Programa Interdisciplinar de Pós-Graduação em

Computação Aplicada

Mestrado Acadêmico

Leandro Andrioli

Elastic-RAN: Um modelo de elasticidade multinível com grão adaptativo para Cloud Radio Access Network

São Leopoldo, 2018

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

LEANDRO ANDRIOLI

ELASTIC-RAN: UM MODELO DE ELASTICIDADE MULTINÍVEL COM GRÃO ADAPTATIVO PARA CLOUD RADIO ACCESS NETWORK

São Leopoldo
2018

Leandro Andrioli

ELASTIC-RAN: UM MODELO DE ELASTICIDADE MULTINÍVEL COM GRÃO ADAPTATIVO PARA CLOUD RADIO ACCESS NETWORK

Dissertação apresentada como requisito parcial para a obtenção do título de Mestre pelo Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo
2018

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)

Andrioli, Leandro

Elastic-RAN: Um modelo de elasticidade multinível com grão adaptativo para Cloud Radio Access Network / Leandro Andrioli — 2018.

108 f.: il.; 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, São Leopoldo, 2018.

"Orientador: Prof. Dr. Rodrigo da Rosa Righi, Unidade Acadêmica de Pesquisa e Pós-Graduação".

1. C-RAN. 2. Computação em Nuvem. 3. Elasticidade Multinível. 4. Grão Elástico Adaptativo. I. Título.

CDU 004.62

Vanessa Borges Nunes — CRB 10/1556

Leandro Andrioli

Elastic-RAN: Um modelo de elasticidade multinível com grão adaptativo para Cloud Radio Access Networks

Dissertação apresentada à Universidade do Vale do Rio dos Sinos – Unisinos, como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Aprovado em 31 de julho de 2018.

BANCA EXAMINADORA

Prof. Dr. Rodrigo da Rosa Righi - Unisinos

Prof. Dr. Cristiano André da Costa - Unisinos

Prof. Dr. Antônio Marcos Alberti - Inatel

Prof. Dr. Rodrigo da Rosa Righi (Orientador)

Visto e permitida a impressão
São Leopoldo, 2018

Prof. Dr. Rodrigo da Rosa Righi
Coordenador PPG em Computação Aplicada

AGRADECIMENTOS

Se você está lendo esta página é porque mais um grande desafio foi superado. E não foi fácil chegar até aqui. Do processo seletivo, passando pela aprovação até a entrega da presente dissertação, foi um longo caminho percorrido. Nada foi fácil, nem tampouco tranquilo. Mas com muita dedicação, motivação e apoio de familiares, professores e amigos consegui concluir mais essa etapa.

Neste momento é difícil agradecer todas as pessoas que de algum modo fizeram ou fazem parte da minha vida, está sendo um período de dedicação e comprometimento, repleto de desafios e oportunidades, onde fui em busca de mais uma importante etapa em minha vida.

Primeiramente agradeço a Deus, por proporcionar todas as oportunidades que tive e que tornaram minha vida excelente, com uma família maravilhosa, amigos sinceros e, ainda, por renovar a cada momento a minha força e disposição ao longo dessa jornada.

Agradeço a todos os meus professores que de diversas maneiras tiveram uma imensa contribuição na minha formação, mas principalmente a meu orientador, Prof. Dr. Rodrigo da Rosa Righi, que através de sua dedicação e incentivo, me guiou e proporcionou o suporte necessário para o desenvolvimento dessa dissertação. Uma parceria que vem desde a minha graduação... que sigamos em frente com essa parceria para nossos futuros estudos e pesquisas.

Aos meus pais, Roque e Rosane, muito obrigado por toda educação e valores que sempre me deram. Pelo amor, incentivo e apoio incondicional em todos os momentos. Por acreditarem em mim, no meu potencial e por proporcionarem a possibilidade de tornar o meu sonho de formação uma realidade. Vocês são a minha base e essa conquista também é de vocês!

A minha irmã, Fernanda, sempre disposta a me orientar e oferecer suporte para a continuação da minha jornada em diferentes aspectos, obrigado por tudo magrinha.

A Talita, sempre me incentivando, motivando e apoiando nos mais variados momentos. Obrigado pela sua incansável boa vontade em me ajudar, e me incentivar a prosseguir. Por todas revisões, por todo o teu carinho, atenção, os saborosos cafés nos dias de estudo e, principalmente, a tua parceria em todas as horas vibrando, com as minhas conquistas. Você é incrível meu amor!

Agradeço aos meus amigos por acreditarem em mim e sempre apoiarem minhas decisões, mesmo que essas tenham nos deixado afastados em diversos momentos de lazer. Eu sei que vocês entendem, é apenas uma fase.

Enfim, muito obrigado de coração a todos que de alguma maneira contribuíram para esta jornada. Continuarei a constante busca por evolução! Sempre em frente! E que venham os próximos desafios!

"*Escolha um trabalho que você ame e não terás que trabalhar um único dia em sua vida*"
- Confúcio

RESUMO

Até o ano de 2020, espera-se que a área de cobertura das redes de celulares aumente em 10 vezes, com mais de 50 bilhões de dispositivos conectados, suportando 100 vezes mais equipamentos de usuários e elevando a capacidade da taxa de dados em 1000 vezes. Tal circunstância gerará um aumento massivo no tráfego de dados, fomentando o desenvolvimento da 5G e fazendo com que a indústria e as iniciativas científicas passem a voltar seus esforços para atender a essa demanda. Ganha força, então, as pesquisas relacionadas a *Cloud Radio Access Networks* (C-RANs), uma arquitetura que consolida as *base stations* (BSs) para um ponto centralizado na nuvem, mudando a ideia de atuar com recursos fixos e limitados, na medida em que se beneficia de uma das características chave da Computação em Nuvem: a elasticidade de recursos. Um dos grandes desafios na arquitetura C-RAN reside na complexidade em orquestrar todos esses recursos computacionais de forma que o processamento das requisições seja realizado com alto desempenho e com o menor custo de infraestrutura possível. Diante de todo esse contexto, a presente dissertação busca desenvolver o modelo Elastic-RAN, propondo um conceito de elasticidade multinível não bloqueante, com orquestração automática de recursos através da coordenação de BBU Pools e seus BBUs, junto a um mecanismo de grão elástico adaptativo. A elasticidade multinível não bloqueante permite controlar o nível de BBU Pool (máquina física), haja vista o alto volume de tráfego e a distância máxima sugerida entre as antenas e os pools, e o nível de BBU (máquina virtual), em razão do alto processamento de CPU e memória necessária para as requisições, de modo a não penalizar os processamentos correntes. O mecanismo de grão elástico adaptativo permite provisionar e mapear os recursos sob demanda e em tempo de execução, considerando o uso corrente dos recursos, para que cada ação elástica seja executada com um grão próximo das necessidades correntes de processamento. O modelo Elastic-RAN foi avaliado por intermédio de experimentos que simularam diferentes perfis de cargas, os quais são executados em uma aplicação intensiva de CPU e de tráfego na rede, explorando a transferência de *streamings* e processando decodificação de blocos. Como resultados, foi possível constatar que o Elastic-RAN pode atingir ganhos que vão de 4% a 26%, em relação aos custos de execução, quando comparado à abordagem de elasticidade tradicional. Além disso, obteve melhor eficiência para todos os perfis de carga e reduziu em até 55% a quantidade de operações elásticas necessárias. Outrossim, frente a abordagem sem elasticidade, os ganhos de custos foram ainda superiores, ficando entre 51% e 70%.

Palavras-chave: C-RAN. Computação em Nuvem. Elasticidade Multinível. Grão Elástico Adaptativo.

ABSTRACT

It is expected that, by 2020, cell phone networks will have been increased 10 times their coverage area, with more than 50 billion connected devices, supporting 100 times more user equipment and increasing data rate capacity by 1000 times. This will lead to a massive increase in data traffic, fostering the development of 5G and making industry and scientific initiatives turn their efforts to meet this demand. In this scenario, Cloud Radio Access Networks (C-RANs) based researches, an architecture that consolidates base stations (BSs) to a cloud-centric point, are gaining momentum, changing the idea of fixed and limited resources, as it benefits from one of the key features of Cloud Computing: resource elasticity. One of the major challenges in C-RAN architecture lies in the high complexity of orchestrating all of these computational resources in order to perform the requests processing with high performance and the lowest possible infrastructure cost. Considering this context, the present dissertation seeks to develop the Elastic-RAN model, proposing a multilevel non-blocking elasticity concept, with automatic orchestration of resources through the coordination of BBU Pools and their BBUs, with an adaptive elastic grain mechanism. The multilevel non-blocking elasticity allows it control the level of BBU Pool (physical machine), given the high volume of traffic and the suggested maximum distance between antennas and pools, and the level of BBU (virtual machine), due to the high CPU processing and memory required for the requests, so as not to penalize the current processing. The adaptive elastic grain mechanism allows the provisioning and mapping of resources on demand and at runtime, considering the current use of resources, so that each elastic action is performed with a grain close to the current processing needs. The Elastic-RAN model was evaluated through experiments that simulated different load profiles, which are executed in an intensive CPU and network traffic application, exploiting the transfer of streamings and processing block decoding. As a result, it was possible to observe that Elastic-RAN may achieve gains ranging from 4 % to 26 %, in relation to execution costs, when compared to the traditional elasticity approach. In addition, it achieved better efficiency for all load profiles and reduced by 55 % the amount of elastic operations required. Also, given the non-elasticity approach, cost gains were even higher, going from 51 % to 70 %.

