para uma determinada aplicação. Este planejamento pode variar em escala de tempo, abrangendo prazos mais extensos, como um ano ou mais, até períodos mais curtos, como dias, horas ou minutos [15; 16; 30]. Nesse estudo, as políticas utilizadas são focadas em um planejamento de curto prazo, que decide a quantidade de recursos que devem ser alocados para os próximos minutos.

Também é importante destacar a natureza das políticas de provisionamento, que podem

ser classificadas como reativas, proativas ou preditivas. Os trabalhos de Moreno-Vozmediano et al. [16], Novak et al [19], Netto et al. [17] e Baarzi et al. [3] utilizam políticas reativas para ajustar a alocação de recursos em tempo real, baseando-se em métricas que monitoram as condições atuais do sistema, utilização de CPU, tempo de CPU por processo, contagem total de threads, entre outras.

Em contraste com o presente estudo, que utiliza políticas reativas Syu e Wang [27], Yadav et al. [29] e Kumar e Singh [14] utilizam políticas proativas para tentar prever as variações na demanda antes que ocorram, utilizando dados históricos e análises de tendências. Esses estudos são particularmente notáveis por sua capacidade de prever demandas futuras com precisão, o que representa uma contribuição significativa para a otimização de recursos.

Dentre as políticas reativas, a técnica introduzida por Baarzi et al. [3] é um exemplo prático, utilizando uma combinação de instâncias regulares e *burstable* [1] para ajustes dinâmicos baseados na análise da carga de trabalho, visando economia de custos e desempenho. Da mesma forma, a proposta de Netto et al. [17] é outro exemplo de política reativa, onde a quantidade de instâncias a serem adicionadas ou removidas é determinada em cada ação de provisionamento automático, podendo ser pré-definida, configurada, ou adaptativa, conforme a utilização atual dos recursos. Nesse trabalho as políticas utilizadas são de natureza reativa, alterando a quantidade de recursos disponíveis para a aplicação de acordo com o nível de utilização de CPU.

3.3 Avaliação do provisionamento automático

Com o surgimento de inúmeras políticas de provisionamento automático nos últimos anos, tornou-se evidente a necessidade de avaliá-las. Nesse contexto, o trabalho de Papadoulos et al. [20] destaca-se, pois oferece uma estrutura que proporciona garantias probabilísticas sobre a Qualidade de Serviço (QoS) alcançada pelos sistemas responsáveis por realizar o provisionamento automático. Além disso, os autores enfatizam a importância de uma avaliação mais detalhada das técnicas de provisionamento automático apresentadas na literatura, visto que existe uma tendência de generalização que pode não refletir a eficácia real dessas estratégias.

[1] Explicar o que siginifica

Ilyushkin et al. [12] desenvolvem uma abordagem metodológica que visa a avaliação e comparação de políticas de provisionamento automático, destacando métricas pertinentes para a análise do desempenho dessas estratégias. Eles analisam tanto as soluções de provisionamento automatizado desenhadas especificamente para aplicações particulares quanto aquelas de natureza mais genérica, sem se aprofundar como as configurações individuais impactam o desempenho específico de cada aplicação. Em contraste com nosso estudo, que busca entender em profundidade como diferentes configurações afetam o desempenho de aplicações específicas. Mas de forma semelhante ao nosso as comparações são realizadas através de métricas avaliativas encontradas na literatura.

Straesser et al. [25] também introduzem uma abordagem metodológica para a avaliação e configuração de políticas de provisionamento. Eles implementaram um escalonador reativo simples e ajustaram os padrões de dimensionamento da carga de trabalho multiplicando todos os registros por uma determinada proporção, para testar o escalonador com grandes cargas. Este estudo busca identificar indícios de configurações inadequadas, analisando se o comportamento real corresponde ao esperado, e oferece uma análise quantitativa com base nas métricas discutidas na literatura. Diversos outros autores [2; 6; 9; 17] baseiam-se em métricas presentes na literatura para avaliar se o sistema está provisionando recursos de forma eficiente e atendendo às demandas de QoS.

Os estudos focados na avaliação do provisionamento automático são mais próximos com a abordagem explorada nesta dissertação. Embora esses estudos frequentemente difiram em termos de metodologia e no tipo de resultado produzido, é notável que eles possuem, em sua maioria, o mesmo objetivo geral: a avaliação dos processos de provisionamento automático. Uma das distinções reside na metodologia adotada, com uma das principais diferenças sendo a escolha da carga de trabalho. Muitos desses estudos utilizam cargas de trabalho sintéticas, reduzidas ou tradicionais, que podem não refletir adequadamente a realidade operacional. Em contraste, neste trabalho, optamos por empregar uma carga de trabalho real e, com base nela, conduzimos simulações para avaliar o comportamento e desempenho de política de provisionamento automático.

Observamos também que, frequentemente, os estudos na área falham em fornecer detalhamentos suficientes sobre o desenvolvimento de sua metodologia, o que pode ser um obstáculo para a replicação e validação dos resultados por outros pesquisadores. Em contraste,

nosso estudo prioriza uma metodologia detalhada, facilitando que cada passo do processo de criação seja acessível. Isso não apenas facilita a verificação e reprodução do nosso trabalho por outros na comunidade científica, mas também contribui para a transparência da pesquisa.

Capítulo 4

Metodologia

Neste capítulo, detalhamos a abordagem metodológica adotada para este estudo. Na Seção 4.1, delineamos a natureza do estudo, incluindo seus objetivos, características e abordagens específicas. A Seção 4.2 é dedicada à descrição dos dados empregados na pesquisa. Posteriormente, na Seção 4.3, apresentamos uma visão aprofundada sobre o simulador utilizado para replicar as ações de provisionamento. Na Seção 4.4, discutimos as métricas selecionadas para avaliação e análise neste estudo, esclarecendo sua relevância e aplicação no contexto da pesquisa. E por fim, a Seção 4.5 apresenta como foram feitas as simulações e análises dos resultados.

4.1 Caracterização do Estudo

O presente estudo é caracterizado por possuir **natureza primária**, uma vez que se dedica a gerar informações inéditas através da análise, avaliação e comparação de distintas estratégias de configuração de provisionamento automático. Além disso, o trabalho apresenta uma contribuição importante ao propor um fluxo que auxilia na realização de análises comparativas entre as diversas configurações.

No que diz respeito aos objetivos, a pesquisa se destaca por possuir uma natureza **descritiva**, ao buscar descrever as diferenças de desempenho entre as configurações. Além disso, um caráter **exploratório** é inerente ao estudo, visto que ele se empenha em investigar a relação da carga de trabalho e da quantidade de recursos alocados diante das variadas estratégias de provisionamento automático. Ao introduzir um artefato destinado à análise comparativa,

a pesquisa revela sua abordagem **aplicada**.

Em relação aos procedimentos técnicos o trabalho adota uma abordagem de estudo de **caso exploratório**, uma vez que procura compreender mais as configurações e avaliações do provisionamento automático em um cenário simulado, através da **análise quantitativa**.

4.2 Dados

Os dados utilizados neste estudo são oriundos de uma multinacional tecnológica brasileira especializada em soluções de *e-commerce* baseadas em computação na nuvem. Enquanto a empresa oferece uma plataforma unificada no modelo SaaS (*Software as a Service*), para a criação e gestão de lojas online e o monitoramento da jornada de compra dos clientes, os dados analisados neste trabalho derivam de recursos de uma Infraestrutura como Serviço (IaaS).

Para a realização deste trabalho, recorremos a dados coletados em um estudo anterior [23], os quais foram disponibilizados em bucket no S3, um serviço de armazenamento de dados oferecido pela Amazon Web Services (AWS). No qual, um coletor foi desenvolvido para captar informações com uma granularidade temporal de um minuto, abrangendo recursos variados como instâncias de máquinas virtuais, grupos de provisionamento automático e balanceadores de carga. As tabelas 4.1, 4.2 e 4.3 descrevem os dados que foram selecionados para serem utilizados neste estudo.

Dado	**Descrição**
Timestamp	Data e horário da medição.
InstanceId	Identificador único da instância.
InstanceType	Conjunto de caracteristica associados a uma instância, consistem em várias combinações de CPU, memória, armazenamento e capacidade de rede.
Metric	Nome da métrica coletada (Como exemplo, quantidade de *cores* utilizados).
Average	Valor médio da utilização de CPU.

Tabela 4.1: Dados utilizados das instâncias de máquinas virtuais.

Dado	**Descrição**
Timestamp	Data e horário da medição.
InstanceId	Identificador único da instância.
LoadBalancer	Nome do identificador do *load balancer*.
Metric	Nome da métrica coletada (Nesse estudo só foi utilizada a quantidade de requisições).
Sum	Somatório da quantidade de requisições que chegam na aplicação.

Tabela 4.2: Dados utilizados dos balanceadores de carga.

Dado	**Descrição**
InstanceId	Identificador único da instância.
ApplicationName	Nome da aplicação.

Tabela 4.3: Dados utilizados dos grupos de provisionamento automático.

Temos à disposição dados que abrangem um período de quase três anos e são caracterizados pela sua natureza discretizada, uma vez que são coletados em intervalos regulares de 5 minutos. Inicialmente, realizamos o download refentes a um mês de dados, escolhido de forma arbitrária, que incluem as informações detalhadas nas tabelas 4.1, 4.2 e 4.3.

Posteriormente, realizamos uma filtragem baseada no nome da aplicação desejada. Essa escolha se deu pelo fato de que cada aplicação possui um comportamento distinto que demanda configurações específicas. Além disso, é importante destacar que essas aplicações são gerenciadas individualmente, com cada uma possuindo seu próprio sistema de provisionamento automático. Isso significa que cada aplicação pode ajustar seus recursos de forma independente, conforme a demanda.

Ao correlacionar informações das instâncias, balanceadores de carga e grupos de escalonamento automático, conseguimos extrair os dados necessários para nossa análise, que estão descritos na Tabela 4.4.

Para a análise, selecionamos duas aplicações da empresa em questão (designadas como Aplicações A e B). A Aplicação A foi selecionada devido ao fato de estar entre as sete responsáveis por gerar maiores despesas em relação a utilização de recursos. A Aplicação B

Dado	**Descrição**
Timestamp	Data e horário da medição.
Cores utilizados	Quantidade de *cores* utilizados.
Qnt. de requisições	Quantidade de requisições que a aplicação está recebendo em determinado momento.
Cores alocados	Quantidade de *cores* alocados para a aplicação.
Utilização do sistema	Utilização calculada a partir do número de *cores*.

Tabela 4.4: Dados gerados após o processamento das tabelas 4.1, 4.2 e 4.3. Com os quais foi possível calcular métricas e gerar gráficos.

foi escolhida arbitrariamente. As duas aplicações usadas neste estudo possuem as seguintes funções:

- **Aplicação A:** essa aplicação é responsável por gerenciar o catálogo de produtos.
- **Aplicação B:** possibilita a criação de entidades para armazenar dados, consultar diretamente da vitrine ou utilizá-las para armazenar informações para alguma integração externa.

4.3 Simulador

O simulador[1] foi desenvolvido com o propósito de replicar o ambiente de uma aplicação na nuvem e suas operações de provisionamento automático, incorpora duas estratégias disponibilizadas pela AWS: o Provisionamento Simples e o Provisionamento com Rastreamento de Meta. Estas são detalhadas na seção 2.2.

O simulador processa entradas que incluem o tempo, o tipo de instância utilizada e o número de vCPUs utilizados em determinado momento. Com base nessas informações, calcula-se a porcentagem de utilização dos recursos (vCPUs), levando em consideração que o tipo de instância determina a quantidade total de vCPUs disponíveis. Assim, é possível identificar tanto a quantidade de cores em uso quanto a quantidade total alocada dos dados passados como entrada.

[1] https://github.com/ufcg-lsd/autoscaling-analyser

Outra entrada que precisa ser fornecida para o simulador é a configuração da politica de provisionamento. A Figura 4.1 ilustra a implementação dessas políticas no simulador, seguindo às especificações documentadas pela AWS. As estratégias de provisionamento, simples e de rastreamento de meta, apresentam características operacionais distintas, embora algumas configurações sejam comuns a ambas.

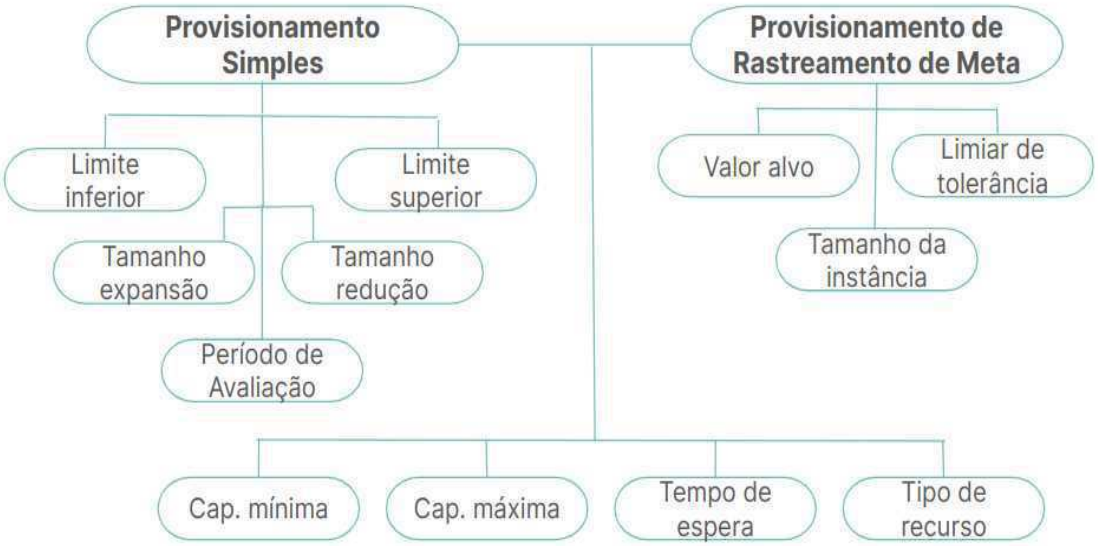

Figura 4.1: Parâmetros das políticas de provisionamento automático implementadas no simulador.

Na política de provisionamento simples, utilizamos os seguintes parâmetros:

- **Limite Inferior**: Representa o limite percentual mínimo de utilização de recursos. Quando a utilização está abaixo desse valor, o sistema inicia a redução de recursos, visando a eficiência e evitando desperdícios.

- **Limite Superior**: Define o limite percentual máximo para a utilização dos recursos. Se a utilização ultrapassa este valor, o sistema responde com o provisionamento adicional de recursos para satisfazer a demanda crescente.

- **Quantidade de *cores* em ações de expansão de recursos**: Estabelece a quantidade de *cores* a serem adicionados quando a utilização excede o limite superior.

- **Quantidade de *cores* em ações de redução de recursos**: Define a quantidade de *cores* a serem removidos quando a utilização cai abaixo do limite inferior. Uma re-

dução de 8 *cores* mostra que o sistema diminui a alocação em 8 *cores* a cada ação de escalonamento descendente.

- **Período de Avaliação**: Esse parâmetro decide a frequência de avaliação da utilização dos recursos pelo sistema, para determinar se ajustes são necessários.

Na política de **Provisionamento de Rastreamento de Meta**, utiliza-se os seguintes parâmetros:

- **Valor alvo**: Define o objetivo de utilização dos recursos, normalmente em percentual. Um valor de 70% significa que o sistema tenta manter a utilização em torno dessa marca, equilibrando eficiência e prontidão para picos de demanda.

- **Limiar de tolerância**: Estabelece o limiar de tolerância para a redução de recursos. Mesmo se a utilização cair abaixo do objetivo, o sistema possui essa margem de tolerância antes de iniciar a desalocação de recursos.

Além disso, algumas configurações são comuns a ambas as políticas:

- **Capacidade Mínima**: Garante um mínimo de *cores* disponíveis para a aplicação, independentemente da demanda atual.

- **Capacidade Máxima**: Limita a alocação máxima de *cores*, assegurando uma ampla disponibilidade de recursos para a aplicação.

- **Tempo de Espera**: Define um período de estabilização após cada ação de provisionamento, durante esse tempo o sistema não realiza novas ações para permitir que os efeitos da última mudança se concretizem.

- **Tipo de recurso**: Especifica a unidade de medida para o provisionamento, que neste estudo são os *cores*, ou núcleos de processamento. Este parâmetro determina como os recursos são alocados ou desalocados durante as operações de provisionamento.

Como resultado do processo de simulação, obtemos informações relacionadas à quantidade de recursos alocados ao longo da simulação. Além disso, temos o número de *cores*

alocados ou desalocados decididos pela política e a quantidade de tempo necessário até a próxima decisão de provisionamento. Este período de espera é acionado imediatamente após a ação de alocação ou desalocação de recursos, com base na configuração da política.

Considerando que, no cenário real, alguns fatores podem influenciar a quantidade de instâncias disponíveis, como a locação no mercado spot, falhas ou outras influências externas, é possível que haja variação na quantidade de instâncias disponibilizadas. Portanto, o desempenho do simulador foi avaliado, no Capítulo 5, utilizando a métrica de Erro Médio Absoluto (MAE) para comparar a quantidade de instâncias utilizadas entre o cenário real e o simulado. Assim, o cenário simulado será executado com as mesmas configurações do cenário real. Esta avaliação permitirá verificar a precisão e a confiabilidade do simulador em relação aos dados específicos deste estudo.

4.4 Métricas de avaliação das políticas

Para avaliar as políticas de provisionamento automático, utilizamos métricas operacionais, que são orientadas ao sistema, de avaliação e de custo. A primeira quantifica o provisionamento excessivo ou insuficiente e são baseadas na análise das curvas de oferta e demanda. Neste estudo, a demanda é entendida como a carga de trabalho imposta à aplicação, medida pela quantidade de *cores* necessários para processar essa carga e manter um nível de SLO aceitável, e a oferta é a quantidade de recursos disponibilizados pelo provedor [2].

A Figura 4.2 mostra um exemplo dessas duas curvas, demonstrando situações de super e sub-provisionamento, representadas pelas letras O e U, respectivamente. Neste estudo, a quantidade de vCPUs foi utilizada para quantificar a oferta e demanda. É importante destacar que, para fins deste estudo, a denominação *cores* foi considerada similar a vCPUs, uma vez que os dados coletados se referem especificamente à utilização dos cores.

Além disso, para calcular a quantidade de instâncias necessárias, assumimos uma utilização de 100% de cada instância. Embora essa não seja a abordagem ideal, essa suposição foi adotada para simplificar as análises.

Como discutido na Seção 2.3, várias métricas relevantes para este estudo foram examinadas. Algumas delas foram escolhidas para serem aplicadas nesta pesquisa, pois ajudam a compreender o comportamento dos recursos alocados, contribuindo assim para a definição

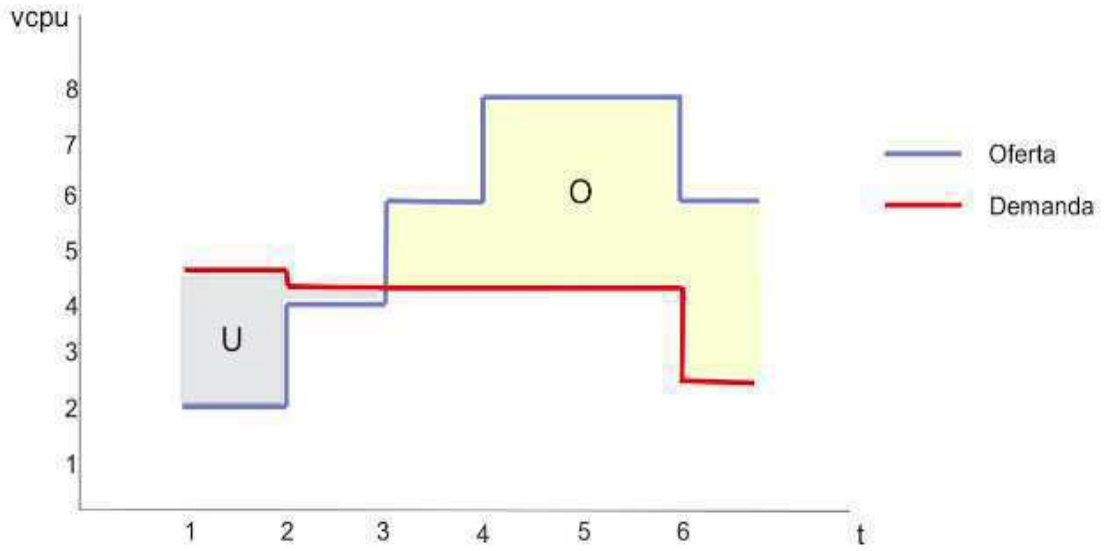

Figura 4.2: Curvas de oferta e demanda, ilustrando situações de sub e superprovisionamento representadas pelas letras U e O, respectivamente.

de como a política de provisionamento deve ser ajustada.

Foram selecionadas as métricas de acurácia, que revelam a quantidade de recursos estão sendo fornecidos além ou aquém da demanda, e o tempo de provisionamento incorreto, que aponta se o subprovisionamento ou superprovisionamento ocorreu durante períodos prolongados ou em intervalos breves. Adicionalmente, propomos a métrica de Taxa de Atendimento de Requisições (TAR), visto o desejo de quantificar quantas requisições estavam sendo atendidas e quantas a mais a aplicação conseguia atender. Também introduzimos métricas para estimar os custos e desperdícios.

4.4.1 Taxa de Atendimento de Requisições (TAR)

Essa métrica tem como objetivo reportar a porcentagem da média de requisições que são efetivamente atendidas pelo sistema em relação a capacidade. De forma mais clara, se uma aplicação teve uma demanda média de 5000 requisições, mas o sistema só possui recursos para atender em média 3000 requisições. Então, a Taxa de Atendimento de Requisições para essa aplicação foi de 166%, indicando que 66% da demanda deixou de ser atendida.

Para aprofundar nossa análise e entender melhor como diferentes fatores influenciam a capacidade do sistema, recorremos à regressão linear múltipla apresentada na Equação

4.1. Esse método estatístico nos permite prever a capacidade do sistema (y) com base em múltiplos fatores, como os *cores* alocados e a utilização do sistema.

$$y = a + b \times \textit{cores} \text{ alocados} + c \times \text{utilização do sistema} \tag{4.1}$$

Nessa equação, *a* representa o valor de *y* quando não há *cores* alocados nem utilização do sistema. Os coeficientes *b* e *c* indicam como a capacidade do sistema é afetada por cada unidade adicional dos *cores* alocados e da utilização do sistema, respectivamente. Essa abordagem nos permite quantificar a influência de cada variável na capacidade do sistema.

Escolhemos a regressão linear múltipla pela sua capacidade de analisar simultaneamente o efeito de várias variáveis. Isso nos dá uma visão mais completa e precisa de como otimizar nossos recursos. Contudo, nossa análise parte da suposição de que todas as requisições demandam a mesma quantidade de recursos, o que simplifica o modelo, mas introduz uma limitação, pois na prática, diferentes requisições podem ter pesos diferentes. Reconhecemos essa limitação e consideramos, para análises futuras, ajustar nosso modelo para refletir a diversidade nas cargas das requisições. Mas dada a natureza dos dados disponíveis, essa abordagem se faz necessária, pois não temos a capacidade de diferenciar as requisições de forma individual.

Uma vez que é calculado quantidade de requisições que o sitema pode atender, para calcular o valor da taxa de atendimento de requisições utiliza-se a equação 4.2. Onde R_a e R_s representam o número de requisições que o sistema pode atender e a quantidade total de requisições que chega ao sistema, respectivamente.

$$TAR = \left(\frac{R_s}{R_a}\right) \times 100 \tag{4.2}$$

A TAR nos fornece uma métrica direta para avaliar a eficácia com que um sistema atende às requisições dentro de sua capacidade atual. Por exemplo, uma TAR de 60% indica que o sistema está utilizando apenas 60% de sua capacidade total para atender às requisições, sugerindo que há um espaço adicional de 40% disponível para lidar com mais demanda. Em contraste, uma TAR de 120% revela que o sistema está sobrecarregado, atendendo 20% acima de sua capacidade, o que implica que algumas requisições não estão sendo atendidas eficientemente.

Com base na métrica TAR, os SREs podem fazer planejamentos mais precisos sobre ajustes na alocação de recursos. Se a métrica indica espaço disponível significativo, pode ser prematuro investir em aumento da infraestrutura; por outro lado, se o valor for alto pode justificar investimentos em capacidade adicional.

4.4.2 Porcentagem de desperdício estimado

Na AWS, existem várias formas de adquirir instâncias, incluindo as opções sob demanda, reservadas ou spot, cada uma com sua própria estrutura de precificação. No entanto, os dados utilizados neste estudo não incluem informações sobre o preço ou o método específico de aquisição das instâncias. Diante dessa lacuna, decidimos basear nossos cálculos no modelo sob demanda, que se caracteriza pela sua simplicidade e pela ausência de compromissos de longo prazo. Embora essa escolha seja razoável para os objetivos de nossa análise, é importante reconhecer que ela pode não refletir a opção mais econômica disponível.

Os preços das instâncias AWS sob demanda são normalmente cotados por hora. No entanto, nossos dados estavam na granularidade de minutos. Para contornar essa limitação, adaptamos nossa metodologia de cálculo para se adequar à estrutura dos dados disponíveis. Primeiro, determinamos o custo por minuto de uma instância, dividindo o custo horário pelo número de minutos em uma hora.

Para estimar o custo total, começamos totalizando o tempo de utilização de todas as instâncias, medido em minutos. Em seguida, multiplicamos este total de minutos pela quantidade de instâncias disponíveis em cada intervalo de tempo. O resultado dessa multiplicação é então multiplicado pelo custo por minuto de cada instância. Quanto à estimativa de desperdício, seguimos um procedimento semelhante: utilizamos a mesma fórmula, mas, em vez de multiplicar pelo número total de instâncias disponíveis, multiplicamos pelo número de instâncias excedentes.

Após a estimativa do custo total e do custo associado às instâncias excedentes, procedemos ao cálculo da porcentagem de desperdício de uma maneira que permite uma interpretação direta e intuitiva dos resultados. Esse cálculo é fundamentado na relação entre o custo total do uso das instâncias (considerado como 100% do custo) e o custo atribuído ao uso excessivo ou não otimizado dessas instâncias, identificado como desperdício.

Para quantificar a porcentagem de desperdício, foi utilizada a Equação 4.3.

$$\text{Porcentagem de Desperdício} = \left(\frac{\text{Custo do Desperdício}}{\text{Custo Total}}\right) \times 100 \tag{4.3}$$

Por exemplo, se o custo total para operar as instâncias for de 8000 unidades monetárias em um determinado período, e identificarmos que 4000 unidades desse total correspondem ao custo de instâncias utilizadas de forma ineficiente (ou seja, as instâncias excedentes), então foi calculada a porcentagem de desperdício como mostra a Equação 4.4.

$$\text{Porcentagem de Desperdício} = \left(\frac{4000}{8000}\right) \times 100 = 50\% \tag{4.4}$$

É importante salientar que este estudo baseou-se nos preços das instâncias da AWS válidos em 2 de janeiro de 2024. Para a Aplicação A e B, o custo da instância computacional utilizada foi de $ 0.289 e $ 0.0765 por hora.

4.5 Simulação de Configurações e Análises

A fase de análise e simulação desempenha um papel essencial no processo de otimização de uma escolha adequada de configuração. Esta etapa se distingue pelas contínuas execuções do simulador e pela avaliação dos resultados obtidos. O ponto de partida desse processo envolve a execução do simulador com base em uma configuração preestabelecida, fornecida ao utilizar o simulador, proporcionando uma compreensão do comportamento do sistema quando submetido a determinada configuração de provisionamento automático.

Para cada aplicação foi realizada uma simulação base com a configuração usada pela empresa e uma avaliação detalhada dos resultados. Esta análise foi embasada nas métricas citadas na subseção 4.4 e foi complementada por visualizações gráficas. O objetivo dessa etapa era entender a dinâmica da demanda e identificar eventuais gargalos ou insuficiências que oferecessem oportunidades de melhoria.

Posteriormente, com base na avaliação do comportamento da demanda em relação à configuração inicial, foi possível extrair indicações sobre quais parâmetros poderiam ter exercido maior influência na capacidade de lidar com a demanda. Dessa forma, após análises e discussões, que serão apresentadas nos próximos capítulos, optou-se por ajustar certos parâmetros. Neste ponto, já se tinha uma hipótese formada: a alteração de um determinado parâmetro deveria resultar em um comportamento específico. Assim, após implementar as mudanças

inferidas, o simulador foi novamente executado. A análise subsequente visou verificar se as expectativas iniciais foram confirmadas e se os ajustes produziram os efeitos antecipados.

Ao comparar os resultados provenientes de diferentes configurações, o estudo também testa a sensibilidade da ferramenta. Se diferentes configurações podem de fato produzir resultados distintos e esperados. Isso corrobora a capacidade da ferramenta de responder com precisão a variações nos parâmetros e das métricas em capturar o comportamento da aplicação.

Capítulo 5

Validação do Simulador

Para validar o simulador, conduziu-se uma análise comparativa da quantidade de *cores* alocados entre os dados reais e os gerados pela simulação. Para gerar essa simulação, utilizou-se a mesma configuração empregada pela empresa para provisionar recursos para a aplicação, visando uma comparação mais justa.

O mesmo experimento foi realizado para as duas aplicações em questão, o que enriquece a validação, visto que as políticas de provisionamento utilizadas são diferentes. A aplicação A utiliza provisionamento simples, enquanto a aplicação B utiliza provisionamento com rastreamento de meta.

5.1 Aplicação A

A Figura 5.1 mostra qual foi a diferença entre a utilização de cores no cenário real e simulado. Apresentando variações diárias e picos que contribuem para essa diferença. A média da diferença, entre a quantidade de recursos alocados no cenário real foi de aproximadamente 15% cores a mais em relação a simulação. Como essa aplicação utiliza instâncias de 8 *cores*, esse resultado equivale a 9 instâncias.

Mas para obtermos uma avaliação quantitativa mais robusta desta disparidade utilizamos a métrica de Erro Absoluto Médio (MAE).

Utilizamos também a métrica de Erro Absoluto Médio (MAE) para obtermos uma avaliação mais robusta. Essa métrica indicou uma divergência média de aproximadamente 80 *cores*. Considerando que as instâncias disponibilizadas para essa aplicação contém 8 vCPUs,

esta margem de erro equivale a 10 instâncias em média. Esta diferença, apesar de presente, é relativamente pequena se considerarmos o volume total de recursos alocados (na casa de 5 milhões de *cores*).

Esses resultado sugerem que o simulador oferece uma representação bastante satisfatória da realidade, o que comprova sua confiabilidade para uso em projeções e análises de alocação de recursos.

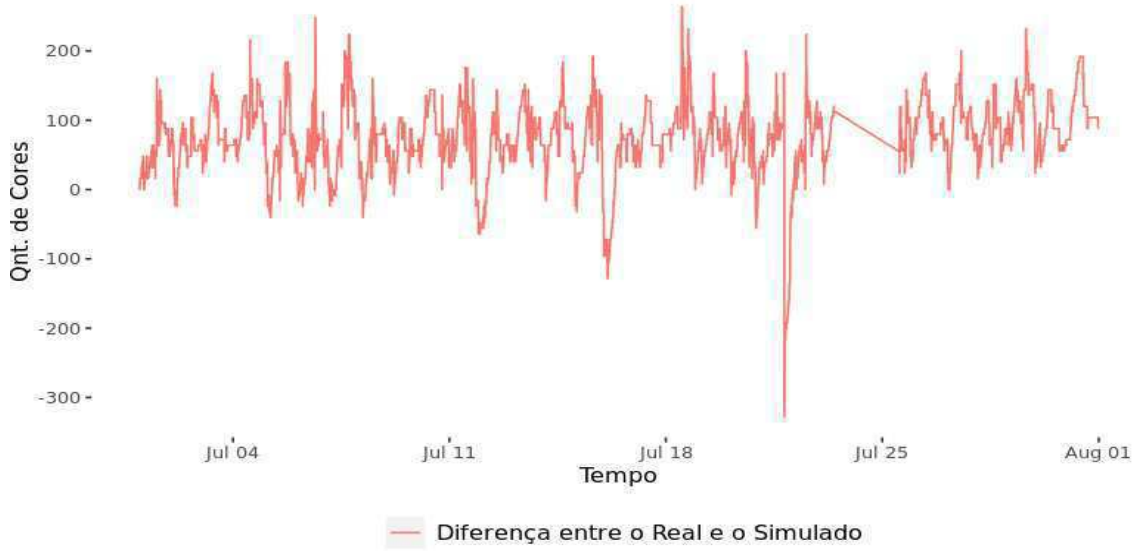

Figura 5.1: Variação temporal da diferença entre a utilização de *cores* no dados reais e simulados, utilizando os dados da Aplicação A.

5.2 Aplicação B

Também avaliamos o comportamento do simulador em relação aos *cores* alocados utilizando os dados da Aplicação B.

A Figura 5.2 complementa o entendimento sobre a diferença de *cores* alocados no dois cenários. A média da diferença, em cada instante, entre a quantidade de recursos alocados no cenário real foi de aproximadamente 11% a mais em relação a simulação. Como essa aplicação utiliza instâncias de 2 *cores*, esse resultado equivale a 12 instâncias. Embora a simulação não tenha alcançado uma correspondência exata, ela conseguiu replicar o comportamento de alocação de recursos da aplicação de forma bastante próxima. O que pode

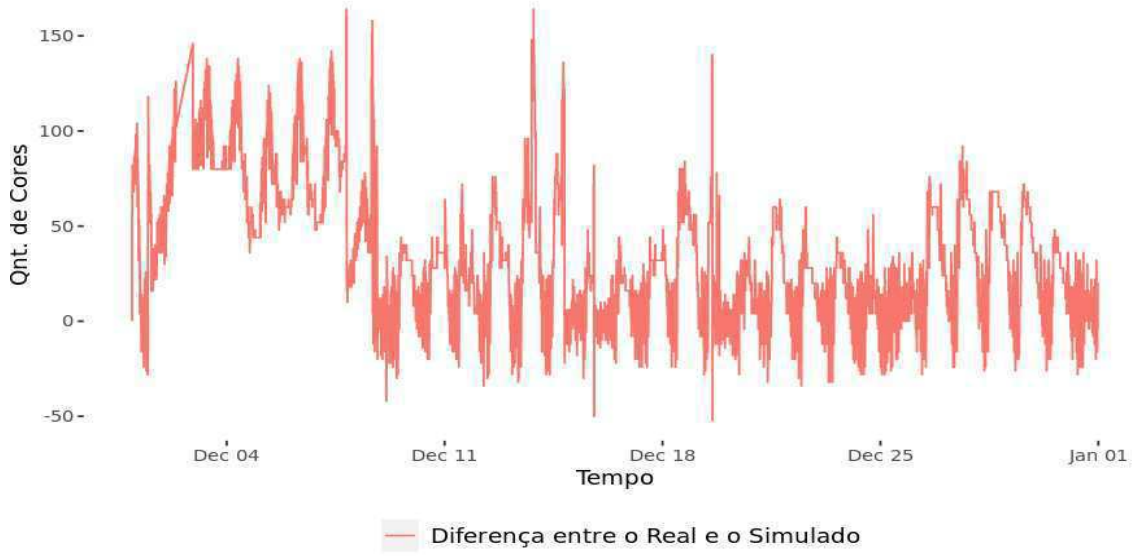

Figura 5.2: Variação temporal da diferença entre a utilização de *cores* no dados reais e simulados, utilizando os dados da Aplicação B.

ser considerado uma variação relativamente estreita, especialmente quando comparada com a quantidade total de cores envolvidas.

A métrica MAE revelou uma discrepância média bastante pequena de aproximadamente 36 *cores*. Ao colocar isso em perspectiva com a infraestrutura de computação utilizada, onde cada instância comporta 2 vCPUs, a divergência representa apenas 18 instâncias. Tal precisão sugere que o modelo de simulação está alinhado de com as operações reais.

Capítulo 6

Análise da configuração da empresa

Neste capítulo, realizamos uma análise detalhada da demanda e da capacidade de processamento, focando em aspectos como *cores* de processamento e volume de requisições. A eficácia na alocação de recursos foi mensurada utilizando métricas específicas de eficiência e custo. Realizamos esse procedimento para as duas aplicações em questão.

6.1 Aplicação A

Os dados utilizados dessa aplicação são referentes ao mês de Julho de 2023. Foi possível verificar que a configuração consiste em uma política reativa de provisionamento simples e utiliza valores próximos entre os limites superior e inferior.

6.1.1 Parâmetros de configuração de provisionamento

Para a operação desta aplicação, a empresa adota a configuração de provisionamento descrita na Tabela 6.1, ao longo do estudo chamaremos de **Configuração Base.**

Na configuração estabelecida pela empresa não foi definido um parâmetro para a capacidade máxima de recursos que podem ser alocados. Por isso, adotamos um valor que consideramos suficientemente elevado para evitar que a limitação de recursos influenciasse os experimentos.

Política	Provisionamento Simples
Limite superior	65%
Limite inferior	55%
Incremento de recursos	32 *cores*
Decremento de recursos	8 *cores*
Capacidade mínima	32 *cores*
Capacidade máxima	240000 *cores*
Período de avaliação	2 min
Tempo de espera	3 min

Tabela 6.1: Parâmetros definidos para a Aplicação A durante o mês de análise dos dados. Exceto a capacidade máxima, os demais valores são os mesmos utilizados pela empresa para essa aplicação no mês de Julho.

6.1.2 Análise da utilização de *cores*

A Figura 6.1 apresenta a utilização de *cores* ao longo do mês de Julho, destacando uma alta oscilação, o que sugere variações na carga de trabalho da aplicação. A existência de picos e quedas acentuadas são considerados pontos fora da curva, possivelmente devido a eventos específicos ou processos que requerem um aumento ou redução na utilização de recursos. O gráfico ainda mostra que a contagem de *cores* varia entre aproximadamente 100 e 300. Então, para atender a demanda da aplicação sem comprometer o desempenho, o sistema precisa adaptar a quantidade de recursos de acordo com essa variação que existe na demanda.

Ainda é possível observar que há uma tendência de aumento ou diminuição na utilização ao longo do mês, indicando que a carga geral permanece relativamente constante em termos de picos e vales. Para compreender melhor esse comportamento, analisamos os valores agrupados por hora do dia na Figura 6.2. Essa visualização apresenta o gráfico de caixa da quantidade de *cores* utilizados a cada hora, onde é possível observar uma variação ao longo do dia. Esse gráfico foi elaborada com o objetivo de determinar se existe um padrão nos horários em que os picos e vales de utilização são mais frequentes.