Keywords: C-RAN. Cloud Computing. Multilevel Elasticity. Adaptive Elastic Grain.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

5G	Quinta Geração de Redes Móveis ou Quinta Geração de Internet Móvel
API	Application Programming Interface
BBU	Baseband Unit
BS	Base Station
CAPEX	Capital Expenditure
C-RAN	Cloud Radio Access Network
GSM	Groupe Special Mobile
HetNet	Heterogeneous Networks
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
KM	Quilômetro
LTE	Long Term Evolution
NFS	Network File System
NIST	National Institute of Standards and Technology
OPEX	Operational Expenditure
PaaS	Platform as a Service
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RRH	Remote Radio Head
SaaS	Software as a Service
SLA	Service Level Agreement

SUMÁRIO

1 INTRODUÇÃO

A quinta geração de redes de celular (5G) gerará um massivo aumento no tráfego de dados, com mais de 50 bilhões de dispositivos conectados até o ano de 2020 (ERICSSON, 2011). Esse aumento será proveniente, em sua grande maioria, de *smartphones*, *tablets* e dispositivos inteligentes. Por conta disso, tanto a indústria quanto as iniciativas científicas passaram a voltar esforços para essa nova geração de redes móveis, explorando principalmente o aperfeiçoamento das *Radio Access Networks* (RANs), que são redes responsáveis por implementar a tecnologia de rádio em sistemas de comunicação. Em uma arquitetura RAN, os equipamentos dos usuários se conectam aos nós da RAN através de conexões sem fio para realizarem comunicações.

Com o escopo de solucionar alguns problemas enfrentados na RAN tradicional, uma nova arquitetura foi proposta, a *Cloud Radio Access Network* (C-RAN), que, ao invés tratar as requisições em hardware, virtualiza as funcionalidades da BS, movendo o processamento para a nuvem (MOBILE, 2011). É justamente o processamento centralizado dos sinais na nuvem que viabiliza a exploração de um dos principais atributos, qual seja, a elasticidade (ROSA RIGHI, 2013). No ponto, importa mencionar que tal característica possibilita o provisionamento ou liberação de recursos de forma flexível – *on-the-fly* – conforme a variação das demandas correntes (MELL; GRANCE, 2011).

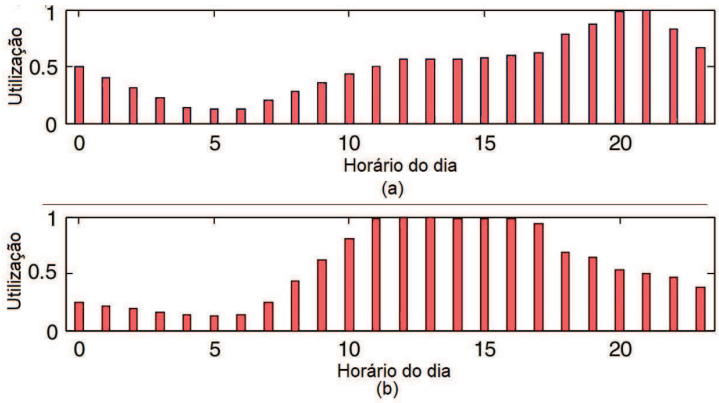
No contexto da C-RAN, o recurso de elasticidade permite o provisionamento de recursos sob demanda para o *Baseband Unit Pool* (BBU Pool), propiciando melhor desempenho no processamento dos sinais e na utilização eficiente dos recursos, além de acarretar uma redução de custos. A melhoria no desempenho é dada pela capacidade de alocação *on-the-fly* de novos recursos computacionais, circunstância que também é válida para a melhor utilização de recursos, já que em momentos de alta demanda ocorre maior provisionamento, enquanto que em momentos de baixa demanda é possível reduzi-los para evitar a ociosidade dos mesmos (RAZA et al., 2016). Como consequência, os custos relacionados a alocação de recursos e os gastos energéticos podem ser reduzidos (SAH; JOSHI, 2014).

Podemos, então, destacar como sendo os principais benefícios de uma arquitetura C-RAN: (i) economia nas despesas operacionais devido à manutenção centralizada; (ii) melhora no desempenho da rede devido a técnicas avançadas de processamento de sinal coordenado; (iii) processamento das tarefas se torna mais ágil e eficiente; e (iv) redução nos gastos ao alocar diferentes recursos conforme às variações de carga (PARK et al., 2013; BHAUMIK et al., 2012).

1.1 Motivação

Dado o incontroverso aumento da utilização de comunicações móveis nos últimos anos, a arquitetura RAN, tradicionalmente utilizada para processar requisições de usuários, passou a necessitar de maior capacidade para suportar altas taxas de dados provenientes de ambientes de alta mobilidade (CMRI, 2010; YASEEN; AL-KHALIDI; AL-RAWESHIDY, 2017). Nessa

Figura 1 – Flutuação da carga de tráfego na BS durante o período de um dia: (a) área residencial; (b) área empresarial.



Fonte: Adaptado de Raza et al. (2016).

arquitetura, as base stations (BSs) precisam ter uma pré-configuração com alta capacidade para lidar com os picos de uso sem interrupções. Acontece que, em cenários práticos de redes móveis, o tráfego raramente está em seu pico, pois a carga muda gradualmente ao longo do dia, sendo possível observar as alterações em um padrão geométrico de tempo, denominado Tidal Effect (RAZA et al., 2016). A Figura 1 retrata a flutuação da carga na BS, demonstrando que, durante a noite, as BSs nas áreas residenciais são muito utilizadas, enquanto que as BSs nas áreas empresarias permanecem ociosas e consumindo muita energia (CHECKO et al., 2014).

Passou a ser imperativo resolver esse problema, de modo a liberar os recursos e diminuir custos de processamento e energia, especialmente porque, de acordo com informações divulgadas pela Cisco, a transformação digital global continuará a ter um impacto significativo nas demandas e requisitos das redes, mormente em razão do aumento de usuários da Internet. Previsões apontam que o número de consumidores dos serviços de Internet saltará de 3,3 para 4,6 bilhões, o que representa o percentual de 58% da população mundial. Dispositivos móveis pessoais, conexões máquina-máquina (M2M), avanços nas aplicações de Internet das Coisas em casas conectadas, carros inteligentes, saúde, e outros serviços de nova geração, vão impulsionar esse crescimento. Projeta-se, por conseguinte, que as redes Wi-Fi e os dispositivos conectados por celular gerarão 73% do tráfego total da Internet até 2021 (VNI, 2017).

Diante desse cenário, propôs-se a C-RAN, arquitetura que explora principalmente os recursos de elasticidade da computação em nuvem, tornando-se, por isso, mais eficiente no que diz respeito às questões de gerenciamento dos recursos computacionais. Tal arquitetura é formada por três componentes principais, quais sejam: o *remote radio head* (RRH), que são antenas que desempenham funções analógicas de frequência de rádio; o *baseband unit* (BBU), que são unidades para processamento do sinal digital; e o *fronthaul*, que é a conexão entre o BBU e o RRH (WANG et al., 2016). Nessa arquitetura, o processamento do sinal digital do RRH é executado em um BBU Pool localizado na nuvem, o qual encontra-se em uma distância máxima recomendada de aproximadamente até 40Km do RRH, devido a questões de latência, tempo

de resposta e perda de sinal (CHANCLOU et al., 2013). Assim, ocorre um provisionamento dinâmico de acordo com as necessidades correntes de tráfego (SIGWELE; PILLAI; HU, 2015), reduzindo os custos, em razão dos recursos computacionais serem alocados de forma mais eficiente, aumentando a economia de energia e compartilhando o processamento dos sinais nos BBUs para diferentes BSs (NIKAEIN et al., 2015; RAZA et al., 2016). Ademais, é possível gerenciar os recursos computacionais disponíveis, alocando-os durante os períodos de alto tráfego e realizando ações para reduzir a capacidade de processamento dos BBUs nos períodos de baixa.

Levar a efeito a arquitetura C-RAN não é, no entanto, tarefa trivial, pois há diversos desafios a serem superados, abrindo espaço para oportunidades de pesquisa na literatura. A exemplo, podemos citar a orquestração dos recursos, na qual cada adição ou remoção de instâncias exige uma reorganização de processos e uma atualização da topologia de comunicação. Conseguir lidar com a restruturação para melhor distribuir as requisições a serem executadas por cada um dos recursos compartilhados, de maneira que ocorra um balanceamento de carga entre os BBUs para um processamento ágil e com baixos custos de infraestrutura, é um procedimento complexo e, ao mesmo tempo, vital para manter o equilíbrio e a qualidade na entrega das requisições (MAROTTA et al., 2015; DUAN et al., 2017). Outro importante obstáculo a ser enfrentado liga-se a necessidade de lidar de forma mais eficiente com as diferentes demandas de carga, utilizando-se de abordagens que realizem operações elásticas, as quais devem ser precisas para as demandas correntes e com capacidade de tratar momentos de eventuais picos ou quedas repentinas, circunstância que pode gerar ações de falso-positivo ou falso-negativo (ROSA RIGHI et al., 2016a). Além disso, as operações elásticas antes mencionadas não podem penalizar a distribuição e processamento corrente de tarefas, de modo a evitar perdas de desempenho nas entregas e a degradação na qualidade de entrega do serviço.

1.2 Questão de Pesquisa

Tendo em vista os diversos desafios das abordagens atuais em C-RAN para orquestrar e compartilhar os recursos, explorando a elasticidade da nuvem de forma automática e eficiente, o modelo proposto nesta dissertação busca responder a seguinte questão de pesquisa:

Como seria um modelo de elasticidade multinível para C-RAN com orquestração automática de recursos capaz de adaptar o grão de elasticidade para provisionar tanto BBU Pools quanto BBUs de forma não bloqueante, com desempenho e o menor custo de infraestrutura possível?

A ideia é propor um modelo que se baseia no uso dos recursos da rede e computacionais, capaz de realizar o provisionamento ou liberação de recursos de forma reativa, automática e adaptativa. A orquestração automática tem como ponto de partida regras previamente definidas e torna o sistema autônomo para realizar as ações elásticas sem intervenção manual. Já a elasticidade multinível se refere à orquestração dos recursos nos BBU Pools e nos BBUs de

forma individual. A forma de elasticidade não bloqueante vai em direção a realização de operações de maneira assíncrona, no intuito de não impactar no processamento das tarefas durante a reorganização de recursos. Por fim, adaptar o grão de elasticidade se refere ao redimensionamento do tamanho do grão em cada ação elástica, considerando o uso corrente dos recursos e o comportamento das cargas. Isso contribui para o aumento da reatividade ao permitir que o sistema responda mais rapidamente as diferentes variações de carga quando comparado com a abordagem de elasticidade tradicional.

1.3 Objetivos

Buscando proporcionar elasticidade em uma nuvem computacional para arquiteturas C-RANs, a presente dissertação propõe o modelo chamado Elastic-RAN, que oferece um orquestrador que coordena os BBU Pools (máquinas físicas) e os seus BBUs (máquinas virtuais) de forma individual, permitindo, assim, de forma reativa, a tomada de decisão sobre qual tratamento será dado aos recursos da nuvem. Essas rotinas ocorrem de forma automática e transparente aos provedores da C-RAN, pois o Elastic-RAN realiza o provisionamento e liberação de recursos para as diferentes demandas que ocorrem durante o dia de forma autônoma e não bloqueante. Ou seja, essas mudanças na infraestrutura dos BBUs e atualizações na topologia da rede não interrompem ou interferem a execução do processamento das outras instâncias.

Considerando esses pontos, o objetivo principal dessa dissertação é: *Elaborar um modelo de elasticidade multinível não bloqueante para C-RAN, com orquestração automática de recursos através da coordenação de BBU Pools e seus BBUs junto a um mecanismo de elasticidade com grão adaptativo, que considera o uso corrente dos recursos, com capacidade de provisionar e mapear os recursos sob demanda e em tempo de execução, processando as tarefas com alta reatividade, agilidade e o menor custo de infraestrutura possível.*

Para atingir esse objetivo principal, foram estabelecidos os seguintes objetivos específicos:

- Analisar o estado da arte com base em critérios definidos para identificar lacunas em aberto referentes a abordagens de alto desempenho em arquiteturas C-RAN;
- Elaborar um modelo que supra as lacunas identificadas nos trabalhos relacionados;
- Propor uma abordagem para atuar com elasticidade multinível não bloqueante;
- Criar um mecanismo de adaptação do tamanho do grão de elasticidade conforme a utilização corrente dos recursos;
- Desenvolver uma aplicação intensiva de CPU e rede para simular diferentes perfis de carga;
- Avaliar o modelo através de testes em laboratório, executando a aplicação com diferentes cenários e parâmetros;

Figura 2 – Fluxo das etapas de desenvolvimento da pesquisa.

Fonte: elaborado pelo autor.

1.4 Plano de Pesquisa

Como se pode observar na Figura 2, o desenvolvimento da presente pesquisa foi dividido em seis etapas, iniciando-se com o estudo das teorias envolvidas no tema em exame, afim de elaborar o embasamento teórico. Após, passou-se ao levantamento de trabalhos relacionados com a matéria da pesquisa, mais especificamente com os objetivos apresentados. Realizou-se, então, uma análise com o objetivo de identificar as lacunas presentes no estado da arte, sendo que, somente ao fim dessa etapa, é que o modelo foi proposto e desenvolvido com foco em atender os objetivos elencados e responder a questão de pesquisa. Veja-se que o desenvolvimento do modelo incluiu os pontos não explorados que puderam ser identificados a partir da análise das lacunas antes mencionadas.

As implementações de protótipos foram realizadas na quarta etapa, na qual a parte técnica da pesquisa foi produzida. Com o desenvolvimento do modelo e dos protótipos, avançou-se para o quinto ciclo, no qual os testes do modelo foram realizados e os resultados foram analisados. Em seguida, ofereceu-se a possibilidade de finalizar ou propor novas adaptações no modelo, de modo que, no lugar de encerrar a pesquisa, seguiu-se para a sexta etapa, na qual melhorias e adaptações no modelo foram propostas, baseadas nas análises de resultados realizadas na fase anterior. Retornou-se, por fim, ao estágio de desenvolvimento do modelo (quinta etapa), com o fim de realizar modificações e adaptações propostas, encontrando, a pesquisa, seu termo com a escrita da presente dissertação.

1.5 Organização do Texto

Organizada em sete capítulos, a presente dissertação inicia apresentando os conceitos relacionados aos assuntos que norteiam a pesquisa, introduzindo tecnologias e conceitos que são utilizados para o entendimento e desenvolvimento do modelo proposto. Na sequência, no Capítulo 3, foram trazidos à discussão trabalhos relacionados, de maneira a apresentar o estado

da arte e as lacunas observadas envolvendo alto desempenho em arquiteturas C-RAN. Já no Capítulo 4 são apresentadas as decisões do modelo, sua arquitetura, o modelo de elasticidade utilizado, os principais algoritmos e técnicas envolvidos. A metodologia para a avaliação do modelo, no qual são expostos detalhes de implementação, cenários e parâmetros dos testes, é apresentada no quinto Capítulo; enquanto que a avaliação do modelo é apresentada no sexto capítulo, momento em que os resultados obtidos em laboratório são analisados. Por fim, no Capítulo 7 são apontadas conclusões sobre o trabalho, enfatizando as contribuições do modelo e novas oportunidades de pesquisa em aberto.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos relacionados ao estudo desenvolvido e está dividido em quatro seções, a fim de facilitar o entendimento.

2.1 Computação em nuvem

Por se tratar de um novo modelo de operações, que reúne um conjunto de tecnologias existentes para executar negócios de maneira diferente, a computação em nuvem pode ser interpretada conforme diferentes percepções (ZHANG; CHENG; BOUTABA, 2010). A par dessa informação, salienta-se que, para esta dissertação, foi adotada a definição dada pelo *National Institute of Standards and Technology* (NIST), pois repute-se que ela cobre todos os principais aspectos da computação em nuvem. De acordo com o NIST, computação em nuvem se trata de um modelo que permite o acesso à rede sob demanda para um conjunto compartilhado de recursos computacionais configuráveis que podem ser rapidamente providos e lançados com o mínimo de esforço de gerenciamento ou interação com o provedor de serviços (MELL; GRANCE, 2011). Esses recursos podem ser reconfiguráveis de forma dinâmica, de modo que consigam se ajustar a cargas variáveis. A otimização no uso dos recursos é explorado através de um modelo conhecido por "pague pelo uso", no qual as garantias de *quality of service* (QoS) são oferecidas por um provedor de infraestrutura através de um *Service Level Agreement* (SLA) (VAQUERO et al., 2009).