Analisando a distribuição da utilização de *cores* por hora do dia, Figura 6.2, é perceptível

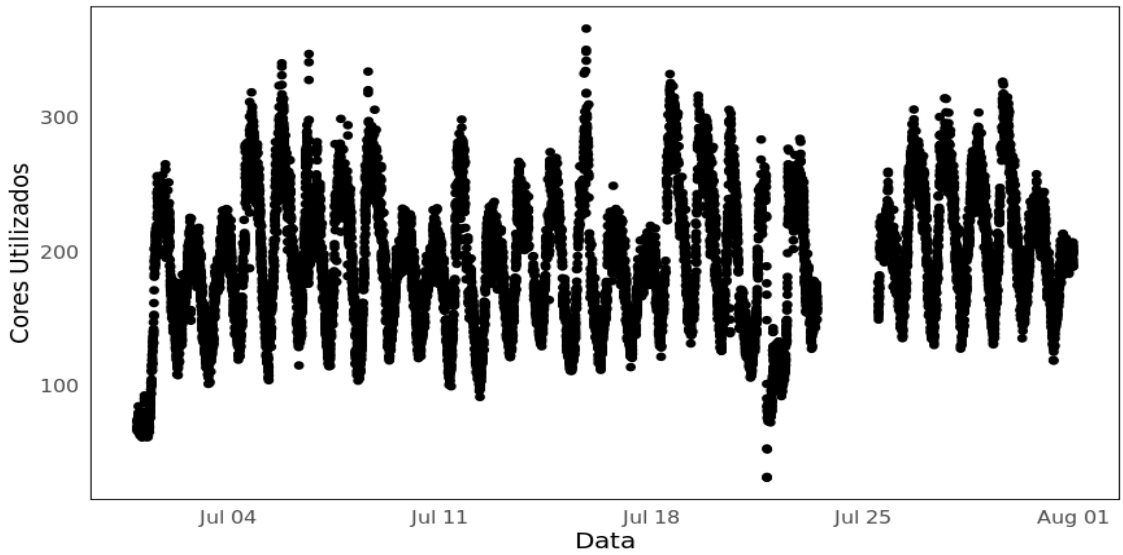

Figura 6.1: Variação diária no uso de *cores* em Julho da Aplicação A, indicando uma oscilação entre aproximadamente 100 e 300.

que durante a madrugada e as primeiras horas da manhã, as medianas estão posicionadas mais baixo e as caixas são menores, indicando uma menor utilização e variação na utilização de recursos. Isso mostra que a demanda que chega nessa aplicação é menor nesse período, o que poderia ser esperado, pois geralmente corresponde a horas fora do horário comercial.

Por outro lado, durante a tarde, observa-se que as medianas se elevam e os boxplots se tornam maiores, refletindo um aumento significativo na utilização de recursos e na variação. Isso pode ser atribuído a uma maior utilização da aplicação, como pode ocorrer em horários de pico de trabalho ou quando tarefas computacionais intensivas estão agendadas para serem executadas. Essas observações apontam para uma necessidade de garantir a disponibilidade de recursos de processamento adequados durante as horas de maior demanda e, possivelmente, uma oportunidade de reduzir a capacidade quando a utilização for menor.

6.1.3 Avaliação da alocação de recursos

Após analisarmos o comportamento da utilização dos *cores*, torna-se importante entender como a alocação de recursos está sendo gerenciada para essa aplicação, a fim de identificar possíveis ocorrências de subprovisionamento ou superprovisionamento.

Portanto, a Figura 6.3 ilustra a comparação entre as instâncias disponíveis e as instâncias

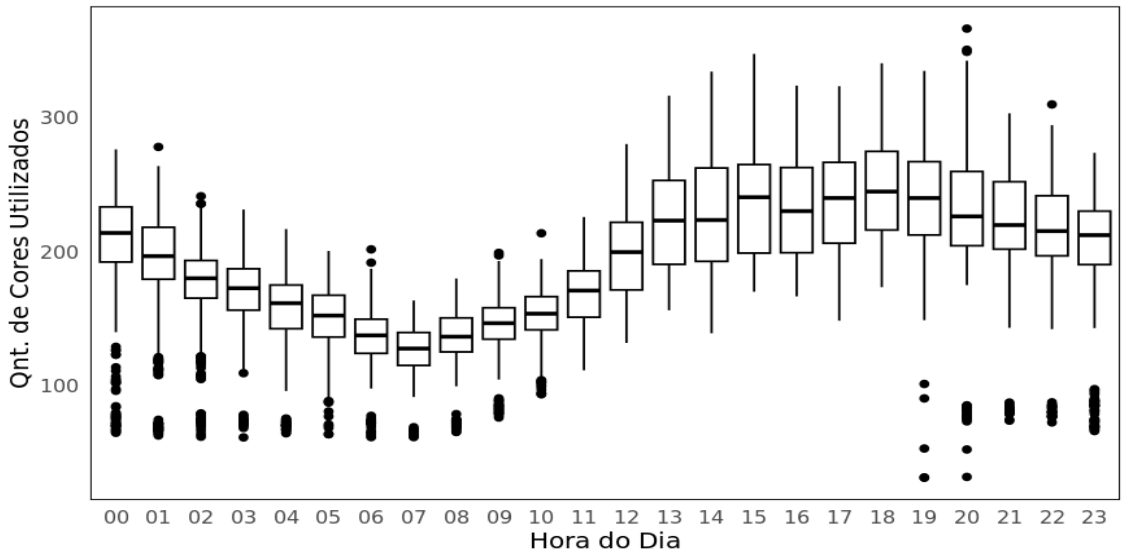

Figura 6.2: Distribuição do uso de *cores* durante o mês de Julho na Aplicação A, com os dados organizados por hora, evidenciando o padrão de uso em cada período do dia.

necessárias. É possível observar os momentos em que ocorrem descompassos entre a oferta e a demanda de recursos ao longo do período analisado. Observa-se que a linha vermelha, representando as instâncias disponíveis, frequentemente supera a linha azul, que indica as instâncias necessárias, sugerindo períodos de superprovisionamento.

Para calcular as instâncias necessárias, consideramos uma utilização de 100% das instâncias, esse cálculo representa uma extrapolação, pois para alguns tipos de aplicações, valores de utilização nesse percentual não são aceitáveis ou práticos.

Ao mensurar o desperdício médio de recursos esse valor foi de 43.93%. O somatório da quantidade de instâncias que podem ser consideradas desnecessárias, ou seja, aquelas que permaneceram ociosas, em cada momento do tempo foi de 258083.9 e haviam disponíveis 586834. Esse resultado sugere que há uma margem de recursos disponíveis que ultrapassa o uso atual, o que, embora benéfico visto que pode acomodar picos de demanda, também pode sinalizar desperdício e uma oportunidade de redução de custos.

Ajustar a alocação de recursos para alinhá-la com os padrões de utilização poderia não só evitar o subprovisionamento, que poderia interromper as operações, mas também reduzir o superprovisionamento, reduzindo assim os custos com infraestrutura.

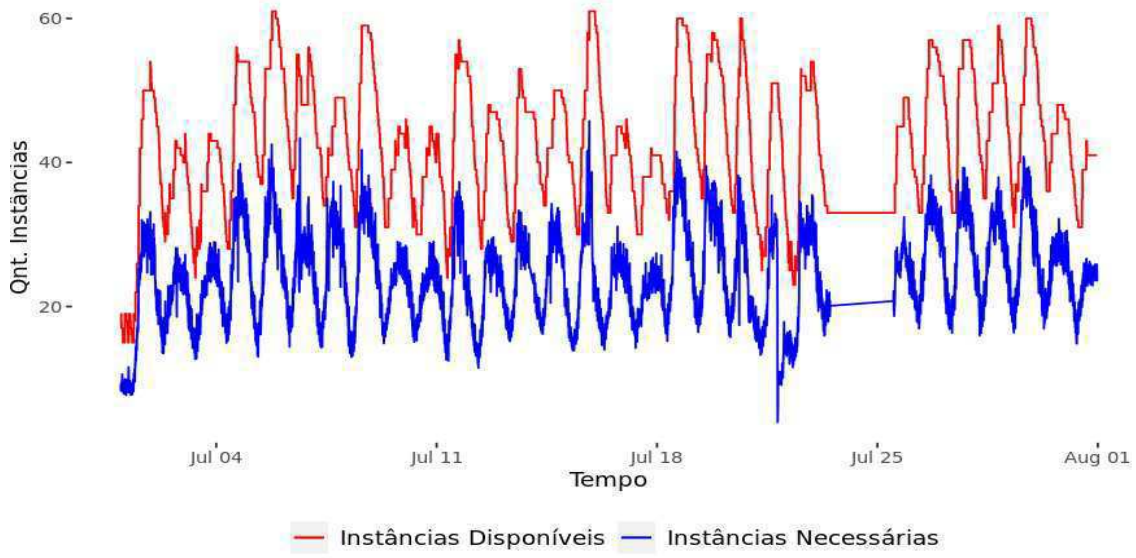

Figura 6.3: Comparativo diário das instâncias disponíveis em relação às necessárias em Julho da Aplicação A, mostrando que, predominantemente, a oferta de recursos supera a demanda.

6.1.4 Avaliação dos limiares que acionam ações de provisionamento

A Figura 6.4, complementa as análises realizadas anteriormente mostrando a utilização percentual do sistema em relação aos limiares estabelecidos para o provisionamento. A política de provisionamento está configurada para alocar mais recursos quando a utilização atinge 65%, e desaloca quando fica abaixo de 55%.

A utilização excede o limiar superior em apenas 9.75% do período analisado e permanece abaixo do limiar inferior em aproximadamente 39.3%, o que reforça a ideia de que há recursos alocados em excesso, considerando que foram necessárias mais ações para desprovisionar recursos.

Este comportamento indica que o sistema pode estar sendo excessivamente conservador ao alocar recursos adicionais, reforçando a hipótese de superprovisionamento sugerida anteriormente.

Nesse contexto, poderia ser considerado um ajuste desses limiares para uma melhor utilização dos recursos alocados. Por exemplo, aumentar ligeiramente o limiar de provisionamento para um valor acima de 65% poderia reduzir a frequência de alocação de novas instâncias, permitindo que o sistema utilize de maneira mais eficiente os recursos já disponíveis.

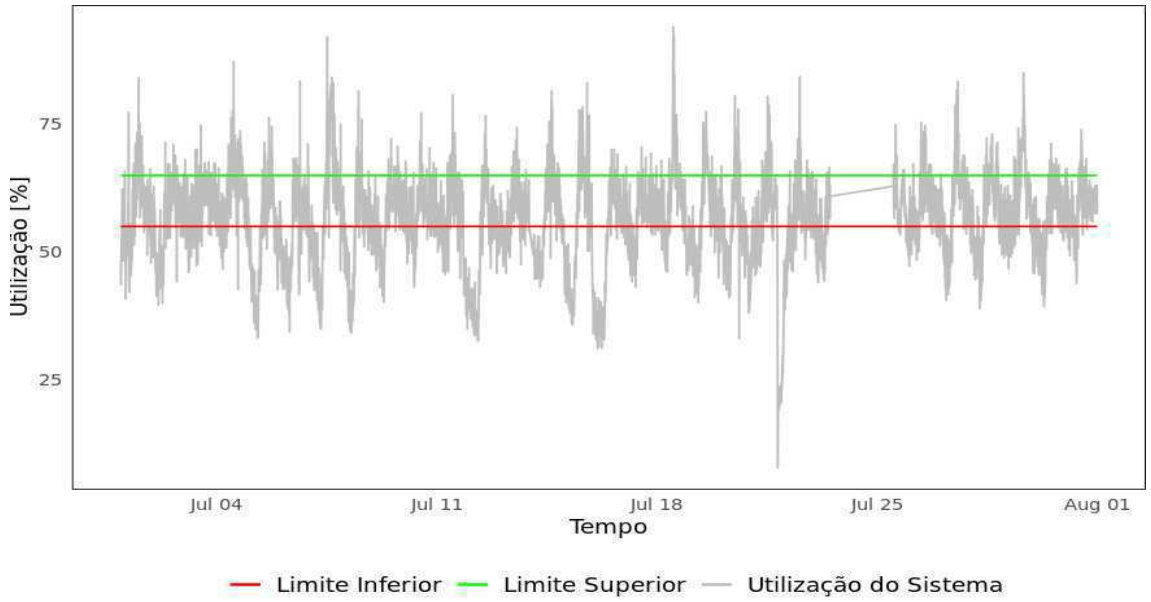

Figura 6.4: Monitoramento da utilização percentual do sistema em Julho.

6.1.5 Análise da capacidade de processamento baseado em requisições

Considerando a demanda em termos de número de requisições, realizamos uma análise das requisições que chegaram ao sistema e daquelas que não foram atendidas devido à limitação de capacidade. Com a suposição de que todas as requisições possuem um tamanho padrão, aplicamos um modelo de regressão linear múltipla com o objetivo de estabelecer a capacidade máxima de processamento do sistema. A formulação da capacidade estimada, obtida pela regressão, é expressa pela Equação 6.1, onde o valor atribuído à variável utilização, que foi de 100%, indica o uso completo do processamento disponível.

$$\text{Capacidade} = -696096.5 + 4265.679 \times \textit{Cores}\ \text{Alocados} + 22209.82 \times \text{utilização} \quad (6.1)$$

O modelo apresentou um coeficiente de determinação (R^2) de 0.7, indicando que 70% da variabilidade nas requisições recebidas é explicada pelas variáveis incluídas no modelo. Isso sugere uma correlação moderadamente forte entre as variáveis independentes e a capacidade do sistema.

Além do R^2, as métricas de Erro Absoluto Médio (MAE) e Raiz do Erro Quadrático Médio (RMSE) foram utilizadas para avaliar a precisão do modelo. O MAE obtido foi de 158861.9, o que é relativamente pequeno em comparação com a média de requisições, que

é de 2006472 (na casa de 2 milhões). Isso indica que o modelo tem, em média, uma boa precisão, errando por menos de 8% da média de requisições. O RMSE foi de 234163.9, também pequeno em relação à média de requisições, mas maior que o MAE, refletindo o impacto de alguns erros grandes nas previsões. Esses valores sugerem que, embora o modelo forneça previsões razoavelmente precisas, ainda há espaço para melhoria, principalmente na investigação e redução dos efeitos dos *outliers*, que podem estar contribuindo para o aumento do RMSE.

Na Figura 6.5 as linhas representam as séries temporais da quantidade de requisições recebidas (em vermelho) e a capacidade de requisições que o sistema consegue atender (em azul), com base na regressão. Em alguns momentos, como no início e meados de Julho, o volume de requisições excedeu a capacidade de processamento do sistema, sugerindo períodos de possível sobrecarga. Após calcular os valores, observamos que apenas 0.08% das requisições não foram atendidas, enquanto uma expressiva maioria de 99.92% das requisições foram atendidas com sucesso.

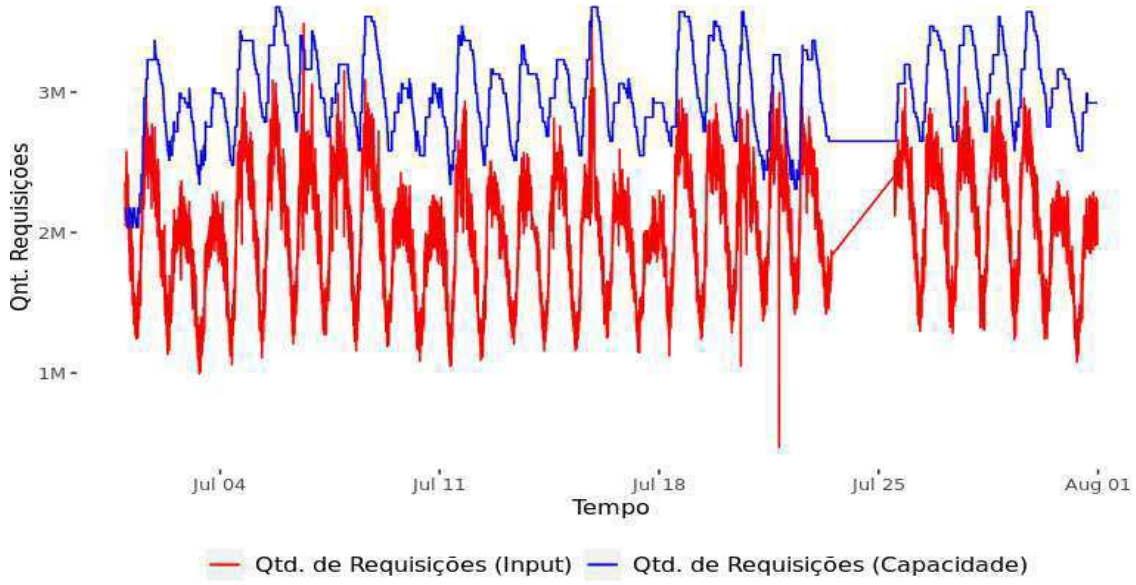

Figura 6.5: Análise temporal da capacidade de processamento de requisições.

6.1.6 Avaliação quantitativa de desempenho e eficiência de recursos

Para proporcionar uma análise mais precisa do desempenho da aplicação, quantificamos o comportamento da demanda e oferta de recursos através das métricas descritas nas Seções

2.3 e 4.4. Com a finalidade de avaliar tanto a eficiência quanto o custo-benefício da aplicação em questão.

Na Tabela 6.2 são apresentados os valores obtidos para a configuração de provisionamento automático analisada. Os resultados mostram que a configuração atual de provisionamento automático da aplicação está levando a um excesso dos recursos. Com um tempo de superprovisionamento de 100%, a aplicação está consistentemente operando com mais recursos do que o necessário, o que não é eficiente do ponto de vista de custo. Isso é evidenciado pela acurácia de superprovisionamento de 8341%, indicando uma disponibilidade de recursos muito acima do necessário.

Métrica	**Config. Base**
Acurácia Subprovisionamento	0%
Acurácia Superprovisionamento	8341%
ADI	5.88e+08
Tempo de subprovisionamento	0%
Tempo de superprovisionamento	100%
TAR	67.27%
jitter	-5.85
Custo estimado	9131.81 USD
Desperdício estimado	3961.14 USD

Tabela 6.2: Métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando a configuração estabelecida pela empresa com os dados da Aplicação A.

O ADI elevado (5.88e+08) reforça essa observação, sinalizando uma grande discrepância entre o uso de recursos e os níveis desejados de utilização.

A métrica TAR revela que 67.27% das requisições foram atendidas, indicando que existe pontencial para que redução da quantidade de recursos ofertados ou aumento da demanda da aplicação.

O custo estimado em 9131.81 USD, juntamente com um desperdício estimado de 3961.14 USD (43.38%), ressalta o impacto financeiro negativo de manter recursos não utilizados. Além disso, um jitter de -5.85 sugere que a variação na oferta de recursos está enfrentando desafios para se alinhar com as oscilações na demanda.

6.1.7 Propostas de melhoria dos parâmetros de provisionamento automático

Com base nos resultados observados, elencamos algumas propostas para explorar como modificações em parâmetros específicos da configuração de provisionamento automático podem influenciar a eficiência e a alocação dos recursos fornecidos para a aplicação. As propostas estão descritas a seguir:

- **Ajuste do limite superior em casos de superprovisionamento:** Propomos aumentar o limite superior quando identificado o superprovisionamento, visando melhorar a utilização dos recursos disponíveis. A configuração atual, com limites inferior e superior muito próximos, pode estar limitando a capacidade do sistema de responder de maneira econômica às variações na demanda.

- **Modificação na quantidade de *cores* em ações de expansão de recursos:** Sugerimos revisar a quantidade de *cores* alocados durante as ações de expansão dos recursos. A prática atual tende a levar a um superprovisionamento, possivelmente devido à alocação de recursos em grandes quantidades por ação de provisionamento. Uma abordagem mais refinada, com a alocação de instâncias de menor capacidade, pode permitir ajustes mais precisos e reduzir a capacidade ociosa.

- **Revisão da capacidade máxima, período de avaliação e tempo de espera para ações de provisionamento:** É crucial revisar a capacidade mínima e máxima de recursos, o período de avaliação e o tempo de espera após as ações de provisionamento.

Essas propostas serão avaliadas no próximo capítulo para compreendermos a influência de cada parâmetro na ação de provisionamento. Essa análise auxiliará na tomada de decisão sobre como ajustar esses valores de forma estratégica.

6.2 Aplicação B

Os dados utilizados dessa aplicação são referentes ao mês de Dezembro de 2023. E para essa aplicação a configuração consiste em uma política reativa de provisionamento de rastreamento de meta.

6.2.1 Parâmetros de configuração de provisionamento

Para a operação desta aplicação, a empresa adota a configuração de provisionamento descrita na Tabela 6.3, ao longo do estudo chamaremos de **Configuração Base.**

Política	Provisionamento Simples
Valor alvo	55%
Limiar de tolerância	10%
Capacidade mínima	16 *cores*
Capacidade máxima	140 *cores*
Tempo de espera	9 min

Tabela 6.3: Parâmetros definidos para a Aplicação B durante o mês de análise dos dados.

O valor específico para o limiar de tolerância ainda não foi estabelecido. Portanto, adotamos um valor de 10%, que foi selecionado arbitrariamente.

6.2.2 Análise da utilização de *cores*

A Figura 6.6 ilustra a quantidade de *cores* empregadas pela aplicação ao longo do mês de Dezembro. Observa-se uma oscilação considerável, similarmente ao observado na Aplicação A. No entanto, a demanda é relativamente menor, com uma variação aproximada entre 40 e 120 *cores*. Para a aplicação A, essa variação possui uma aplitude maior com valores entre 100 e 300.

Assim como para a aplicação A, o comportamento da utilização ao longo do dia foi investigado. A figura 6.7 mostra como foi essa distribuição, evidenciando que há uma variação na medianas indicando uma flutuação no número de *cores* usadas ao longo do dia, que pode ser influenciada por diversos fatores, como a demanda dos usuários ou processos internos que ocorrem em horários específicos.

Assim como ocorre com a Aplicação A, observa-se que a Aplicação B também tem períodos de menor demanda, particularmente durante a madrugada e as primeiras horas da manhã. No entanto, a utilização da Aplicação B é menos intensa, resultando em uma dispersão de dados menos acentuada quando comparada à Aplicação A. Indicando que, embora ambas as aplicações apresentem reduções de atividade em horários similares.

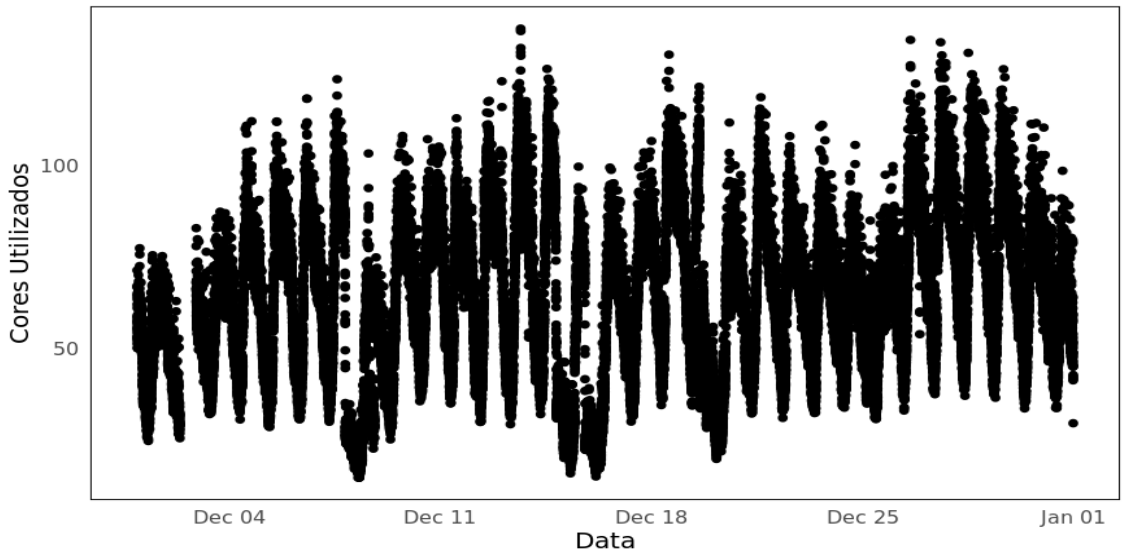

Figura 6.6: Variação diária na utilização de *cores* em Dezembro, com os dados da Aplicação B.

6.2.3 Avaliação da alocação de recursos

Na Figura 6.8 observa-se que há uma oferta suficiente de instâncias para atender à demanda quase todo o período analisado, pois a linha vermelha permanece acima da linha azul em parte significativa do tempo.

Existe um desperdício médio de recursos alocados de 40%. A quantidade acumulada de instâncias que permaneceram ociosas durante o período analisado totalizou 911376.8. Esse número corresponde a 39.36% do total de instâncias que estavam disponíveis ao longo do tempo, que foi de 2315651.

A consistência do excesso de instâncias disponíveis sugere uma capacidade robusta que pode ter sido planejada para garantir que a aplicação opere sem interrupções, mesmo durante os períodos de pico de demanda. No entanto, esse excesso de capacidade resulta em um custo significativo de recursos que não estão sendo efetivamente utilizados. Essa situação aponta para uma possível melhoria na utilização de recursos, onde o número de instâncias poderia ser ajustado para mais perto da demanda real.

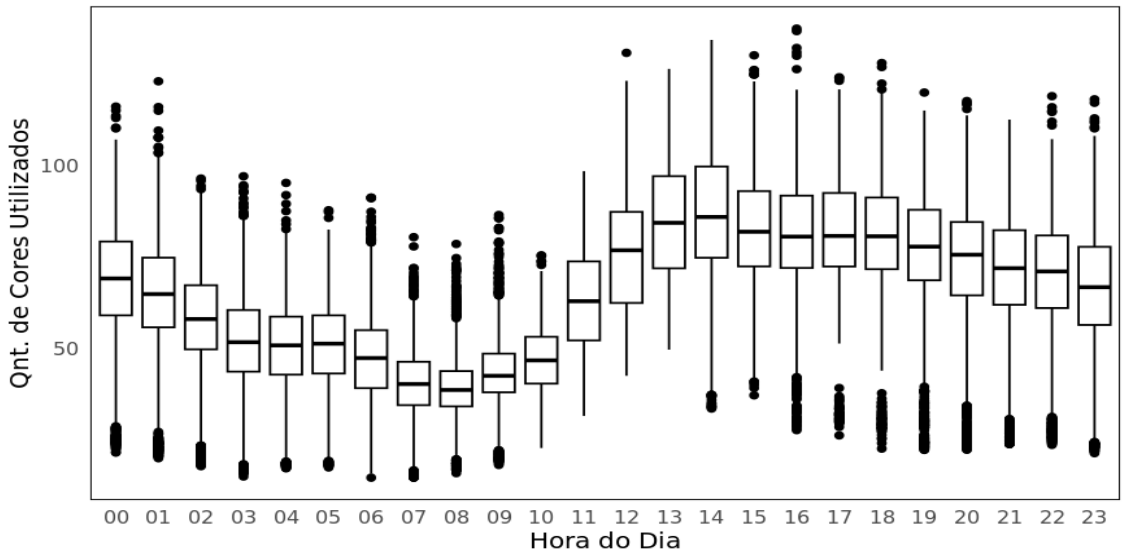

Figura 6.7: Distribuição da utilização de *cores* no mês de Dezembro com os dados agrupados por hora, utilizando os dados da Aplicação B.

6.2.4 Avaliação dos limiares que acionam ações de provisionamento

A política adotada por esta aplicação visa manter a utilização próxima ao limiar definido, o que é visualmente representado pela única linha vermelha na Figura 6.9. Durante o período analisado, a utilização excedeu o limiar estabelecido em 54.93% do tempo. E a aplicação operou em sua capacidade máxima, ou seja, com 100% de utilização, durante 0.12% do período, o que corresponde a um total de 835 minutos. Em contrapartida, a utilização permaneceu abaixo do limiar em 45.07% do tempo. Esta tendência sugere que houve uma propensão maior para a mobilização de recursos adicionais, ao invés da sua redução, indicando que os provisionamento acionados tenderam mais para o reforço da capacidade.

6.2.5 Análise da capacidade de processamento baseado em requisições

De forma semelhante à Aplicação A, determinamos uma fórmula que estimasse quantas requisições a Aplicação B conseguiria atender com base em sua capacidade. Utilizando uma abordagem de regressão linear múltipla, consideramos variáveis como o número de *cores* alocados e a porcentagem de utilização da CPU, esperando capturar a capacidade máxima de processamento da aplicação, o valor definido para a utilização foi de 100%. A capacidade

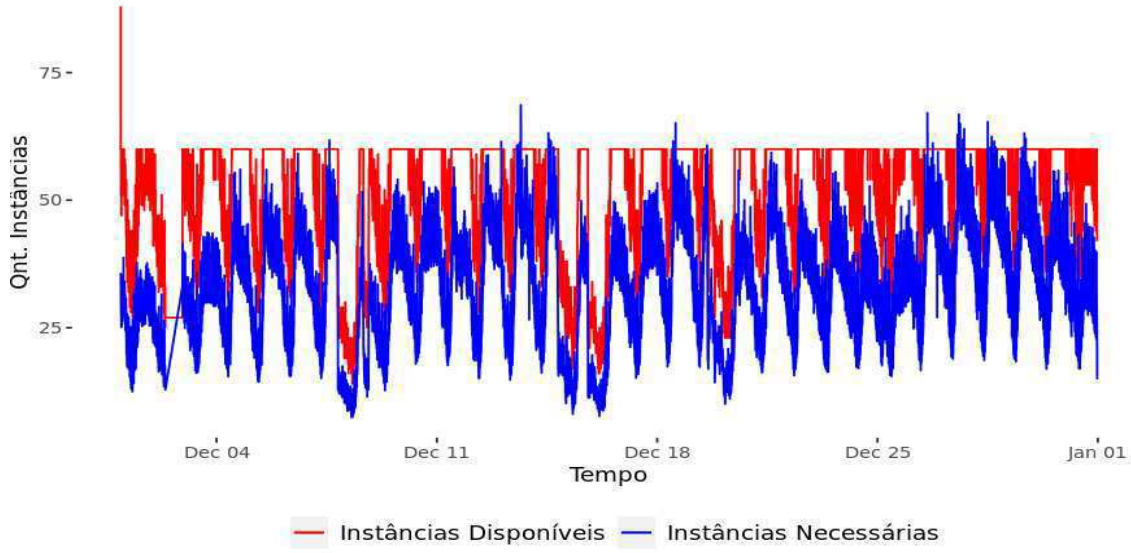

Figura 6.8: Comparativo diário entre instâncias disponíveis e necessárias em Dezembro, utilizando os dados da Aplicação B, mostrando que houveram momentos de sub e superprovisionamento.

estimada de requisições que a Aplicação B pode atender foi modelada pela seguinte Equação 6.2.

$$\text{Capacidade} = -286665.8 + 2025.66 \times \textit{Cores}\ \text{Alocados} + 10785.73 \times \text{utilização} \quad (6.2)$$

Para a Aplicação B, a análise de regressão resultou em um coeficiente de determinação de 0.46. Este valor indica que menos da metade da variabilidade no número de requisições recebidas pode ser atribuída às variáveis utilizadas no modelo. Isso sugere que existem outros fatores influentes não capturados pelo modelo que afetam a capacidade de processamento da aplicação. Tal resultado enfatiza a complexidade da Aplicação B e a possível necessidade de explorar variáveis adicionais ou modelos alternativos que possam fornecer uma compreensão mais abrangente do comportamento das requisições.

A tendência geral e os picos ocasionais destacados na Figura 6.10 não são completamente explicados pelo modelo atual, como evidenciado pelo R^2 de 0.46. A capacidade de processamento, ilustrada pela linha azul consistente, permanece estável ao longo do tempo, sugerindo uma infraestrutura de processamento que não varia sua capacidade em resposta à demanda. No entanto, a linha vermelha, que representa as requisições recebidas, mostra uma dinâmica

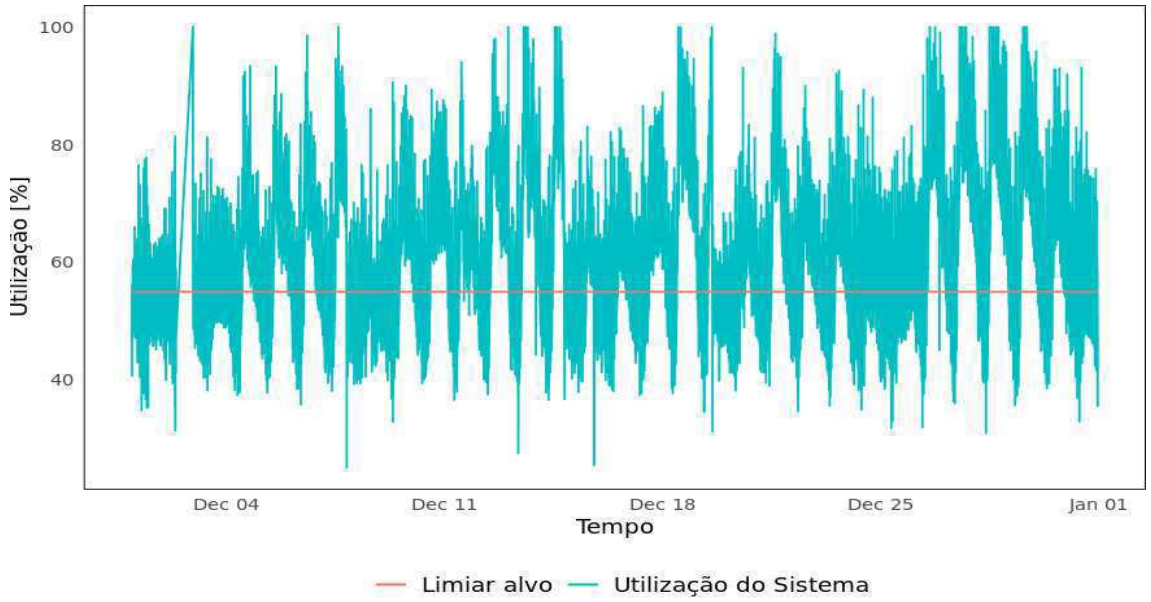

Figura 6.9: Monitoramento da utilização percentual do sistema em Julho, utilizando os dados da Aplicação B com a Configuração Base.

muito mais volátil, com picos que ultrapassam em muito a capacidade do sistema. Os números mostram que a mesma porcentagem da Aplicação A foi seguida para essa aplicação, a quantidade de requisições não atendidas foi de 0.08% e 99.92% foram atendidas.

Os valores das métricas MAE de 113575.9 e RMSE de 159320.6, quando avaliados no contexto de um volume de dados na ordem de milhões, podem ser considerados relativamente baixos. Isso sugere que o modelo de previsão, apesar de ter um R^2 de apenas 0.46, ainda fornece estimativas com uma boa precisão em termos absolutos. Então, mesmo que o modelo possa não explicar toda a variabilidade observada nas requisições recebidas, as métricas de erro indicam que suas previsões têm uma precisão operacionalmente útil.

6.2.6 Avaliação Quantitativa de Desempenho e Eficiência de Recursos

A Tabela 6.4 apresenta os valores obtidos para as métricas selecionadas para estudar neste estudo. A análise desses resultados revela alguns aspectos interessantes sobre a eficiência e os custos operacionais do sistema.

A acurácia de subprovisionamento é de apenas 1%, o que indica que o sistema raramente teve menos recursos do que o necessário, um indicativo de confiabilidade na manutenção da performance. No entanto, um quadro diferente surge com a acurácia de superprovisiona-

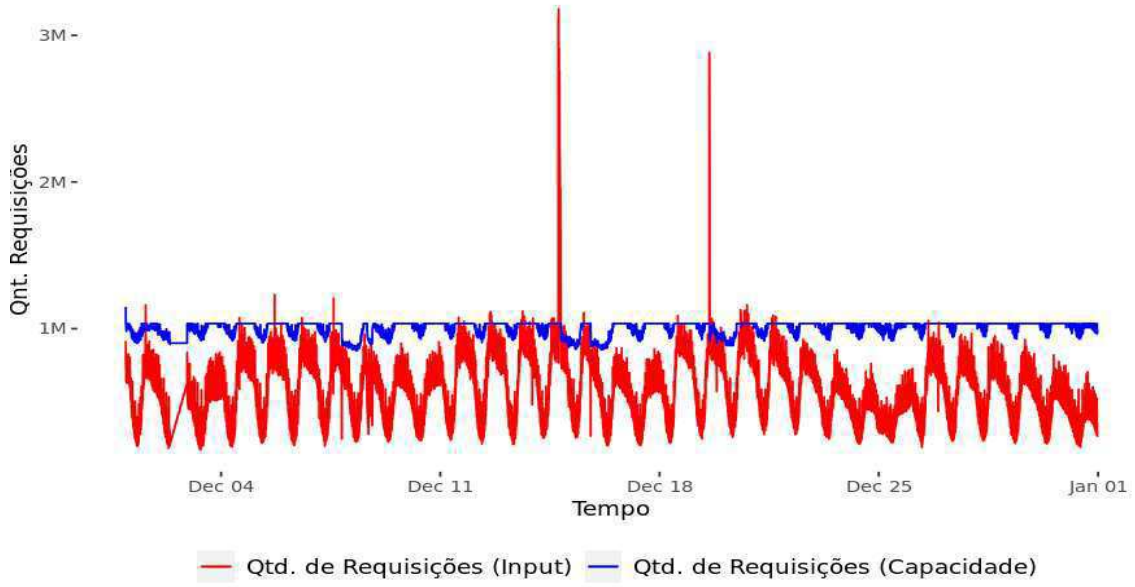

Figura 6.10: Análise temporal da capacidade de processamento de requisições, utilizando os dados da Aplicação B.

mento, que está em 7367%. Este número sugere que os recursos frequentemente excediam as demandas reais, levando a um superprovisionamento significativo.

O tempo de subprovisionamento foi de apenas 0.12%, 34 minutos do período analisado, reforçando a ideia de que a falta de recursos foi um evento raro. Em contraste, o sistema esteve superprovisionado durante 99.89% do tempo operacional, o que destaca uma tendência para alocar mais recursos do que o necessário durante quase todo o tempo.

A métrica TAR de 67.27% indica que ainda existe espaço para uma melhor utilização dos recurso e que o sistema poderia atender mais requisições. O jitter negativo de -2.34, que os recursos ofertados não conseguiram se adaptar bem a demanda. Mas que em comparação com a Aplicação A essa tentativa de adaptação produziu melhores resultados.

Do ponto de vista financeiro, a configuração do sistema levou a um custo estimado de 2979.194 USD. Notavelmente, o desperdício estimado foi de 1161.94 USD, o que equivale a aproximadamente 39% do custo total. Isso indica uma área significativa de ineficiência, onde quase dois quintos do investimento podem não estar contribuindo para o desempenho efetivo do sistema.

Esses dados apontam para uma oportunidade clara de melhoria. Reduzir o superprovisionamento pode diminuir os custos desnecessários sem comprometer a confiabilidade do

sistema. Para uma gestão de recursos mais eficaz, a empresa poderia considerar ajustar suas políticas de provisionamento automático para melhor alinhar a capacidade de recursos com as demandas reais.

Métrica	**Config. Base**
Acurácia Subprovisionamento	1%
Acurácia Superprovisionamento	7367%
ADI	2.3e+10
Tempo de subprovisionamento	0.12%
Tempo de superprovisionamento	99.89%
TAR	67.27%
jitter	-2.34
Custo estimado	2979.19 USD
Desperdício estimado	1161.94 USD

Tabela 6.4: Métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando a configuração estabelecida pela empresa, utilizando os dados da Aplicação B.

6.2.7 Propostas de Otimização dos Parâmetros de Provisionamento Automático

A partir da análise realizada, observamos que a empresa tem uma tendência a alocar recursos além do necessário, resultando em um desempenho robusto da aplicação durante a maior parte do tempo. Contudo, essa prática vem com um alto custo financeiro e gera uma quantidade considerável de desperdício. Para mitigar o superprovisionamento e os custos associados, várias estratégias podem ser implementadas. Neste estudo, focaremos em avaliar o impacto de duas propostas para melhorar a utilização de recursos, usando as métricas já estabelecidas para medir suas eficácias.

As duas estratégias propostas são:

- **Ajustar o limiar alvo para um valor mais elevado**: com o intuito de aproveitar melhor os recursos que já foram alocados, queremos analisar se é essa proposta aumenta a eficiência sem comprometer o desempenho da aplicação.

- **Utilizar uma política de provisionamento simples**: a intenção é verificar como será feita a alocação de recursos e se com a configuração que iremos utilizar, é possível diminuir os custos sem prejudicar o desempenho.

Essas abordagens serão analisadas cuidadosamente para avaliar qual delas, ou que combinação, pode resultar em um equilíbrio mais eficiente entre custo e desempenho, alinhando o provisionamento de recursos mais de perto com a utilização efetiva e minimizando o desperdício financeiro.