Pode-se destacar cinco características essenciais de uma nuvem computacional (MELL; GRANCE, 2011; SOSINSKY, 2011):

- Auto serviço sob demanda: o consumidor é capaz de provisionar por conta própria recursos de computação automaticamente e quando necessário, sem intervenção humana dos provedores de serviços;

- Amplo acesso por rede: todos os recursos estão disponíveis na rede e são acessados através de mecanismos padronizados, que promovem portabilidade de uso de dispositivos de plataformas heterogêneas, por exemplo, *smartphones* e *tablets*;

- Agrupamento de recursos: os recursos de computação do provedor são agrupados para atender a múltiplos consumidores com máquinas físicas e virtuais diferentes, dinamicamente atribuídos conforme a demanda dos consumidores;

- Elasticidade rápida: os recursos podem ser provisionados e liberados elasticamente, em alguns casos automaticamente, para rapidamente aumentar ou diminuir de acordo com a demanda corrente. Para o consumidor, os recursos disponíveis para provisionamento muitas vezes parecem ser ilimitados e podem ser alocados em qualquer quantidade e a qualquer tempo;

- Serviço mensurado: os sistemas na nuvem automaticamente controlam e otimizam o uso dos recursos através de medições em um nível de abstração apropriado para o tipo de serviço (como armazenamento, processamento, comunicação de rede e contas de usuário ativas). A utilização de recursos pode ser monitorada, controlada e informada, gerando uma transparência ao fornecedor e ao consumidor do serviço utilizado.

2.1.1 Modelos de Serviço

O modelo tradicional, em que as empresas compram servidores e softwares, tem passado por uma grande transformação desde a chegada da computação em nuvem, o que fez surgir três modelos de serviços predominantes: (i) infraestrutura como serviço (IaaS); (ii) plataforma como serviço (PaaS); e (iii) software como serviço (SaaS) (MELL; GRANCE, 2011; GIBSON et al., 2012; ZHANG; CHENG; BOUTABA, 2010).

No IaaS, aos usuários são oferecidos tanto recursos de hardware quanto de softwares, o que inclui servidores, armazenamento, acesso a rede e virtualização para permitir utilidades como serviços para os usuários. Assim, esses recursos são fornecidos aos consumidores sob demanda na forma de máquinas virtuais. Salienta-se que a virtualização dos recursos é uma das características mais marcantes deste modelo, pois permite ao consumidor processar seu próprio sistema operacional e realizar o gerenciamento de suas máquinas virtuais que são executadas a partir da infraestrutura contratada.

O PaaS, por sua vez, define-se como sendo um provedor que oferece acesso à APIs, linguagens de programação e desenvolvimento de *middleware* que permite que os assinantes desenvolvam aplicativos personalizados sem instalar ou configurar o ambiente de programação. O consumidor tem acesso a estas plataformas com objetivo de criar e implementar aplicações, enquanto o provedor da nuvem fica encarregado de administrar toda a infraestrutura.

Por fim, o SaaS oferece aos usuários uma forma de acesso a software ou serviço, no formato pague "pelo que usou", que reside na nuvem e não no dispositivo do usuário. Uma característica marcante desse modelo é que a aplicação oferecida pelo fornecedor pode ser acessada por diversos consumidores de forma simultânea. Dessa maneira, o consumidor apenas utiliza essa aplicação sem preocupações relacionadas a infraestrutura ou os serviços necessários para execução da aplicação desejada, uma vez que tal responsabilidade é transferida para o provedor da nuvem.

2.1.2 Modelos de Implantação

Para disponibilizar o uso dos recursos computacionais compartilhados em nuvem é importante considerar o tipo que deve ser implantada. Esses tipos classificam-se em quatro modelos principais, baseados nos aspectos determinantes para o estabelecimento da forma como os recursos estão disponíveis e a maneira que a nuvem está localizada (MELL; GRANCE, 2011;

MALATHI, 2011; SOSINSKY, 2011; RIMAL; CHOI; LUMB, 2009):

- Nuvem Privada: esse tipo de nuvem é utilizado para atender um centro de dados de uma única organização. Neste modelo a computação em nuvem é emulada em redes privadas e oferece a capacidade para hospedar aplicações ou máquinas virtuais em um próprio conjunto da organização. Dessa forma, toda a infraestrutura é alocada apenas para a organização, sem haver o compartilhamento dos recursos com outros usuários.

- Nuvem Pública: modelo mais tradicional, no qual os recursos são provisionados dinamicamente através da Internet, a partir de um provedor *off-site* de terceiros que cobram pela utilização. Essa infraestrutura é mantida pelo provedor, que oferece a diferentes consumidores os mais variados serviços.

- Nuvem Comunitária: este é o modelo onde várias organizações têm exigências similares e buscam compartilhar uma mesma infraestrutura, no intuito de perceber alguns dos benefícios da computação na nuvem. Normalmente, é provisionada exclusivamente para um determinado grupo de usuários que compartilham os mesmos requisitos.

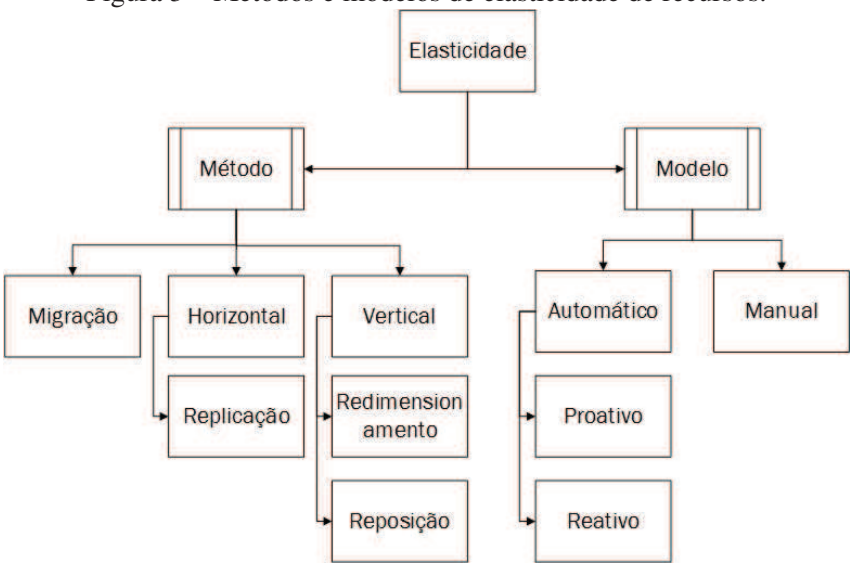
- Nuvem Híbrida: este modelo é composto por múltiplas infraestruturas distintas de nuvens (pública, privada ou comunitária), as quais mantêm suas identidades originais, mas são unidas como uma unidade única através de determinados protocolos que permitem a portabilidade das aplicações entre as diferentes infraestruturas.

2.1.3 Elasticidade

Por possibilitar a adição ou redução de recursos para atender às necessidades dos clientes, de forma flexível e sem interromper a execução dos processos, a elasticidade é uma característica fundamental da computação em nuvem (HU et al., 2016). MELL e GRANCE (2011) refere-se a elasticidade como capacidade de provisionar e liberar elasticamente, de maneira que se escale os recursos rapidamente em conformidade com a variação das demandas correntes. Para o consumidor, isso cria um cenário em que as capacidades disponíveis para provisionamento muitas vezes parecem ser ilimitadas e podem ser provisionadas a qualquer momento. Trata-se da principal característica que distingue a computação na nuvem de outras abordagens de sistemas distribuídos (JAMSHIDI; AHMAD; PAHL, 2014).

Importante frisar que, conquanto o termo elasticidade seja comumente usado como sinônimo de escalabilidade, com este não se confunde, pois, enquanto a escalabilidade é uma propriedade estática, que descreve a capacidade dos sistemas de atingir uma determinado limiar, a elasticidade é uma propriedade dinâmica, que permite ao sistema escalar sob demanda em um sistema operacional (AGRAWAL et al., 2011; DAS, 2011). Ainda, escalabilidade se conceitua como sendo a capacidade do sistema de ser ampliado para um tamanho que deveria acomodar um crescimento futuro quando recursos adicionais forem adicionados (AGRAWAL et al.,

Figura 3 – Métodos e modelos de elasticidade de recursos.



Fonte: Adaptado de Galante e Bona (2012)

2011).

Na visão do provedor da nuvem, a elasticidade permite que seus recursos sejam melhor aproveitados, evitando gastos com recursos não utilizados e liberando esses para diferentes usuários de forma simultânea. Já para os consumidores da nuvem, a elasticidade contribui para evitar a utilização inadequada dos recursos, reduzindo o altos custos relacionados com *over provisioning* (ARMBRUST et al., 2009). Nesse contexto, tem-se que a elasticidade é um atributo primordial oferecido pelo modelo de computação em nuvem (ROSA RIGHI, 2013).

Para além dos diferentes tipos de métodos que oferecem elasticidade, os quais são responsáveis por definir quais as operações a serem utilizados para provisionar ou liberar recursos, há também os modelos, que definem a maneira como as operações vão ser utilizadas (GALANTE; BONA, 2012; AL-DHURAIBI et al., 2017). Na Figura 3, pode-se visualizar esses diferentes métodos e modelos para a elasticidade na nuvem.

2.1.3.1 Modelos

Para a elasticidade, existem dois principais modelos, o manual e o automático. No modelo manual, o consumidor da nuvem se encarrega de monitorar todo seu ambiente e aplicações, sendo que, com base nas informações colhidas a partir do supervisionamento, deve realizar todas as ações de elasticidade que julgar necessárias (GALANTE; BONA, 2012). O provedor da nuvem, de seu turno, deve fornecer pelo menos uma interface com a qual o usuário vai interagir com o sistema. Como exemplos de sistemas nos quais os recursos são gerenciados manualmente, podemos citar os provedores públicos GoGrid (DATAPIPE, 2017) e Microsoft Azure (WINDOWS, 2017).