Capítulo 7

Análise das hipóteses sobre como alterar ações de provisionamento

Neste capítulo, analisamos as propostas de melhoria dos parâmetros de provisionamento automático sugeridas no Capítulo 6. Considerando que a demanda da aplicação permanece constante, enquanto os recursos alocados variam, focamos nossas análises nesse aspecto.

7.1 Aplicação A

Com base nas análises realizadas no capítulo anterior, foram realizadas simulações de provisionamento automático considerando mudanças nos seguintes parâmetros das configurações das políticas:

- Alteração do limite superior de 65% para 75% (Config. 2);
- Modificação na quantidade de cores em ações de expansão de recursos de 32 para 16 (Config. 3);
- Alteração simultânea no limite superior de 65% para 75% e modificação na quantidade de *cores* em ações de expansão de recursos de 32 para 16 (Config. 4);
- Alteração da capacidade máxima de 240000 para 300 (Config. 5);
- Alteração do período de avaliação de 2 minutos para 4 minutos (Config. 6);

- Alteração do tempo de espera de 3 minutos para 1 minuto (Config. 7).

A capacidade máxima foi ajustada para 300, levando em conta a média de utilização de *cores* de 270 e o pico máximo atingido de 350. Essa alteração foi feita considerando o potencial impacto na demanda, uma vez que influencia a quantidade máxima de recursos que podem ser alocados. Os demais valores foram escolhidos de forma arbitrária, com a intenção de alterar a forma como os recursos seriam provisionados.

Nas seções 7.1.1, 7.1.2, 7.1.3 e 7.1.3, serão verificadas a alocação de recursos, os limiares que acionam ações de provisionamento, a análise de requisições e capacidade de processamento baseadas em requisições e a avaliação quantitativa de desempenho e eficiência de recursos.

7.1.1 Avaliação da alocação de recursos

Ao redefinir as ações de provisionamento, buscamos avaliar o impacto na alocação de recursos através das diversas configurações aplicadas à Aplicação A.

As simulações realizadas mostraram que ao ajustar o limite superior para 75% (Config. 2), observamos uma redução significativa na quantidade de instâncias disponíveis, levando a uma porcentagem média de desperdício de 38%. Isso representou uma diminuição de 22.45% no total de instâncias desnecessárias em comparação com a Configuração Base, como ilustrado na Figura 7.1.

Analisando as configurações subsequentes, notamos que ao implementar a Configuração 3, com um incremento de 16 *cores* na expansão de recursos, o desperdício de recursos reduziu para 42.33%, indicando uma melhoria de 1.6% em relação à Configuração Base (43.93%). Isso sugere que o ajuste preciso no volume de recursos tem um impacto direto na redução do superprovisionamento.

A Configuração 4 demonstrou um alinhamento ainda mais próximo entre as instâncias disponíveis e as necessárias. Com uma taxa de desperdício médio de 37%, a Configuração 4 apresentou a menor porcentagem de desperdício entre todas as configurações analisadas, mostrando-se uma combinação benéfica para mitigar o superprovisionamento de recursos.

As Figuras 7.2 e 7.3 apresentam as análises comparativas das quantidades de instâncias disponíveis e necessárias sob as políticas de provisionamento das Configurações 3 e 4. É

evidente que as mudanças implementadas contribuíram para uma gestão de recursos mais ajustada e econômica.

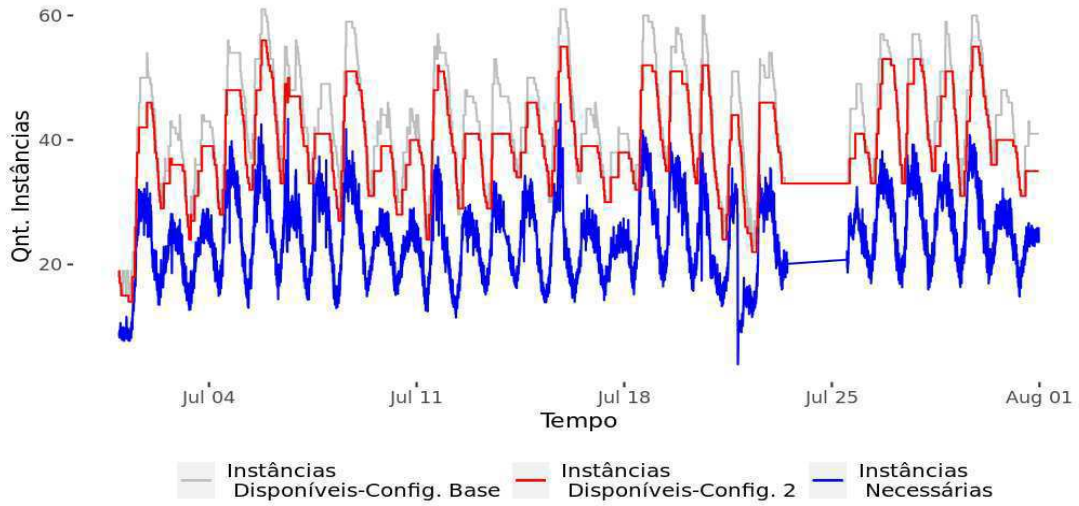

Figura 7.1: Análise diária do número de instâncias disponíveis em relação às necessárias, sob a política de provisionamento simulada com o limite superior ajustado para 75%.

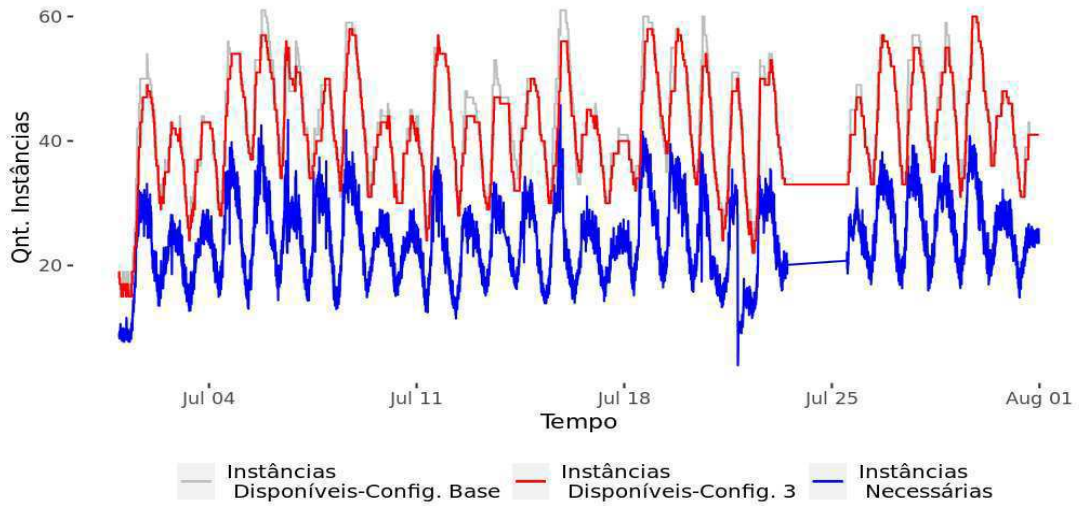

Figura 7.2: Análise diária do número de instâncias disponíveis em relação às necessárias, sob a política de provisionamento simulada com os dados da Aplicação A e com a Configuração 3, na qual a quantidade de incremento de recursos igual a 16 *cores*.

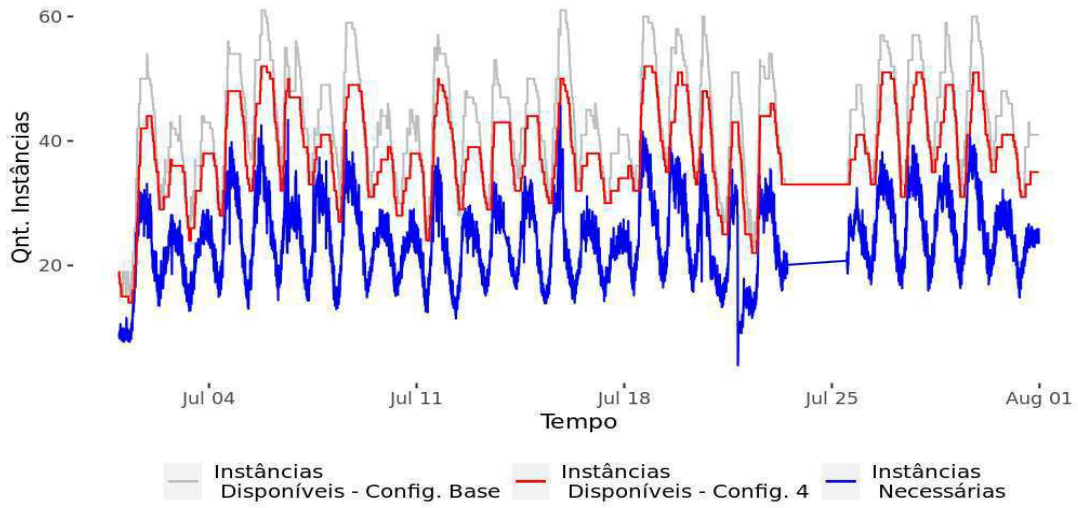

Figura 7.3: Análise temporal do comparativo entre a quantidade de instâncias disponíveis e necessárias, utilizando o limite superior igual a 75% e quantidade de *cores* em ações de expansão de recursos igual a 16, com os dados da Aplicação A.

As figuras 7.4, 7.5 e 7.6 apresentam, respectivamente, a comparação das Configurações 5, 6 e 7 em relação à Configuração Base. Estas configurações resultaram em um desperdício de recursos de 43.77%, 43.77% e 42.39%, respectivamente.

7.1.2 Avaliação dos limiares que acionam ações de provisionamento

Na análise das políticas de provisionamento e sua influência sobre a utilização dos recursos da Aplicação A, observamos alguns efeitos quando alterado os limiares. A Figura 7.7 revela que, ao configurar o limite superior para 75% (Config. 2), a utilização dos recursos atingiu este patamar em apenas 6.15% das ocasiões, representando uma diminuição no número de ações de alocação de recursos, em relação a Configuração Base (9,75%), e indicando uma gestão mais eficiente, com menos ocorrências de superprovisionamento.

Além disso, em 27.06% dos intervalos, a utilização permaneceu abaixo do limite inferior estabelecido, sugerindo uma redução no desperdício de recursos quando comparado à Configuração Base. Contudo, houveram episódios esporádicos de subprovisionamento, indicados pela utilização que alcançou 100% em 0.05% dos intervalos, não observados na configuração com o limite de 65%, destacando o *trade-off* entre economia de recursos e risco de não

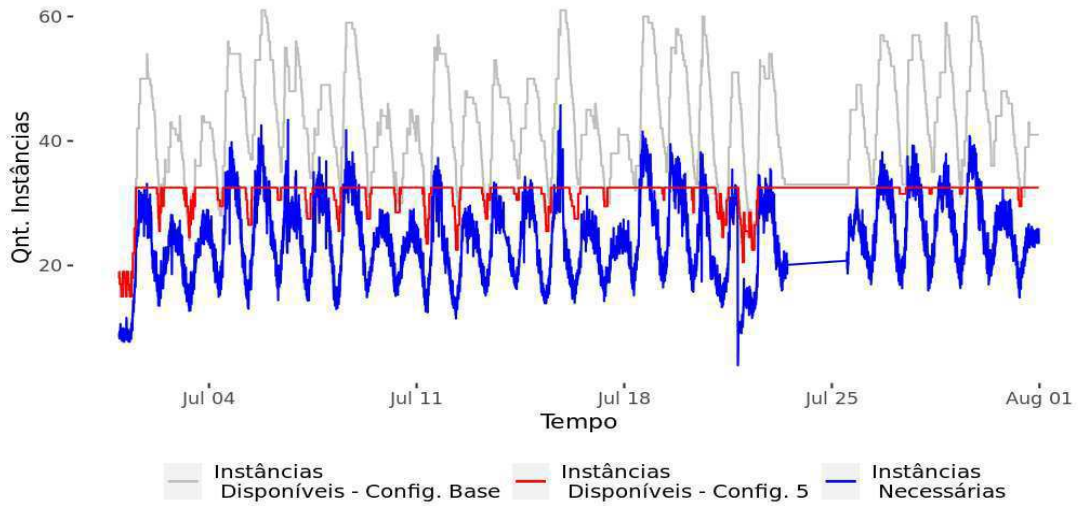

Figura 7.4: Esta análise foca no comparativo temporal entre a quantidade de instâncias disponíveis e as que são efetivamente necessárias. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 5. Esta configuração específica difere da Configuração Base apenas no que diz respeito à capacidade máxima de recursos alocados.

atender completamente a demanda em momentos pontuais.

A Configuração 3, ilustrada na Figura 7.8, mostrou que a utilização ultrapassou o limiar superior em 17.05% das vezes, um aumento em comparação com a Configuração Base. Isso implicou em um número maior de ações de provisionamento automático, refletindo a necessidade de uma alocação adicional de recursos em resposta a picos de demanda. No entanto, a utilização ficou abaixo do limiar inferior em 34.6% das ocasiões, uma diminuição nas ações de redução da infraestrutura quando comparamos com a Configuração Base, que foi de 39.3% .

A Figura 7.9 mostra como a Configuração 4 comportou-se em relação à utilização dos recursos. Com o limite superior a 75% e um incremento de recursos de 16 *cores*, a utilização excedeu este limite em 11.72% das ocasiões. Embora este valor seja maior do que na Configuração 2, ele é menor que o observado na Configuração 3, indicando uma eficácia intermediária na resposta às demandas. Essa configuração também resultou em menos ações de redução da infraestrutura, com a utilização alcançando o limite inferior em apenas 24,13% das ocasiões.

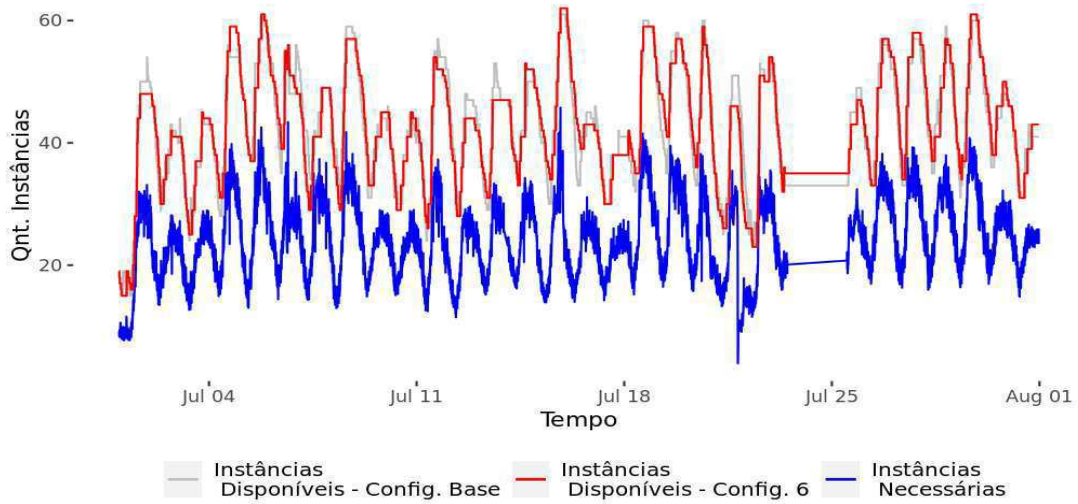

Figura 7.5: Esta análise foca no comparativo temporal entre a quantidade de instâncias disponíveis e as que são efetivamente necessárias. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 6. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao período de avaliação que foi alterado de 2 para 4 .

Os resultados da simulação com a Configuração 5 mostraram que esse foi o cenário no qual a utilização de recursos ficou mais frequentemente fora dos limites definidos nos parâmetros da configuração. Houve uma incidência de atingimento do limite superior em 15.72% das vezes, e dentro desse valor, em 1.12% das ocasiões, o limite de 100% dos recursos disponíveis foi alcançado. Além disso, a utilização ficou abaixo do limite inferior em 49.21%.

A Configuração 6 disparou mais ações para provisionar recursos em relação a Configuração Base, sendo a diferença de porcentagem entre esses valores de 3.36%. Consequentemente o limite inferior também foi atingido mais vezes (40.5%), indicando maior desperdício de recursos em comparação com a Configuração Base.

Os resultados da simulação com a Configuração 7 mostra que em 8.23% das vezes o limite superior foi atingido e em comparação com a Configuração Base menos ações de alocação de recursos foram acionadas. O limite inferior foi atingido em 29% das vezes, indicando que haviam menos recursos alocados em excesso quando comparado com a Configuração Base, que atingiu esse limite em 39.3% das vezes.

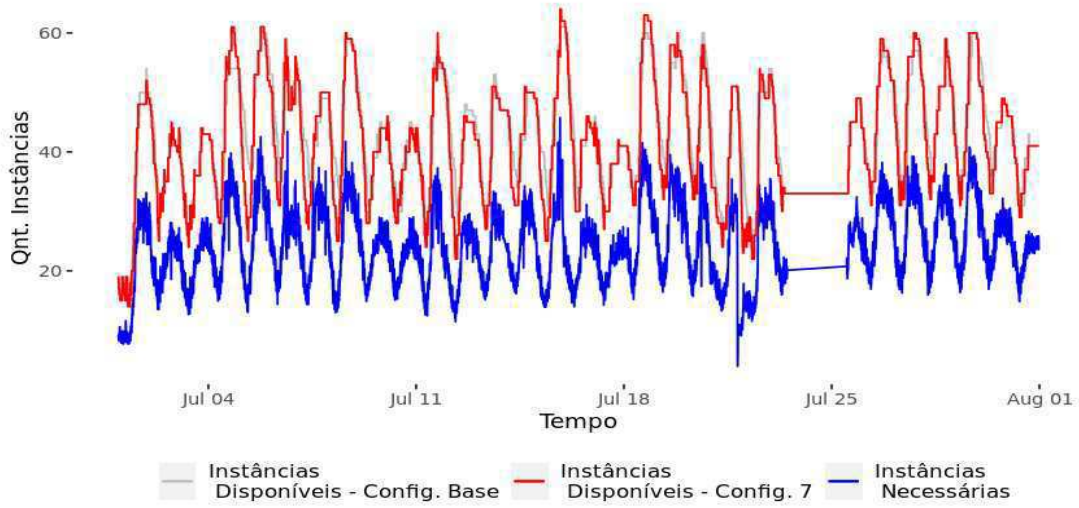

Figura 7.6: Esta análise foca no comparativo temporal entre a quantidade de instâncias disponíveis e as que são efetivamente necessárias. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 7. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao tempo de espera que foi alterado de 3 para 1.

As Figuras 7.10, 7.11 e 7.12 mostram o comportamento da utilização de recursos em relação aos limiares para as Configurações 5, 6 e 7, respectivamente.

7.1.3 Análise de requisições e capacidade de processamento baseado em requisições

Para consolidar as análises referentes às diferentes configurações e seus impactos na capacidade de processamento de requisições da aplicação, realizamos uma série de regressões múltiplas.

Cada regressão visa estimar a capacidade de atendimento de requisições em função da quantidade de *cores* alocados, bem como da utilização do sistema, que foi fixada em 100%, refletindo o uso pleno do processamento disponível. Para essa aplicação, os valores de R^2 obtidos foram de aproximadamente 0.7.