No automático, o controle e as ações passam a ser tomadas pelo sistema da nuvem, de acordo

com as regras e configurações definidas pelo usuário, ou especificadas no SLA. O sistema de controle utiliza serviços de monitoramento para coletar informações, tais como carga de CPU, memória e tráfego na rede, para decidir quando e como escala os recursos (GALANTE; BONA, 2012), e com base nestas informações são desencadeadas as ações de elasticidade.

Cabe mencionar que o modelo automático pode ser subclassificado de forma reativa e preditiva. As soluções reativas são baseadas em mecanismos de Regra-Condição-Ação. Uma regra é composta de um conjunto de condições que, quando satisfeito, desencadeiam algumas ações sobre a nuvem subjacente. Toda condição, de seu turno, considera um evento ou uma métrica do sistema que é comparada com um limite. A informação sobre valores e eventos de métricas é, por fim, fornecida pelo sistema de monitoramento de infraestrutura ou pelo aplicativo. Já as soluções preditivas utilizam técnicas heurísticas e matemáticas ou analíticas para antecipar o comportamento da carga do sistema e, com base nesses resultados, decidir quando e como escalar os recursos (GALANTE; BONA, 2012; AL-DHURAIBI et al., 2017).

2.1.3.2 Métodos

De acordo com Galante e Bona (2012), os métodos de execução de elasticidade podem ser classificados em três grupos, a saber: replicação, redimensionamento e migração. A replicação consiste em adicionar ou remover instâncias do ambiente virtual do usuário. Essas instâncias podem ser, por exemplo, máquinas virtuais ou contêineres. Atualmente trata-se do método mais utilizado para fornecer e suportar a elasticidade de alguns tipos de aplicações. Oferecem como recurso adicional, mecanismos de balanceamento de carga para dividir a carga entre as várias réplicas.

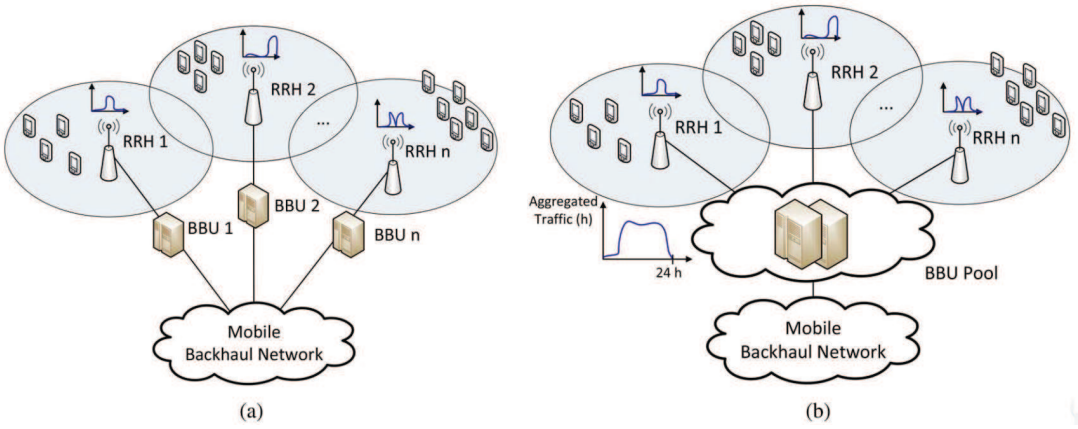
Nos métodos de redimensionamento, também conhecidos por elasticidade vertical, os recursos de processamento, como memória, CPU e armazenamento, podem ser adicionados ou removidos de uma instância virtual em execução. Redimensionar é o método padrão para aplicações, consistindo na adição de dados ou estruturas de controle, como exemplo, processos ou *threads*, para explorar novos recursos disponíveis na máquina virtual (GALANTE; BONA, 2012).

Por fim, o processo de transferência de uma máquina virtual que está sendo executada em um servidor físico para uma outra diferente, é denominado de migração. Ou seja, a elasticidade pode ser implementada pela migração de uma máquina virtual para uma máquina física que melhor se encaixa na carga da aplicação, ou por meio da consolidação e desconsolidação de um conjunto de máquinas em um servidor único (GALANTE; BONA, 2012).

2.2 *Cloud Radio Access Networks* - C-RAN

Como consequência do crescimento exponencial das comunicações móveis nos últimos anos, as *Radio Access Networks* (RANs) passaram a ter maiores taxas de dados trafegados

Figura 4 – Diferença entre arquiteturas: (a) arquitetura RAN; (b) arquitetura C-RAN.



Fonte: Adaptado de Checko et al. (2014)

em grandes áreas de cobertura localizadas em ambientes de alta mobilidade. Essa circunstância contribuiu para que o custo de construção, operação e atualização dessas RANs aumentasse substancialmente (CMRI, 2010).

Além disso, a arquitetura RAN tradicional passou a enfrentar uma série de outros desafios. Como exemplo, podemos citar o equipamento da *base station* (BS), que se localiza próximo da sua torre de antena e é conectado por um cabo coaxial. A BS pode servir apenas canais de radiofrequência em sua própria célula física, onde os recursos de hardware não podem ser compartilhados para atender às necessidades variáveis de transferência de comunicação em células diferentes. Não bastasse, uma BS altamente carregada não pode compartilhar energia de processamento com outras BS ociosas ou menos carregadas, resultando em baixa utilização dos recursos. Isso tudo sem contar com o fato de que as BSs são criadas em hardware proprietário, resultando em falta de flexibilidade para atualizar a rede de rádio (SIGWELE; PILLAI; HU, 2014).

Em face desse cenário, China Mobile, no ano de 2010, apresentou a arquitetura *Cloud Radio Access Network* (C-RAN) como uma maneira econômica de implantar células pequenas e como solução promissora para atender às demandas de alta capacidade. A C-RAN consolida BSs para um ponto centralizado na nuvem conhecido por *baseband unit pool* (BBU Pool). Dessa forma, ao invés de atuar com recursos limitados, a C-RAN se beneficia da elasticidade da computação em nuvem, permitindo o provisionamento dinâmico de recursos conforme a demanda corrente (CMRI, 2010). Na Figura 4 podem ser observadas as diferenças existentes entre a arquitetura de uma RAN e de uma C-RAN, já que nesta ocorre um desacoplamento da unidade de processamento dos RRH's para a nuvem, onde as tarefas passam a ser processadas por instâncias da nuvem.

Essa nova arquitetura é formada por três componentes principais: (i) o *remote radio heads* (RRH), que se localiza em um site remoto e possui a antena para comunicação entre os dispositivos dos usuários e a rede; (ii) a *Base Band Unit Pool* (BBU Pool), que é formado por processadores de alta velocidade e é responsável por levar a efeito o processamento do sinal,

além de realizar comunicação com a rede externa; e (iii) o *fronthaul*, que se trata de um canal de comunicação de alta largura de banda e baixa latência, que conecta os RRHs no BBU Pool. As funcionalidades do *Base Band Unit* (BBU) são implementadas em máquinas virtuais (VMs), que estão alojadas na nuvem. Essa característica centralizada, juntamente com a tecnologia de virtualização e RRHs de baixo custo, oferece um maior grau de liberdade para tomar decisões otimizadas e fez da C-RAN um candidato promissor a ser incorporado nas redes 5G (TRAN; HAJISAMI; POMPILI, 2016; HAJISAMI; POMPILI, 2017).

O fluxo de funcionamento da C-RAN ocorre da seguinte forma: um usuário se aproxima de uma antena RRH da rede com o seu dispositivo e então ocorre uma conexão. Ao enviar ou receber algum dado, a antena encaminha o sinal para o RRH que realiza um pré-processamento necessário para digitalização e compressão do mesmo. O RRH encaminha, então, para o BBU Pool, através do *fronthaul*, esse sinal já comprimido. Ao chegar no BBU Pool essa tarefa é processada e, se necessário, a comunicação com a rede externa é realizada. Por fim o resultado é retornado ao usuário.

Essa arquitetura traz uma série de benefícios, onde destaca-se: (i) maior eficiência energética, pois devido ao seu processamento centralizado e a possibilidade de compartilhamento entre diversas BSs na nuvem, onde ocorre um melhor aproveitamento do poder computacional dos recursos. Ainda, em momentos de baixa utilização, é possível desligar ou deixar em um modo de redução de energia; (ii) adaptação com tráfego não uniforme uma vez que a elasticidade da nuvem, possibilita provisionar ou liberar recursos, "*on-the-fly*", conforme as demandas; (iii) simplificação de novos RRHs, visto que a adição de novos RRHs se torna mais simples e barata, uma vez que o processamento dos recursos são provisionados virtualmente; e (iv) melhora no CAPEX e OPEX, devido aos RRHs se tornam mais simples e baratos pois não precisam mais lidar com o processamento dos sinais, diminuindo também o seu consumo energético.