A Configuração 2 é modelada pela Equação 7.1. Essa configuração demonstrou uma

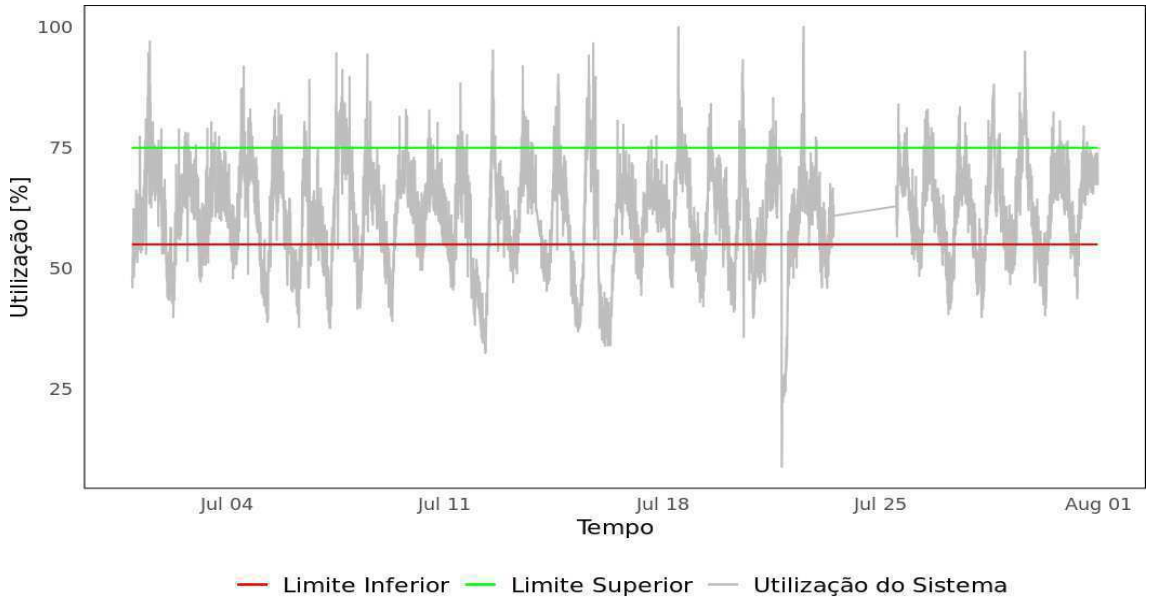

Figura 7.7: Utilização do sistema ao longo do tempo, destacando os limites que acionam as ações de provisionamento automático. Simulando com a Configuração 2 e os dados da Aplicação A.

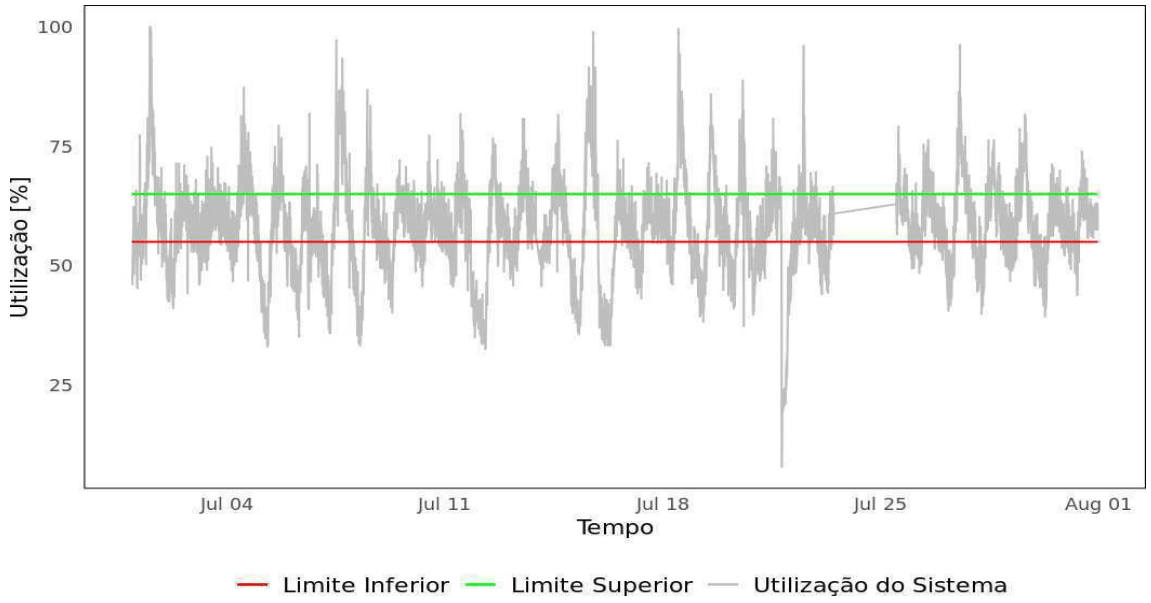

Figura 7.8: Monitoramento de utilização percentual do sistema, destacando os limites superior e inferior. Utilizando os dados da Aplicação A com a Configuração 3, na qual a quantidade de *cores* em ações de expansão de recursos igual a 16.

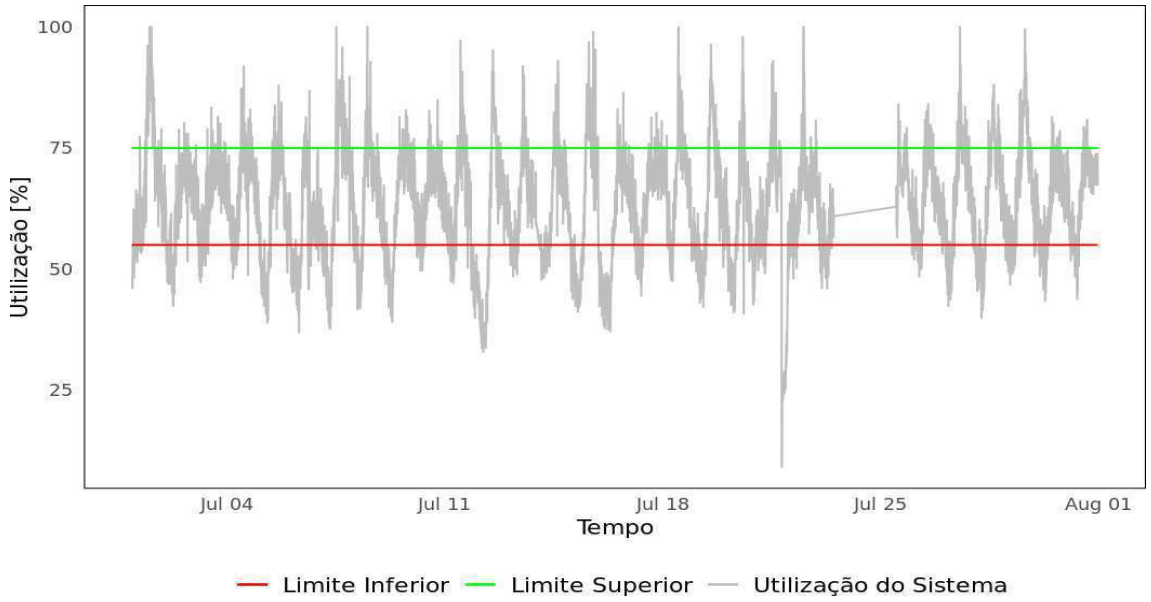

Figura 7.9: Monitoramento da utilização de recursos destacando os limites superior e inferior, empregando o limite superior igual a 75% e quantidade de *cores* em ações de expansão de recursos igual a 16.

ligeira melhoria na precisão do modelo, com uma redução de 0.84% no MAE e um aumento de 1% no RMSE, indicando desafios no tratamento de valores atípicos.

$$\text{Capacidade}_{\text{Config.2}} = -769688.4 + 4379.928 \times \textit{Cores Alocados} + 22944.28 \times \text{utilização} \quad (7.1)$$

A Figura 7.13 apresenta uma capacidade de atendimento de requisições levemente inferior à Configuração Base, indicando um superprovisionamento reduzido.

A Configuração 3 é descrita pela Equação 7.2.

$$\text{Capacidade}_{\text{Config. 3}} = -745997.1 + 4306.40 \times \textit{Cores Alocados} + 22887.2 \times \text{utilização} \quad (7.2)$$

A métrica MAE da Configuração 3 foi de 156860.9, ligeiramente inferior (1.26%) ao da Configuração Base de 158861.9, indicando uma precisão ligeiramente melhor na previsão das requisições.

Além disso, o RMSE também apresentou uma redução, passando de 234163.9 para 233406.8. Essa diminuição (0.32%) sugere que essa configuração pode estar lidando melhor com os valores extremos, resultando em previsões mais acuradas.

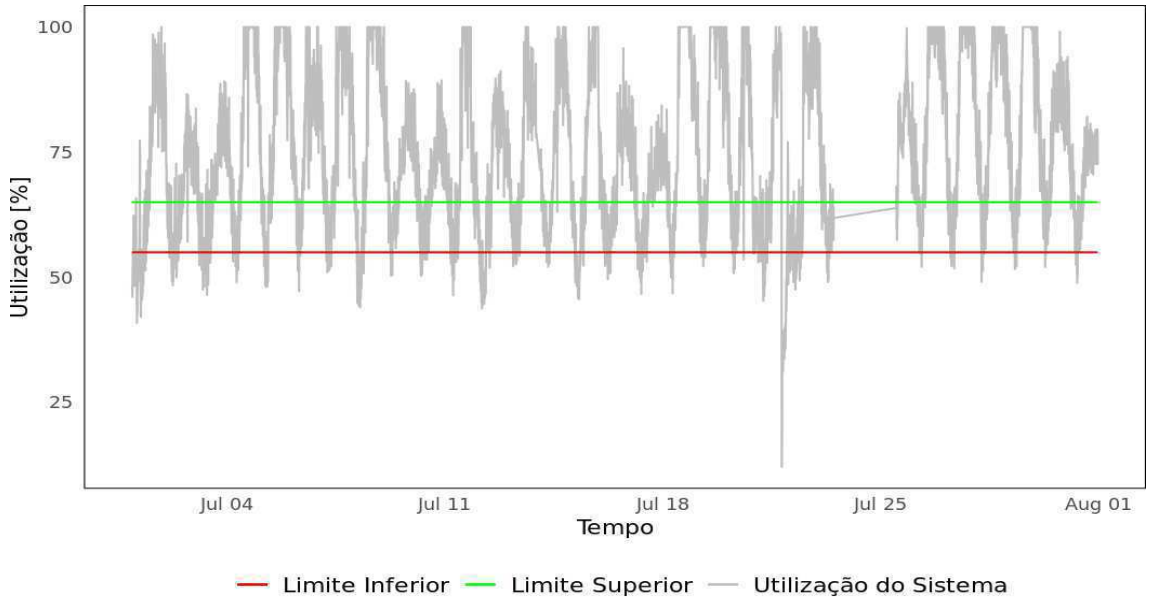

Figura 7.10: Esta imagem apresenta a utilização de recursos, mostrando também os limites que foram estabelecidos na configuração para acionar ações de provisionamento automático. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 5. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao a capacidade máxima.

A Configuração 4 é representada pela Equação 7.3.

$$\text{Capacidade}_{\text{Config4}} = -764965.6 + 4428.418 \times \textit{Cores Alocados} + 22603.57 \times \text{utilização} \quad (7.3)$$

Ao compararmos a Configuração 4 com as configurações anteriores, observamos que houve um desenvolvimento contínuo em termos de precisão do modelo, conforme refletido pelas métricas MAE e RMSE.

Para a Configuração 4, o MAE alcançou 156394.9, o que representa a menor pontuação entre todas as configurações analisadas. Em contraste, o RMSE foi de 235683.3, que é superior ao da Configuração Base (234163.9) e da Configuração 3 (233406.8), mas ainda inferior ao da Configuração 2 (236543.8). Essa métrica sugere que, apesar da melhoria na média dos erros (MAE), a Configuração 4 pode não ter tratado eficazmente os *outliers*.

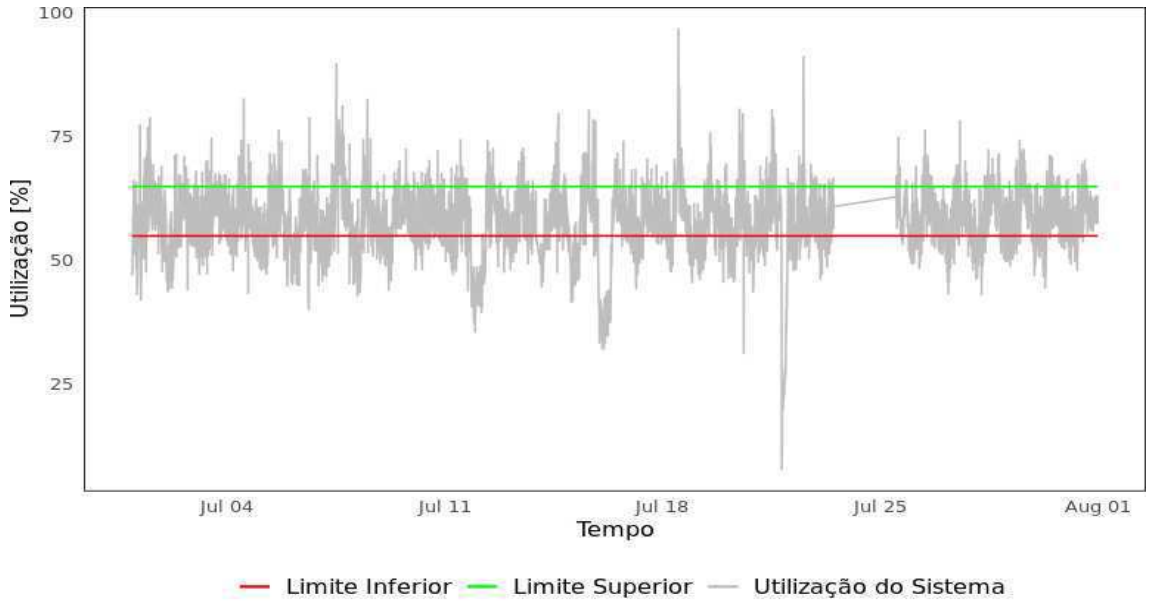

Figura 7.11: Esta imagem apresenta a utilização de recursos, mostrando também os limites que foram estabelecidos na configuração para acionar ações de provisionamento automático. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 6. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao período de avaliação que foi alterado de 2 para 4.

Avaliação quantitativa de desempenho e eficiência de recursos

Devido a quantidade de resultados gerados, para melhor visualização dividimos em duas Tabelas 7.1 e 7.2.

A Tabela 7.1 mostra o comportamento das Configurações 2, 3 e 4, comparando os resultados com a Configuração Base.

Ao analisarmos a Tabela 7.1, observamos um panorama detalhado que nos permite comparar as métricas de eficiência e custo das diferentes configurações de provisionamento automático em relação à configuração base.

Todas as configurações mantiveram uma acurácia de subprovisionamento de 0%, exceto pela Configuração 4, que apresentou um leve aumento para 2%.

A acurácia de superprovisionamento mostra uma variação entre as configurações, com a Configuração 2 (6930%) e a Configuração 3 (7908%) apresentando melhorias em relação à base, enquanto a Configuração 4 oferece a redução mais substancial para 6437%. Isso implica uma gestão de recursos mais afinada nas Configurações 2 e 3, e ainda mais na 4, su-

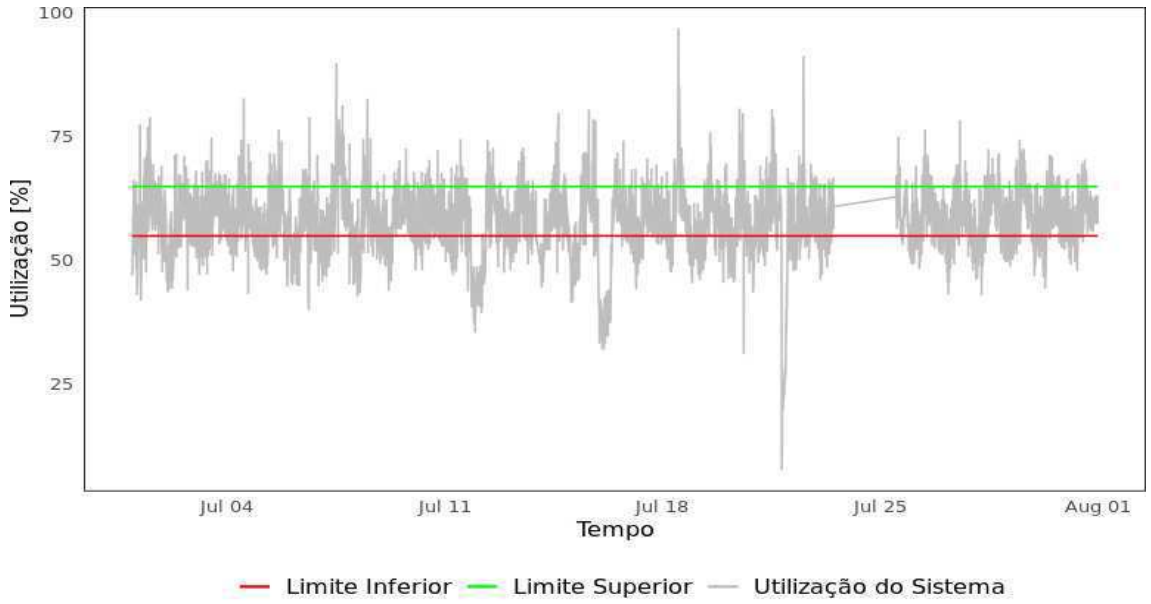

Figura 7.12: Esta imagem apresenta a utilização de recursos, mostrando também os limites que foram estabelecidos na configuração para acionar ações de provisionamento automático. Utiliza-se, como referência, os dados provenientes da Aplicação A operando sob a Configuração 7. Esta configuração específica difere da Configuração Base apenas no que diz respeito ao tempo de espera que foi alterado de 3 para 1.

gerindo um alinhamento mais próximo com a demanda real e menor desperdício de recursos.

O tempo de subprovisionamento e superprovisionamento também revela diferenças sutis entre as configurações, com a Configuração 4 mostrando um ligeiro aumento no tempo de subprovisionamento (0.25%) e uma pequena redução no tempo de superprovisionamento (99.75%), evidenciando uma tentativa de equilibrar entre eficiência e a disponibilidade de recursos.

A métrica TAR mostra a consistência na alocação de recursos, respectivamente. A Configuração 4, por exemplo, conseguiu alcançar o TAR mais elevado (70.58%), sugerindo uma alocação de recursos mais precisa.

Do ponto de vista financeiro, as Configurações 2, 3 e 4 conseguiram reduzir tanto o custo estimado quanto o desperdício em comparação com a configuração base, com a Configuração 4 sendo a mais econômica em termos de custo total e desperdício. Essa análise financeira destaca a importância de escolher a configuração adequada para otimizar tanto a eficiência operacional quanto o custo.

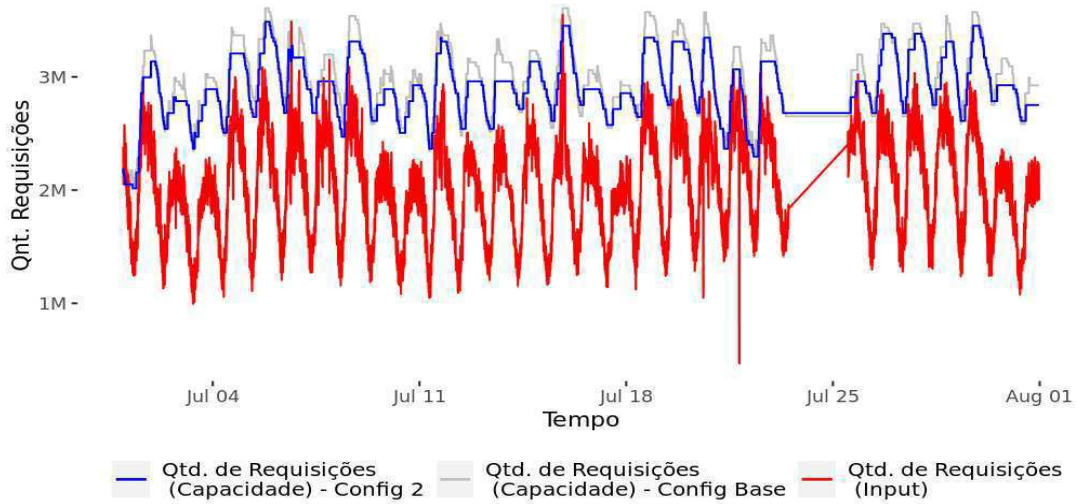

Figura 7.13: Análise temporal do comparativo entre a quantidade de requisições do volume de entrada e a capacidade de atendimento das mesmas. Utilizando os dados da Aplicação A e a Configuração 2, cujo o limite superior é igual a 75%

A Tabela 7.2 mostra o comportamento das Configurações 5, 6 e 7 comparando com a Configuração Base.