Ao mesmo tempo, a C-RAN traz consigo alguns desafios, onde destacam-se: (i) a distribuição das tarefas provenientes dos RRHs para os BBU Pools; (ii) garantir que RRHs sejam processados por BBU Pools que se encontrem dentro da distância máxima recomendada de aproximadamente até 40km, devido a questões de latência, tempo de resposta e perda de sinal (CHANCLOU et al., 2013); (iii) o provisionamento dinâmico de recursos, que envolve o dimensionamento e gerenciamento dos recursos do pool, para tratar momentos de alta utilização, bem como momentos de baixa utilização; (iv) congestionamentos no *fronthaul* proveniente da grande quantidade de RRHs e dispositivos dos usuários, pode acabar gerando congestionamentos que levam ao aumento de latência na comunicação e tempo de resposta das requisições; e (v) gerenciamento da elasticidade dos recursos na nuvem para realizar processamento das tarefas com alto desempenho e baixo custo de infraestrutura.

2.3 Estratégias de balanceamento de carga

Um balanceador de carga é basicamente um dispositivo ou uma unidade de software que distribui a carga uniformemente pelos recursos do sistema. Dessa maneira, o tráfego recebido é distribuído no nível da rede, através de um algoritmo que é responsável por tomar decisões referentes ao encaminhamento desses dados (KAUR; KAUR, 2015). Um algoritmo de balanceamento de carga tem como principal objetivo impedir, sempre que possível, que alguns processadores fiquem sobrecarregados com um conjunto de tarefas, enquanto outros estão levemente carregados, levando a uma maior utilização de recursos, aumentando da taxa de transferência e redução do tempo de resposta (XU; LAU, 1997). Esses algoritmos são projetados para balancear a carga de aplicações definidas tanto com paralelismo de controle como paralelismo de dados.

2.3.1 Estratégia Estática

A estratégia de balanceamento de carga estática é considerada o tipo mais trivial, pois é caracterizada apenas por uma política de distribuição que divide igualmente o conjunto inicial de tarefas entre todos os processadores. Essa divisão ocorre apenas no início da execução, onde se assume que o número e comportamento de cada tarefa é conhecido quando o sistema ainda está sendo compilado. Não há, por essa razão, políticas que permitam a troca de tarefas entre os processadores para reduzir os efeitos dinâmicos de desequilíbrio de carga. Nessa abordagem, fica claro o conceito de mestre e escravo, uma vez que o desempenho dos nós é determinado no início da execução e as tarefas são atribuídas aos nós escravos em tempo de compilação, sem que seja possível realizar mudanças em tempo de execução (SEMCHEDINE; BOUALLOUCHE-MEDJKOUNE; AÏSSANI, 2011).

A vantagem dessa política é a baixa comunicação entre os processadores, na medida em que, em um sistema formado por N nós executando uma aplicação que não gera novas tarefas, serão necessários apenas N envios de tarefas. Além disso, é uma estratégia simples e de baixa complexidade de implementação. Em contra partida, como desvantagem, traz o fato de não ser possível realizar mudanças em tempo de execução, o que leva alguns processadores a ficarem ociosos enquanto outros estão sobrecarregados quando a presença de cargas externas à aplicação ocasiona desbalanceamento (ZHANG; KAMEDA; HUNG, 1997).

2.3.2 Estratégia Dinâmica

Ao contrário das políticas de balanceamento de carga estática, as dinâmicas visam equilibrar as tarefas com base em informações de uso dos servidores. Nessa modalidade a atribuição das tarefas é realizadas em tempo de execução (SEMCHEDINE; BOUALLOUCHE-MEDJKOUNE; AÏSSANI, 2011). Em algoritmos dinâmicos, assume-se que se tem pouco co-

nhecimento, a priori, sobre as necessidades de recursos das tarefas que compõem o sistema, portanto, as decisões de escalonamento são realizadas apenas quando o sistema está em execução, o que resulta em uma melhor tomada de decisão em comparação com o balanceamento de carga estático.

Isso resulta em uma melhor tomada de decisão em comparação com o balanceamento de carga estático, no entanto, é necessário um monitoramento continuo da carga atual em todos os nós, o que gera uma sobrecarga extra, na medida em que o monitoramento consome ciclos de CPU. Além disso, possui uma complexidade de implementação maior do que em relação à abordagem estática (ZHANG; KAMEDA; HUNG, 1997).

2.4 Análise de desempenho através da aplicação de séries temporais

Devido a adoção da computação em nuvem como alternativa para o processamento elástico de dados, diferentes estratégias de elasticidade são exploradas pelos provedores da nuvem e em iniciativas de pesquisa acadêmica. Destas estratégias, diversas utilizam técnicas de monitoramento que dependem de previsão de valores futuros e cálculos de suavização de valores. Análise de séries temporais oferecem diversos métodos para calcular a previsão da carga de processamento baseando-se em históricos de monitoramento. Estes métodos tornam-se relevantes em contextos de tomadas de decisões onde cada um possui diferentes parâmetros e características. Um dos modelos mais usados para a previsão de séries temporais é o modelo ARIMA (*Autoregressive Integrated Moving Average*) (HERBST et al., 2013).

Segundo Herbst et al. (2013), dentre os métodos para suavização de valores através do uso de análise de séries temporais, destacam-se: (i) *moving average* (MA), o qual depende de uma quantidade fixa de valores da janela de observação; e (ii) *simple exponential smooting* (SES), que utiliza o histórico de todos os valores coletados, onde o valor mais recente recebe um peso maior e os valores mais antigos vão recebendo pesos menores, ou seja, ocorre uma suavização exponencial nos valores para diminuir o impacto de ruído na lista de valores.

3 TRABALHOS RELACIONADOS

Esse capítulo apresenta os trabalhos relacionados com a área dessa pesquisa e suas contribuições para a mesma. Na primeira seção, são apresentados os critérios de seleção dos trabalhos e detalhes sobre a execução da pesquisa. Nas seções subsequentes, é apresentado um breve resumo de cada trabalho relevante e seus respectivos resultados obtidos. Por fim, é realizada uma análise comparativa entre os trabalhos e são apresentadas lacunas encontradas na área da pesquisa.

3.1 Escolha dos Trabalhos Relacionados e Metodologia de Pesquisa

Com objetivo de obter publicações que representam o estado da arte relacionado a alto desempenho em arquiteturas C-RAN, a pesquisa foi restringida a base de dados bem conceituadas pela comunidade científica. Através de seus motores de busca, as seguintes bibliotecas digitais foram pesquisadas: IEEE Xplore Digital Library[1], Springer[2], ScienceDirect[3] e ACM Digital Library[4].

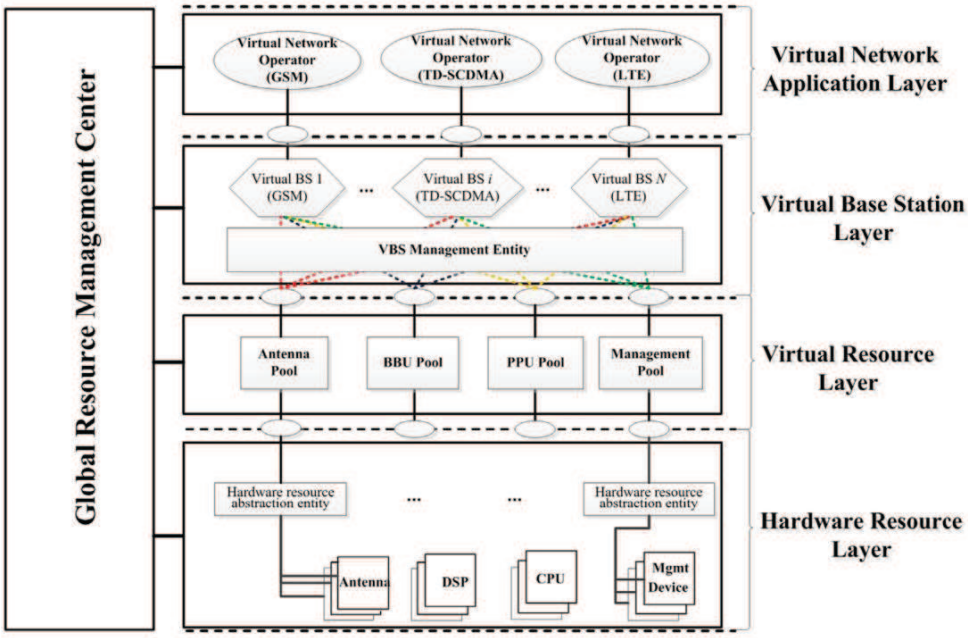
A pesquisa foi restrita a termos de busca simples que, apesar de retornar um grande número de publicações, resultam em uma gama mais abrangente de trabalhos que se enquadram na área de pesquisa. Além disso, foi utilizado os motores de busca avançados disponibilizados por essas bibliotecas digitais. Assim foi possível utilizar operadores booleanos nos termos de busca para obter melhores resultados. Os seguintes termos de busca foram utilizados: ("c-ran"OR "cloud-ran"OR "centralized-ran"OR "bbu"OR "base band unit") AND ("elasticity"OR "elastic"OR "scalability"OR "performance"). A busca foi realizada em setembro de 2017 e os resultados da pesquisa foram avaliados com o objetivo de encontrar trabalhos que explorassem alto desempenho no processamento de tarefas em uma arquitetura C-RAN.

3.2 Trabalhos Analisados

Nessa seção, são apresentados os trabalhos coletados através da aplicação da metodologia descrita anteriormente. Cada trabalho é apresentado individualmente em uma subseção. Tenta-se destacar a abordagem utilizada por cada autor, os resultados obtidos e sua contribuição.

[1] http://ieeexplore.ieee.org
[2] http://www.link.springer.com
[3] http://www.sciencedirect.com
[4] http://www.dl.acm.org

Figura 5 – Arquitetura do sistema Super BS.



Fonte: (QIAN et al., 2015).

3.2.1 A Super Base Station Based Centralized Network Architecture for 5G Mobile Communication Systems (QIAN et al., 2015)

Apresenta uma arquitetura denominada Super Base Station (Super BS), a qual é utilizada na C-RAN como uma solução para o uso eficiente de energia em redes móveis 5G. Na Figura 5, pode-se observar a arquitetura proposta, a qual é formada por três componentes principais: (i) o BBU Pool de alto desempenho; (ii) as unidades de processamento do pool; (iii) centro de gerenciamento dos recursos. A ideia dos autores é desacoplar as funções lógicas e entidade físicas das BS tradicionais para que diferentes tipos de recursos de sistemas possam ser compartilhados de forma horizontal entre as *virtual base stations* (VBS).

Através de técnicas que utilizam um esquema de alocação de hardware que considera a diversidade das cargas, diferentes recursos podem ser compartilhados horizontalmente entre VBS. Esses recursos de hardware podem ser alocados dinamicamente para as VBS de acordo com os seus perfis de carga de tráfego. Dessa forma, conforme a carga de tarefas corrente, ocorre um mapeamento em tempo de execução entre os BBUs e as VBS para desativar ou reativar BBUs, contribuindo para a economia de energia. Apesar de não ser apresentado detalhes referentes aos testes da arquitetura, como resultados constatou-se que através da utilização do Super BS, o consumo de energia no BBU pode ser diminuído e a utilização do sistema pode ser aumentada.

3.2.2 An Energy-Effective Network Deployment Scheme for 5G Cloud Radio Access Networks (LI et al., 2016)

Esse artigo introduz uma solução para selecionar um grupo de RRHs de forma que se adaptem às cargas dinâmicas de dados na C-RAN e aumentem a eficiência energética. Para tal, Li et al. (2016) propuseram um novo esquema de implantação de rede ciente da energia e das demandas de tráfego de dados que foi denominado EEND. O EEND foi projetado para reduzir o consumo de energia da rede por meio de uma combinação, semelhante ao problema da mochila (KELLERER; PFERSCHY; PISINGER, 2004), com base na demanda de tráfego corrente dos RRHs.

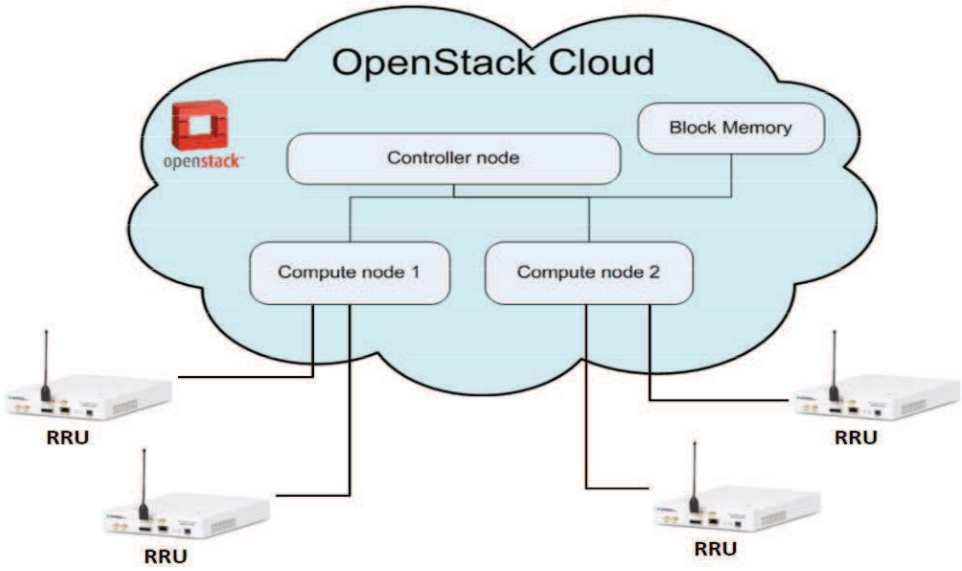
Além disso, uma estrutura C-RAN é proposta onde o sistema pode selecionar dinamicamente um subconjunto de RRHs de acordo com a demanda de tráfego corrente e as diferentes capacidades dos RRHs. Foram realizadas simulações para avaliar o desempenho do EEND sob diversos cenários de rede e os resultados foram comparados com uma solução de algoritmo genético heurístico amplamente utilizado na literatura. Como principal resultado, constatou-se que o EEND obteve mais de 50% de economia de energia em relação à abordagem heurística.

3.2.3 Analysis of virtual resource allocation for Cloud-RAN Based Systems (RAKOVIC et al., 2017)

Esse artigo apresenta uma plataforma de demonstração C-RAN capaz de realizar uma virtualização de diferentes tipos de tecnologias sem fio. A plataforma de demonstração apresentada utiliza soluções comerciais e de código aberto, como OpenStack e GNU Radio. O objetivo principal do trabalho é explorar a plataforma de demonstração desenvolvida para avaliar o comportamento e desempenho das tecnologias GSM e LTE no contexto de uma arquitetura C-RAN frente diferentes cenários. Para tal, foi proposta a arquitetura da Figura 6, na qual pode-se observar os principais componentes envolvidos e suas interconexões.

Observa-se que a implementação da plataforma é formada por três tipos diferentes de nós que são instalados em máquinas físicas diferentes. O controlador é responsável por gerenciar todas as funcionalidade e recursos da rede. Os nós computacionais são os BBU Pools que podem receber instâncias máquinas virtuais para processamento de tarefas. Por fim, o bloco de memória é uma região para comunicação entre os controladores e os nós computacionais. Com base nessa arquitetura, elaborou-se um protótipo no qual foram realizados testes de desempenho frente diferentes cenários. Os resultados comprovam a aplicabilidade da plataforma de demonstração C-RAN desenvolvida, além de fornecerem uma visão significativa do comportamento e desempenho das tecnologias sem fio analisadas em relação ao seu requisito de recursos da nuvem.

Figura 6 – Arquitetura com os componentes do OpenStack e suas interconexões.



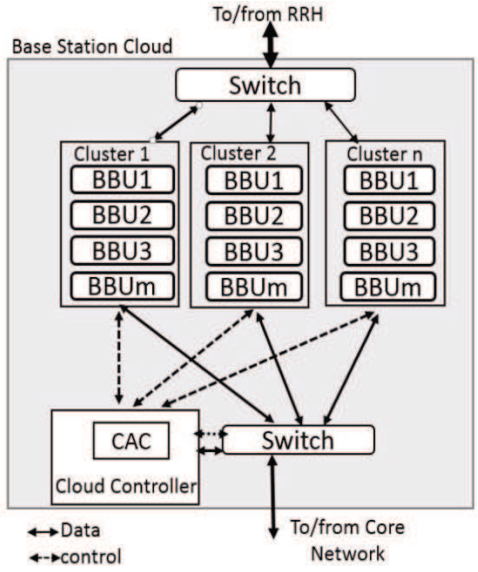
Fonte: Rakovic et al. (2017).

3.2.4 Call Admission Control in Cloud Radio Access Networks (SIGWELE; PILLAI; HU, 2014)

O artigo apresenta um algoritmo para controle de admissão de chamadas (CAC), com o objetivo de garantir qualidade no processamento e entrega de tarefas. Esse algoritmo coleta e utiliza informações de tráfego para verificar a existência de recursos suficientes para garantir a QoS da chamada requisitada. O algoritmo foi implantado junto à arquitetura proposta na Figura 7. Nessa arquitetura o mapeamento do BBU responsável pelo processamento de cada chamada recebida é realizado por um *switch* que encaminha a requisição para o BBU menos utilizado no momento. Quando todos os BBUs estão carregados e a QoS pode ser violado, ocorre o descarte de novas chamadas e são realizadas ações de elasticidade para aumentar a capacidade de processamento dos BBUs. O artigo apenas específica que o componente do controlador é responsável por controlar todos os BBUs presentes nos *clusters*, mas não apresenta detalhes de como realizar ações ou da forma de elasticidade empregada.