A análise das métricas obtidas nas Configurações 5, 6 e 7, comparadas com a Configuração Base, oferece percepções importantes sobre o gerenciamento de recursos.

Começando pela Configuração 5, observamos um aumento de 68% na acurácia de subprovisionamento, contrastando com os 0% da Configuração Base. Isso indica que limitar os recursos oferecidos pode prejudicar o desempenho da aplicação. A acurácia de superprovisionamento nesta configuração teve uma redução de 51.70%. O novo valor foi de 4029%, em comparação com os 8341% da Base.

Outro ponto importante é a redução no desperdício estimado, que caiu 57.59%, evidenciando uma gestão de recursos mais eficiente. Além disso, houve uma ligeira diminuição de 1.96% no custo estimado, sinalizando uma otimização de custos.

A Configuração 6, embora apresente poucas variações nas métricas de sub e superprovisionamento em comparação com a Base, mostra uma diminuição de 2.75% no custo estimado. Isso sugere uma alocação de recursos ligeiramente mais eficiente, apesar de não haver mudanças significativas nas outras métricas.

Métrica	Config. Base	Config. 2	Config. 3	Config. 4
Acurácia subprovisionamento	0%	0%	0%	2%
Acurácia superprovisionamento	8341%	6930%	7908%	6437%
Tempo de subprovisionamento	0%	0.05%	0.03%	0.25%
Tempo de superprovisionamento	100%	99.95%	99.97%	99.75%
TAR	67.27%	69.18%	67.44%	70.58%
Desperdício estimado [%]	43.38%	38.57%	41.78%	36.67%

Tabela 7.1: Comparativo das métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando diferentes configurações, incluindo a Configuração 4 com limite superior de 75% e alocação de *cores* fixa em ações de expansão de recursos.

Já a Configuração 7, ao ajustar o tempo de espera, encontrou um equilíbrio entre eficiência operacional e custo. Ela conseguiu manter o desperdício controlado e o superprovisionamento em um nível gerenciável, com uma redução de 5.31% no desperdício estimado e de 2.30% no custo, em comparação com a Configuração Base.

7.1.4 Discussão das modificações na configuração da Aplicação A

As análises apresentadas anteriormente mostram que, para a Aplicação A, são validas as propostas de elevar o limite superior para evitar situações de superprovisionamento, modificar a quantidade de *cores* em ações de expansão de recursos e alterar os valores para a capacidade máxima, período de avaliação e tempo de espera para ações de provisionamento.

As mudanças descritas nas subseções anteriores tiveram um impacto na redução do superprovisionamento. Algumas dessas modificações resultaram em uma melhor adequação entre a oferta e a demanda de recursos, possibilitando que a aplicação utilize mais eficientemente os recursos já disponíveis. Isso é evidenciado pela redução na acurácia de superprovisionamento e a porcentagem de desperdício de recursos disponíveis. No entanto, essa alteração

Métrica	**Config. Base**	**Config. 5**	**Config. 6**	**Config. 7**
Acurácia subprovisionamento	0%	68%	0%	2%
Acurácia superprovisionamento	8341%	4029%	8322%	7645%
Tempo de subprovisionamento	0%	10.37%	0%	0%
Tempo de superprovisionamento	100%	89.63%	100%	100%
TAR	67.27%	76.83%	64.90%	70.58%
Desperdício estimado [%]	43.38%	18.77%	44.83%	42.04%

Tabela 7.2: Métricas avaliativas alterando a capacidade máxima, período de avaliação e tempo de espera para ações de provisionamento em relação a Configuração Base, com os dados da Aplicação A.

também apresenta desafios, visto que geraram situações de subprovisionamento, nas quais foram observados episódios isolados de utilização de 100% dos recursos, que possivelmente comprometem o desempenho da aplicação.

Do ponto de vista econômico, ambas as modificações se mostraram mais vantajosas do que a Configuração Base. Contudo, como algumas configurações se demonstraram economicamente mais viável que outras, percebemos que isso é indicio de que ainda existe margem para melhorias, especialmente em termos de custo-benefício.

Além disso, é importante destacar a relação entre as métricas de ADI e acurácia. Uma acurácia reduzida não implica necessariamente em uma diminuição do ADI. Isso ocorre porque a acurácia é medida com base na disponibilidade ou indisponibilidade de recursos, enquanto o ADI é calculado em relação aos níveis de utilização estabelecidos na política de provisionamento.

Essa análise destaca a complexidade que é o gerenciamento de recursos em ambientes de computação em nuvem, evidenciando o desafio de equilibrar eficientemente a alocação de recursos, custos e desempenho da aplicação. Os resultados demonstram que as ferramentas

e o fluxo de estudo para servir como um guia para orientar decisões de provisionamento propostos são eficazes para auxiliar os SREs no processo de configuração de políticas de provisionamento automático. Isso inclui uma avaliação detalhada da relação custo-benefício associada a cada ajuste na configuração, proporcionando percepções do comportamento dos recursos alocados sem a necessidade de implementar testes em ambientes de produção.

7.2 Aplicação B

Com base nas análises realizadas no capítulo anterior, em relação a Aplicação B, realizamos simulações de provisionamento automático considerando mudanças nas configurações e na política utilizada.

As configurações executadas foram as seguintes:

- Ajuste no limiar alvo de 55% para 65% (Config. 2);
- Utilização uma política de provisionamento simples (Config. 3), com os seguintes valores para os paramêtros:

Política	Provisionamento Simples
Limite superior	70%
Limite inferior	40%
Incremento de recursos	2 *cores*
Decremento de recursos	2 *cores*
Capacidade mínima	2 *cores*
Capacidade máxima	500 *cores*
Período de avaliação	2 min
Tempo de espera	5 min

Tabela 7.3: Parâmetros definidos para a Aplicação B durante o mês de análise dos dados, utilizando a Configuração 3 para simular.

7.2.1 Avaliação da alocação de recursos

A implementação do limiar mais alto na Configuração 1 não resultou consistentemente em uma redução dos recursos alocados, com as instâncias disponíveis excedendo frequentemente aquelas da Configuração Base, como mostra a Figura 7.14. No entanto, essa configuração resultou em uma redução do desperdício médio de recursos para 35%, uma melhoria em relação aos 40% observados na Configuração Base, indicando uma economia de 5%. Além disso, houve uma diminuição de 19,36% instâncias ociosas, demonstrando eficácia na redução do superprovisionamento.

Em contraste, a Configuração 3, ajustou-se mais estreitamente à demanda variável, alocando recursos adicionais durante os períodos de pico. Esta abordagem assegurou que não houve deficit no atendimento, mas aumentou o desperdício médio de recursos para 45%, 10% acima da Configuração 2 e 5% mais que a Configuração Base. O superprovisionamento também aumentou significativamente para 27,98% instâncias ociosas, refletindo a tendência da Configuração 3 em gerar uma sobra de recursos mais significativa em períodos de menor utilização.

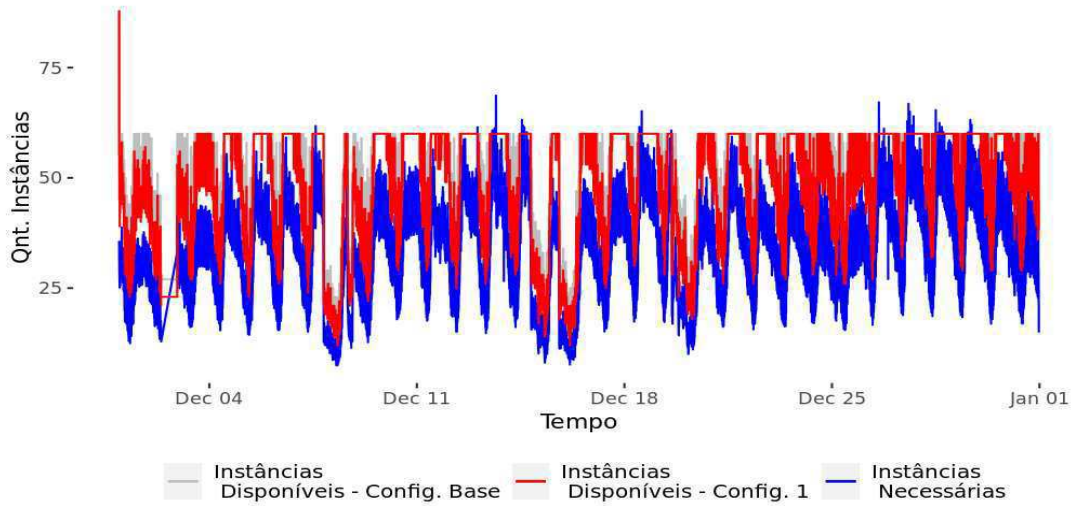

Figura 7.14: Análise diária do número de instâncias disponíveis em relação às necessárias, sob a política de provisionamento simulada com o limiar alvo igual a 65%, utilizando os dados referentes a Aplicação B, com a Configuração 2.

7.2.2 Avaliação dos limiares que acionam ações de provisionamento

A Figura 7.15 ilustra a utilização em relação ao limiar alvo de 65%, mostrando que em 39% das ocasiões, a utilização excedeu esse limiar, uma diminuição de 15.94% comparada à Configuração Base. Este resultado indica uma gestão de recursos mais eficiente, com menos ações para alocar recursos sendo necessárias. Além disso, a capacidade máxima de utilização, ou seja, 100%, foi atingida em 0.14% do tempo analisado (aproximadamente 844 minutos), marcando um incremento de 0.02% em relação à Configuração Base. Em contrapartida, a utilização permaneceu abaixo do limiar alvo em 60.52% do tempo, indicando uma maior proatividade em reduzir os recursos alocados.

A Figura 7.16 apresenta variações na utilização ao longo do tempo, incluindo picos acima e vales abaixo dos limites estabelecidos. Isso sugere uma alternância entre períodos de alta e baixa utilização, apontando para uma potencial ineficiência na alocação de recursos que poderia ser otimizada para manter a utilização dentro de faixas mais desejáveis. Durante o período analisado, a utilização ficou abaixo do limite inferior em cerca de 10% do tempo, levando a medidas para diminuir a quantidade de recursos alocados. Similarmente, o limite superior foi alcançado em aproximadamente 10% do tempo, resultando na alocação de recursos adicionais. Destaca-se que em apenas 0.1% dos casos, a utilização chegou a 100% da capacidade dos recursos disponíveis.

Estas análises, sustentadas pelas Figuras 7.15 e 7.16, destacam as diferentes abordagens e eficiências das configurações na gestão de recursos da Aplicação B. A Configuração 2 demonstra uma melhoria na economia de recursos e uma redução no superprovisionamento em comparação com a base. Por outro lado, a Configuração 3, embora proporcione uma maior flexibilidade em responder a flutuações na demanda, também indica áreas onde a eficiência na gestão dos recursos pode ser aprimorada para evitar tanto o super quanto o subprovisionamento.

Análise de requisições e capacidade de processamento baseado em requisições

A análise da capacidade de processamento de requisições da Aplicação B através das Configurações 2 e 3 revela nuances importantes na previsibilidade e eficácia dos modelos de regressão utilizados. Para a Configuração 2, a Equação 7.4 indicou um coeficiente de deter-

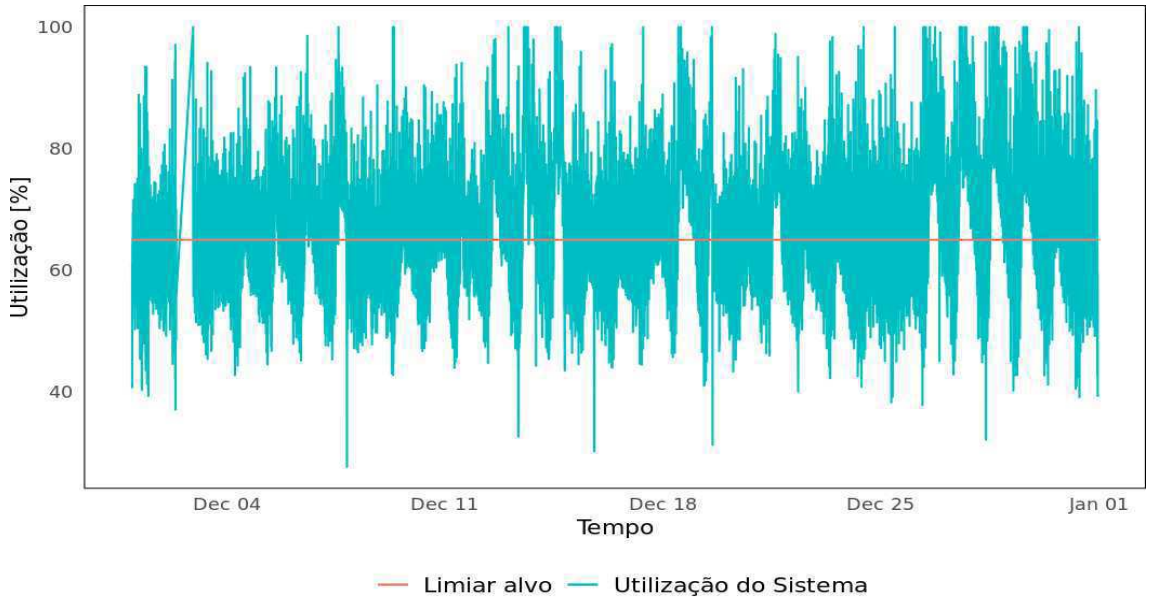

Figura 7.15: Utilização do sistema ao longo do tempo, destacando o limite que aciona as ações de provisionamento automático. Utilizando a política de provisionamento simulada com o limiar alvo igual a 65%, utilizando os dados referentes a Aplicação B, com a Configuração 2.

minação (R^2) de 0.45, uma ligeira redução em comparação ao R^2 de 0.46 da Configuração Base. Essa diminuição sugere uma capacidade levemente reduzida do modelo em explicar a variabilidade no número de requisições recebidas, possivelmente devido a uma captura menos eficaz dos fatores que influenciam a demanda por recursos de processamento.

$$\text{Capacidade} = -407666.8 + 3642.27 \times \textit{Cores} \text{ Alocados} + 9605.51 \times \text{utilização} \quad (7.4)$$

Acompanhando essa observação, a Figura 7.17 ilustra uma tendência de capacidade reduzida para receber requisições ao longo do tempo, com a análise temporal mostrando que as variações no número de requisições não são completamente explicadas pelo modelo, conforme refletido pelo R^2 de 0.45. Após a implementação da Configuração 2, houve um aumento de 1.96% no MAE (de 113,575.9 para 115799.6) e de 1.81% no RMSE (de 159320.6 para 162198.2), indicando alterações modestas nas métricas de erro. Mesmo com o aumento, os valores de MAE e RMSE ainda podem ser considerados relativamente baixos em relação à magnitude dos dados tratados, indicando que o modelo mantém uma utilidade prática.

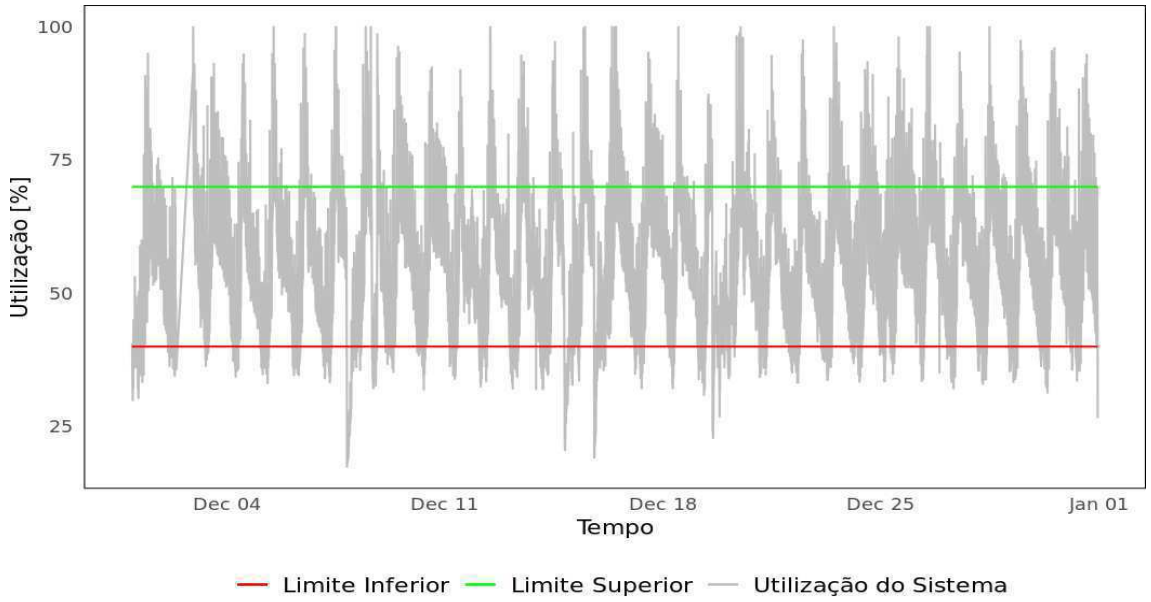

Figura 7.16: Utilização do sistema ao longo do tempo, destacando o limite que aciona as ações de provisionamento automático. Utilizando a política de provisionamento simulada com o limiar alvo igual a 65%, utilizando os dados referentes a Aplicação B, com a Configuração 3.

Por outro lado, a Configuração 3, definida pela Equação 7.5, resultou em um R^2 de 0.43. Este valor representa uma redução adicional em comparação aos resultados anteriores, sinalizando uma possível eficácia menor na incorporação dos elementos que afetam a demanda por recursos de processamento. A Figura 7.18 mostra que, apesar da Configuração 3 ser mais responsiva às variações na demanda, ainda existem limitações em lidar com os picos de demanda mais elevados.

$$\text{Capacidade} = -270243.7 + 3827.347 \times \textit{Cores}\ \text{Alocados} + 7178.28 \times \text{utilização} \quad (7.5)$$

Após a implementação da Configuração 3, observou-se um aumento adicional de 1.73% no MAE (de 115,799.6 para 117,794.8) e de 1.66% no RMSE (de 162198.2 para 164890.5). Esses aumentos, juntamente com a diminuição do R^2 para 0.43, sugerem que a capacidade em estimar o valor alvo não melhorou significativamente com a nova configuração.

Ambas as configurações indicam que, apesar das mudanças introduzidas, uma proporção substancial da variabilidade nas requisições permanece inexplicada, destacando a complexidade da Aplicação B e a necessidade de explorar fatores adicionais que podem influenciar

a demanda. Os resultados também reiteram que, mesmo com leves alterações nas métricas de erro e nos coeficientes de determinação, os modelos mantêm uma utilidade prática, sugerindo a importância de continuar a ajustar e refinar os modelos para melhor capturar a demanda por requisições.

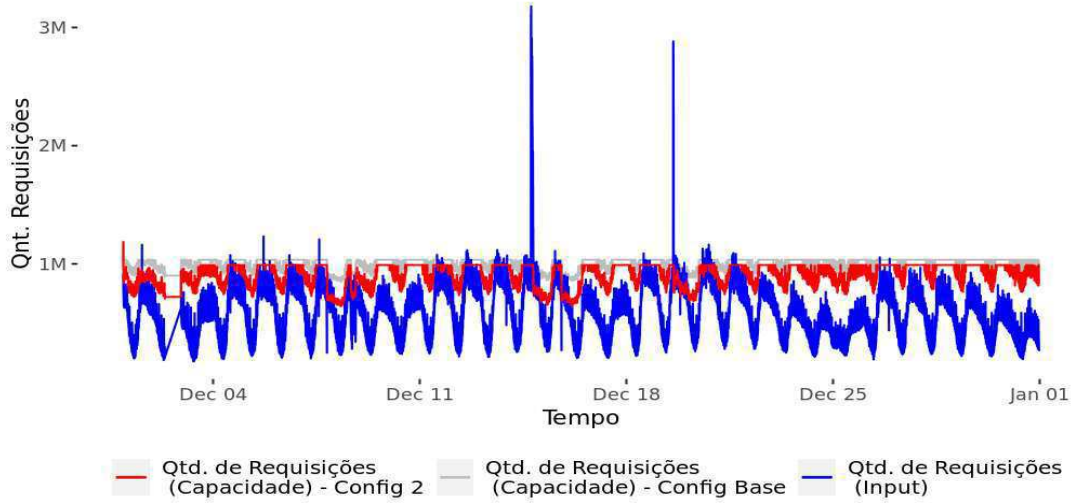

Figura 7.17: Análise temporal da capacidade de processamento de requisições, utilizando os dados da Aplicação B e simulando com a Configuração 2.

Avaliação Quantitativa de Desempenho e Eficiência de Recursos

A Tabela 7.4 apresenta as métricas de desempenho da simulação utilizando a Configuração Base, 2 e 3.

Ao analisar as métricas de desempenho e custo das Configurações Base, 2 e 3 da Aplicação B, conforme detalhado na Tabela 7.4, observa-se que a acurácia do subprovisionamento permaneceu constante em 1% entre as Configurações Base e 2, melhorando para 0% na Configuração 3. Essa melhoria indica a eliminação do subprovisionamento na Configuração 3, otimizando o uso dos recursos. Em contrapartida, a acurácia do superprovisionamento, que mede a alocação excessiva de recursos, subiu de 7367% na Configuração Base para 9297% na Configuração 3, evidenciando um aumento significativo no superprovisionamento e, por consequência, um potencial aumento no desperdício de recursos.

O tempo de subprovisionamento reduziu de 0.12% para 0.08%, enquanto o tempo de

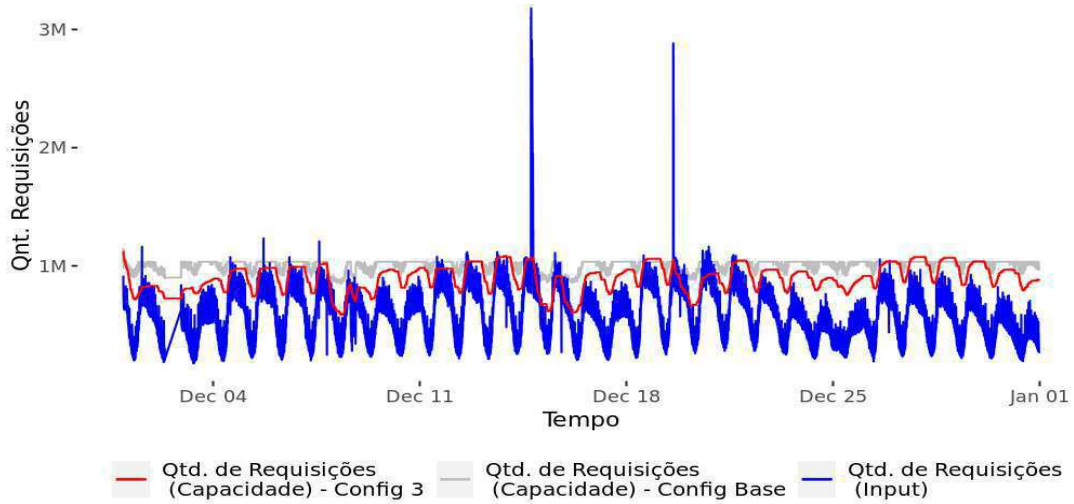

Figura 7.18: Análise temporal da capacidade de processamento de requisições, utilizando os dados da Aplicação B e simulando com a Configuração 3.

superprovisionamento viu um leve aumento de 99.89% para 99.92%, indicando que a Configuração 3 manteve os recursos mais frequentemente ativos, embora muitas vezes em excesso.

A métrica TAR (Total Absolute Residual) diminuiu de 67.27% na Configuração Base para 63.57% na Configuração 3, refletindo uma eficiência ligeiramente melhorada na utilização dos recursos.

Do ponto de vista financeiro, a transição para a Configuração 3 resultou em um aumento no custo estimado de 2979.19 USD para 3313.41 USD e no desperdício estimado de 1161.94 USD para 1489.94 USD. Estes aumentos indicam uma gestão de recursos menos eficiente na Configuração 3, especialmente devido ao maior superprovisionamento observado.

7.2.3 Discussão das modificações na configuração da Aplicação B

Os resultados obtidos para a Aplicação B, sob as diferentes configurações de provisionamento automático, oferecem um panorama detalhado das implicações de ajustes nos limiares e políticas de provisionamento de recursos. A análise da Configuração 2, onde o limiar alvo foi elevado para 65%, revelou uma redução no desperdício de recursos, embora essa configuração ainda não tenha alcançado o ideal de eficiência de recursos. Este resultado sugere que ajustes incrementais no limiar podem ter efeitos positivos.

Métrica	Config. Base	Config. 2	Config. 3
Acurácia subprovisionamento	1%	1%	0%
Acurácia superprovisionamento	7367%	5711%	9 297%
Tempo de subprovisionamento	0.12%	0.15%	0.08%
Tempo de superprovisionamento	99.89%	99.85%	99.92%
TAR	67.27%	62.73%	63.57%
Desperdício estimado [%]	39%	34.11%	44.97%

Tabela 7.4: Comparativo das métricas de eficiência e custo para avaliar ações de provisionamento automático, utilizando os dados da Aplicação B com as Configurações Base, 2 e 3.

A Configuração 3 introduziu uma abordagem mais dinâmica na alocação de recursos, ajustando-se mais estreitamente à demanda variável. Embora isso tenha garantido a cobertura dos picos de demanda sem risco de subprovisionamento, também levou a um superprovisionamento e a um aumento associado nos custos operacionais. Isso destaca a complexidade de encontrar um equilíbrio adequado entre a garantia de disponibilidade de recursos e a gestão eficiente dos mesmos, sugerindo que para encontrar uma configuração de política de provisionamento que melhor se adapte a demanda, é necessário a configuração de diversos parâmetros.

A avaliação quantitativa de desempenho e eficiência de recursos mostrou que, enquanto algumas métricas melhoraram. Além disso, a análise dos limiares que acionam ações de provisionamento sugeriu que há espaço para melhorar a resposta do sistema às mudanças na utilização. Em particular, a utilização permanecendo abaixo do limiar alvo por uma porção significativa do tempo sinaliza que o sistema poderia beneficiar-se de uma política mais reativa e menos conservadora.

Os aumentos modestos no MAE e RMSE, observados após a implementação da Configuração 2 e mantidos na Configuração 3, indicam que as mudanças no provisionamento não melhoraram substancialmente a precisão do modelo que estima a capacidade de quantas requisições o sistema consegue atender. Isso reforça a ideia de que fatores adicionais, não capturados pelas configurações atuais, estão influenciando a demanda e devem ser investigados.

Adicionalmente, os resultados do coeficiente de determinação (R^2) abaixo de 0.5 para

as simulações reiteram que há uma variabilidade significativa nas requisições que não está sendo explicada pelos modelos atuais. O que pode ter ocorrido pelo fato de não termos considerado o tamanho de cada requisição, visto que não foi disponibilizada essa informação. Isso aponta também para a necessidade de uma análise mais aprofundada em buscas de outras variáveis que possam influenciar a estimativa da capacidade, a fim de desenvolver modelos mais robustos que possam efetivamente guiar as estratégias de provisionamento.

Capítulo 8

Conclusões

Diante da dificuldade apresentada neste trabalho em selecionar parâmetros adequados para configurar as políticas de provisionamento automático, este estudo dedica-se a propor um fluxo de estudo que auxilie na avaliação de diferentes opções de configuração de provisionamento automático. O fluxo de estudo proposto envolve a simulação da carga de trabalho de uma aplicação em diversas configurações, seguida pela análise comparativa utilizando métricas específicas e técnicas de visualização.

A análise dos resultados deste estudo revela que mudanças nas configurações de provisionamento automático têm um impacto substancial na alocação de recursos, afetando diretamente os custos operacionais e os SLOs. É importante destacar que, embora uma configuração mais eficiente em termos de custo possa ser alcançada, isso não garante necessariamente a conformidade com os SLOs estabelecidos

As análises das Aplicações A e B reforçam o desafio de identificar configurações ideais que correspondam efetivamente à demanda. Fica evidente que cada ajuste nos parâmetros influencia diretamente a quantidade de recursos provisionados, impactando significativamente nos custos e potencial desperdício financeiro. Com o apoio da ferramenta mencionada, conseguimos identificar diferentes comportamentos nas aplicações, possibilitando também destacar configurações que obtiveram melhor custo-benefício dentre as analisadas,

A Configuração 4 para a Aplicação A se mostrou eficaz, atingindo mais equilíbrio entre a eficiência no uso de recursos e a redução de custos. Por outro lado, apesar da Configuração 5 oferecer custos mais baixos e maior precisão, ela apresenta o risco de falhar ocasionalmente em atender à demanda, destacando a importância de considerar os impactos na experiência

do usuário quando há reduções de custos que podem comprometer, mesmo que brevemente, a SLO.

Quanto à Aplicação B, os resultados evidenciam as consequências de adotar políticas distintas sem uma seleção criteriosa de parâmetros. Utilizar uma política alternativa com parâmetros escolhidos aleatoriamente resultou em superprovisionamento e aumento dos custos, ressaltando a necessidade de um fluxo de estudo e ferramenta que permitam testar variados parâmetros sem implementação direta em ambiente de produção.

Além disso, os resultados indicam que, para algumas aplicações, ajustar limites, modificar a quantidade de cores em ações de expansão de recursos e alterar os valores para a capacidade máxima, período de avaliação e tempo de espera para ações de provisionamento são estratégias eficazes. Estas podem reduzir custos financeiros com impacto mínimo ou inexistente no desempenho das aplicações. No entanto, a alteração de políticas sem uma escolha planejada de parâmetros pode levar a um aumento no superprovisionamento e, consequentemente, nos custos.

Este cenário evidencia a relevância do fluxo de estudo proposto neste estudo. Através dele, foi possível analisar e comparar diversas configurações em termos de custo e desempenho, eliminando a necessidade de realizar testes em um ambiente de produção. Essa abordagem não apenas simplifica o processo de avaliação, mas também oferece percepções importantes para a melhor utilização de recursos, contribuindo significativamente para uma escolha mais acertada de parâmetros da configuração.

Contudo, algumas limitações persistem neste estudo, como a ausência de uma equação com R^2 superior a 0.5 aplicável a todas as aplicações para calcular a capacidade do sistema em suportar a requisições. Essa limitação pode ser atribuída, em parte, à restrição dos dados disponíveis, que levou à consideração de todas as requisições como sendo de tamanho e tipo único. Portanto, este trabalho contribui para a tomada de decisões mais assertivas, permitindo que, a partir de uma amostra de dados, diversos parâmetros sejam testados e analisados em termos de desempenho e custo.

Embora as configurações de provisionamento automático tenham demonstrado potencial para melhorar a eficiência dos recursos, ainda há desafios consideráveis a serem superados. A busca por um modelo de provisionamento que balanceie de maneira ótima a disponibilidade de recursos, o custo e a experiência do usuário é uma área de pesquisa contínua.

Capítulo 9

Trabalhos Futuros

Apartir do que foi elaborado nesse estudo, algumas direções podem ser seguidas com o intuito de aprimorar significativamente o campo do provisionamento automático de recursos em ambientes de nuvem. Uma dessas direções é a automação na escolha de parâmetros. A ideia é desenvolver métodos que permitam a seleção automática, para que o ferramental possa simular distintos cenários e retornar o que obteve melhores métricas. Isso não apenas aumentaria a eficiência, mas também reduziria a necessidade de intervenção manual, otimizando a alocação de recursos de forma mais adaptativa e responsiva.

Outro foco importante é a capacidade de prever os parâmetros de provisionamento com base nas características específicas de uma aplicação é uma área que merece atenção. Isso envolve o uso de técnicas de aprendizado de máquina e análise de dados para identificar padrões e correlações que possam indicar os parâmetros de provisionamento ideais para uma determinada aplicação, considerando fatores como tipo de carga de trabalho, padrões de uso, requisitos de desempenho e outros.

Por fim, a replicação deste estudo com outra fonte de dados também é uma perspectiva valiosa. Isso poderia validar os achados atuais, além de fornecer percepções adicionais sobre a aplicabilidade e eficácia das estratégias de provisionamento em diferentes cenários. Essa abordagem ajudaria a entender melhor a generalizabilidade dos métodos propostos e a adaptá-los a diferentes tipos de ambientes e requisitos de aplicações.

Bibliografia

[1] Mohammad Sadegh Aslanpour, Mostafa Ghobaei-Arani, and Adel Nadjaran Toosi. Auto-scaling web applications in clouds: A cost-aware approach. *Journal of Network and Computer Applications*, 95:26–41, 2017.

[2] Mohsen Attaran and Jeremy Woods. Cloud computing technology: improving small business performance using the internet. *Journal of Small Business & Entrepreneurship*, 31(6):495–519, 2019.

[3] Ataollah Fatahi Baarzi, Timothy Zhu, and Bhuvan Urgaonkar. Burscale: Using burstable instances for cost-effective autoscaling in the public cloud. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC '19, page 126–138, New York, NY, USA, 2019. Association for Computing Machinery.

[4] Abmar Barros, Francisco Brasileiro, Giovanni Farias, Francisco Germano, Marcos Nóbrega, Ana C Ribeiro, Igor Silva, and Leticia Teixeira. Using fogbow to federate private clouds. *Salao de Ferramentas do XXXIII SBRC*, 2015.

[5] André Bauer, Johannes Grohmann, Nikolas Herbst, and Samuel Kounev. On the value of service demand estimation for auto-scaling. In Reinhard German, Kai-Steffen Hielscher, and Udo R. Krieger, editors, *Measurement, Modelling and Evaluation of Computing Systems*, pages 142–156, Cham, 2018. Springer International Publishing.

[6] André Bauer, Nikolas Herbst, Simon Spinner, Ahmed Ali-Eldin, and Samuel Kounev. Chameleon: A hybrid, proactive auto-scaling mechanism on a level-playing field. *IEEE Transactions on Parallel and Distributed Systems*, 30(4):800–813, 2018.

[7] Jeremy Ellman, Nathan Lee, and Nanlin Jin. Cloud computing deployment: a cost-modelling case-study. *Wireless Networks*, pages 1–8, 2018.

[8] Gartner. Gartner says worldwide iaas public cloud services revenue grew 30% in 2022, exceeding $100 billion for the first time, 2023.

[9] Nikolas Herbst, Rouven Krebs, Giorgos Oikonomou, George Kousiouris, Athanasia Evangelinou, Alexandru Iosup, and Samuel Kounev. Ready for rain? a view from spec research on the future of cloud metrics, 2016.

[10] Nikolas Roman Herbst, Samuel Kounev, Andreas Weber, and Henning Groenda. Bungee: an elasticity benchmark for self-adaptive iaas cloud environments. In *2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*, pages 46–56. IEEE, 2015.

[11] Gaopan Huang, Songyun Wang, Mingming Zhang, Yefei Li, Zhuzhong Qian, Yuan Chen, and Sheng Zhang. Auto scaling virtual machines for web applications with queueing theory. In *2016 3rd International conference on systems and informatics (ICSAI)*, pages 433–438. IEEE, 2016.

[12] Alexey Ilyushkin, Ahmed Ali-Eldin, Nikolas Herbst, André Bauer, Alessandro V. Papadopoulos, Dick Epema, and Alexandru Iosup. An experimental performance evaluation of autoscalers for complex workflows. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 3(2), apr 2018.

[13] Waheed Iqbal, Abdelkarim Erradi, Muhammad Abdullah, and Arif Mahmood. Predictive auto-scaling of multi-tier applications using performance varying cloud resources. *IEEE Transactions on Cloud Computing*, 10(1):595–607, 2022.

[14] Jitendra Kumar and Ashutosh Kumar Singh. Workload prediction in cloud using artificial neural network and adaptive differential evolution. *Future Generation Computer Systems*, 81:41–52, 2018.

[15] Paulo Ditarso Maciel, Francisco Brasileiro, Raquel Lopes, Marcus Carvalho, and Miranda Mowbray. Avaliando o impacto do planejamento de contratos de longo prazo no gerenciamento de uma infraestrutura de TI híbrida. In *12º Simpósio Internacional IFIP/IEEE sobre Gerenciamento Integrado de Redes (IM 2011) e Workshops*, pages 89–96, 2011.

[16] Rafael Moreno-Vozmediano, Rubén S. Montero, Eduardo Huedo, and Ignacio M. Llorente. Efficient resource provisioning for elastic cloud services based on machine learning techniques. *Journal of Cloud Computing*, 8(1):5, Apr 2019.

[17] Marco AS Netto, Carlos Cardonha, Renato LF Cunha, and Marcos D Assunçao. Evaluating auto-scaling strategies for cloud computing environments. In *2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*, pages 187–196. IEEE, 2014.

[18] Ali Yadavar Nikravesh, Samuel A. Ajila, and Chung-Horng Lung. An autonomic prediction suite for cloud resource provisioning. *Journal of Cloud Computing*, 6(1):3, Feb 2017.

[19] Joe H. Novak, Sneha Kumar Kasera, and Ryan Stutsman. Cloud functions for fast and robust resource auto-scaling. In *2019 11th International Conference on Communication Systems & Networks (COMSNETS)*, pages 133–140, 2019.

[20] Alessandro Vittorio Papadopoulos, Ahmed Ali-Eldin, Karl-Erik Årzén, Johan Tordsson, and Erik Elmroth. Peas: A performance evaluation framework for auto-scaling strategies in cloud applications. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 1(4), aug 2016.

[21] Alessandro Vittorio Papadopoulos, Laurens Versluis, André Bauer, Nikolas Herbst, Jóakim von Kistowski, Ahmed Ali-Eldin, Cristina L. Abad, José Nelson Amaral, Petr Tůma, and Alexandru Iosup. Methodological principles for reproducible performance evaluation in cloud computing. *IEEE Transactions on Software Engineering*, 47(8):1528–1543, 2021.

[22] Vladimir Podolskiy, Anshul Jindal, and Michael Gerndt. Iaas reactive autoscaling performance challenges. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 954–957, 2018.

[23] Mariana Mendes Silva. Auto-scaling em uma empresa de e-commerce: Um estudo de caso. Master's thesis, Universidade Federal de Campina Grande, Paraíba, 2022. Dissertação de mestrado.

[24] Martin Straesser, Simon Eismann, Jóakim von Kistowski, André Bauer, and Samuel Kounev. Autoscaler evaluation and configuration: A practitioner's guideline. In *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering*, ICPE '23, pages 31–41, New York, NY, USA, 2023. Association for Computing Machinery.

[25] Martin Straesser, Simon Eismann, Jóakim von Kistowski, André Bauer, and Samuel Kounev. Autoscaler evaluation and configuration: A practitioner's guideline. In *Proceedings of the 2023 ACM/SPEC International Conference on Performance Engineering*, ICPE '23, page 31–41, New York, NY, USA, 2023. Association for Computing Machinery.

[26] Martin Straesser, Johannes Grohmann, Jóakim von Kistowski, Simon Eismann, André Bauer, and Samuel Kounev. Why is it not solved yet? challenges for production-a hybrid, autoscaling. In *Proceedings of the 2022 ACM/SPEC on International Conference on Performance Engineering*, pages 105–115, 2022.

[27] Yang syu and Chien-Min Wang. Modeling and forecasting http requests-based cloud workloads using autoregressive artificial neural networks. In *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*, pages 139–145, 2018.

[28] Fei Xu, Fangming Liu, Hai Jin, and Athanasios V. Vasilakos. Managing performance overhead of virtual machines in cloud computing: A survey, state of the art, and future directions. *Proceedings of the IEEE*, 102(1):11–31, 2014.

[29] Mahendra Pratap Yadav, Rohit, and Dharmendra Kumar Yadav. Resource provisioning through machine learning in cloud services. *Arabian Journal for Science and Engineering*, 47(2):1483–1505, Feb 2022.

[30] Yonghua Zhu, Weilin Zhang, Yihai Chen, and Honghao Gao. A novel approach to workload prediction using attention-based lstm encoder-decoder network in cloud environment. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):274, Dec 2019.