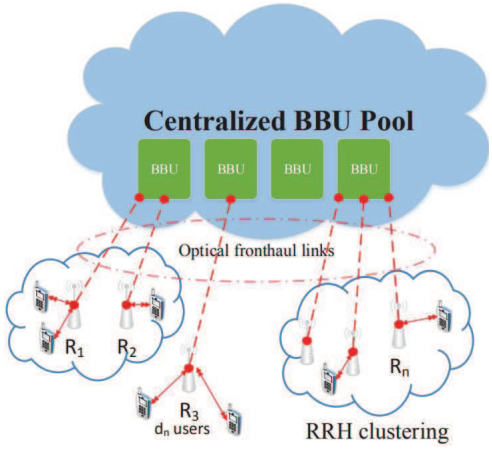
Para avaliar os algoritmos desenvolvidos, foram investigados três parâmetros de desempenho: a probabilidade de bloqueio de chamadas, o tempo médio de espera por chamada e a utilização do sistema. Esses parâmetros foram observados através da comparação do CAC com a RAN convencional. Dessa forma, pode-se concluir que o aumento da quantidade de BBUs na C-RAN acaba diminuindo o tempo médio de esperadas chamadas, pois as requisições são atendidas mais rapidamente. Também foi possível verificar que esse aumento do número de BBUs na nuvem leva a uma baixa utilização dos recursos.

Figura 7 – Arquitetura C-RAN adaptada para atuar com componente do CAC.



Fonte: Sigwele, Pillai e Hu (2014).

Figura 8 – Componentes propostos para o agrupamento na C-RAN.



Fonte: Taleb et al. (2017).

3.2.5 Centralized and Distributed RRH Clustering in Cloud Radio Access Networks (TALEB et al., 2017)

Taleb et al. (2017) apresentaram um modelo a ser utilizado em uma arquitetura C-RAN, com um algoritmo para mapear o processamento do sinal de múltiplos RRHs para cada unidade do BBU Pool, considerando métricas pertinentes na C-RAN. A ideia é otimizar a taxa de transferência de dados na rede, o consumo de energia e a frequência na entrega de sinais. Na Figura 8 é apresentado a disposição dos principais componentes envolvidos nesse processo de agrupamento.

Como premissas para o funcionamento do modelo, os provedores da C-RAN devem informar pesos para a taxa de transferência de dados, consumo de energia e à frequência de entrega.

Essas métricas e seus respectivos pesos são combinados para formar um score de carga, o qual é utilizado na tomada de decisão de agrupamento dos RRHs para cada BBU. Essa abordagem permitiu a investigação das compensações entre esses indicadores de desempenho, tornando a solução flexível e permitindo aos operadores a capacidade de aplicar diferentes estratégias. Foram projetados dois algoritmos para realizar esse agrupamento, um centralizado e um distribuído. Os resultados das simulações realizadas constataram que o algoritmo centralizado se adapta às condições de carga da rede e supera o método tradicional onde apenas um RRH é atribuído a cada BBU. No que pertine a abordagem do algoritmo distribuído, observou-se um desempenho muito próximo da solução ideal, com uma complexidade computacional significativamente menor quando comparado ao método tradicional.

3.2.6 CloudIQ: A Framework for Processing Base Stations in a Data Center (BHAUMIK et al., 2012)

Esse trabalho é o primeiro a realizar um estudo sistemático dos possíveis ganhos provenientes do processamento dos recursos centralizados na nuvem, propondo um *framework* para atingir uma troca precisa entre garantias de alto processamento e os custos do uso dos recursos na C-RAN. Bhaumik et al. (2012) apresentam uma arquitetura centralizada, na qual as BSs existentes são substituídas apenas por antenas e alguns outros componentes de RF ativos, enquanto o restante do processamento digital, incluindo a camada física, é realizado em uma localização central na nuvem. Além disso, o *framework* particiona as BSs para recursos homogêneos e realiza o a distribuição das cargas em tempo real para garantir QoS.

Visando benefícios que incluem redução no custo de operação da rede, devido a redução de visitas ao site, atualizações fáceis e melhora no desempenho da rede com técnicas comuns de processamento de sinal que abrangem diversas BSs, foi criado um protótipo para avaliação da arquitetura. Através de simulações que exploraram variações na carga de processamento, constatou-se que a arquitetura proposta pode resultar em economia de até 22% nos recursos de computação quando comparada com o modelo tradicional.

3.2.7 Dynamic Resource Scheduling in Cloud Radio Access Network with Mobile Cloud Computing (WANG et al., 2016)

Nesse artigo é elaborado um *framework* voltado para os provedores de serviço móvel, de maneira a otimizar a troca entre o gasto de energia e o desempenho no processamento de sinais através da programação de um conjunto de recursos de computação e de rede em arquitetura C-RAN. Assim, tem como o objetivo minimizar o consumo de energia dos provedores, visando maximizar os lucros e ao mesmo tempo garantir a QoS para os usuários finais.

Para tal tarefa, foi projetado um algoritmo capaz de otimizar a configuração dos *links* do *fronthaul*, do despachador de tarefas e dos servidores dos BBUs, para lidar com as requisições

de usuários móveis, uma vez que esses pedidos são imprevisíveis e variáveis conforme o tempo. Através de amplas simulações, foi demonstrado que o algoritmo proposto se aproxima de um tempo médio considerado próximo do ótimo para o MSP, mantendo a estabilidade do sistema e o baixo congestionamento na rede para garantir a QoS dos usuários móveis.

3.2.8 Efficient Algorithm for Baseband Unit Pool Planning in Cloud Radio Access Networks (XU; WANG, 2016)

Nesse artigo é investigado o problema de mapeamento de RRHs para BBU Pools distribuídos. Trata-se de um problema que abrange cenários práticos de redes de celulares em arquitetura C-RAN. De forma geral, a ideia é buscar minimizar o custo de implantação dos BBUs ao considerar: (i) sua capacidade de processamento; (ii) as demandas de tráfego de RRHs; e (iii) a sincronização de sinal entre RRHs e BBUs. Dessa maneira, foi proposto um algoritmo de busca onde a principal tarefa de otimização é selecionar um subconjunto dos sites candidatos para implantar o BBU Pool e assim satisfazer a demanda de tráfego de todos os RRHs com o menor custo possível. Para essa otimização, também foi levada em consideração a distância entre os BBU Pools e as antenas, devido às restrições de latência, e as conexões entre os RRHs e esses pools.

Para avaliação do algoritmo foram realizadas simulações em um cenário de 20KM x 20KM com RRH e BBU Pools uniformemente distribuídos. Definiu-se métricas para capacidade de processamento dos BBUs e de requisição de tarefas dos RRHs. Os resultados obtidos através dos experimentos demonstram que o algoritmo proposto pode reduzir significativamente o custo de planejamento da implantação do BBU Pool em arquiteturas C-RAN.

3.2.9 Energy-Efficient BBU Allocation for Green C-RAN (SAHU et al., 2017)

Nesse artigo é proposto um algoritmo para atribuição dos RRHs que serão processados por cada BBU visando aumentar a eficiência energética do BBU Pool. Este algoritmo segue o paradigma de Divisão e Conquista, o qual consiste em dividir o problema a ser resolvido em partes menores, encontrar soluções para cada uma das partes, e então combinar as soluções obtidas em uma solução global. No momento da atribuição é levado em consideração os requisitos de capacidade de processamento para os RRHs e o volume de troca de dados para comunicação entre eles.

Foram elaborados cenários para simulação envolvendo 100 BBUs com diferentes perfis de carga de tráfego para a rede e diferentes capacidades de processamento. Como resultados foi possível constatar que o algoritmo proposto atingiu reduções de até 40% nas comunicações entre os BBUs e um desempenho energético de até 20% em comparação com o método tradicional, que se refere à atribuição estática. Além disso, quando comparado com o tradicional, a nova proposta reduziu em até 30% a quantidade de *handovers*, que se referem a transição de

uma unidade móvel de uma célula para outra.

3.2.10 Elastic-Net: Boosting Energy Efficiency and Resource Utilization in 5G C-RANs (HAJISAMI; TRAN; POMPILI, 2017)

Hajisami, Tran e Pompili (2017) propuseram um framework para provisionamento de recursos de forma elástica em uma arquitetura C-RAN, denominado Elastic-Net, com o objetivo de se adaptar às flutuações das requisições que são não uniformes ao longo do dia e maximizar a eficiência energética na utilização dos recursos. O *framework* parte do princípio da existência de regiões que são divididas para processamento em *clusters*. Em cada um, ocorre uma adaptação da quantidade de RRHs ativos, da potência de transmissão e da quantidade de recursos das máquinas virtuais com base nas demandas correntes, de modo a minimizar o consumo de energia ao maximizar a utilização dos recursos ativos. Da mesma forma, para minimizar o consumo de energia na nuvem, foi otimizado e adaptado o tamanho das VMs, através da elasticidade vertical, ao mesmo tempo em que se assegura que o prazo de processamento das tarefas seja cumprido.

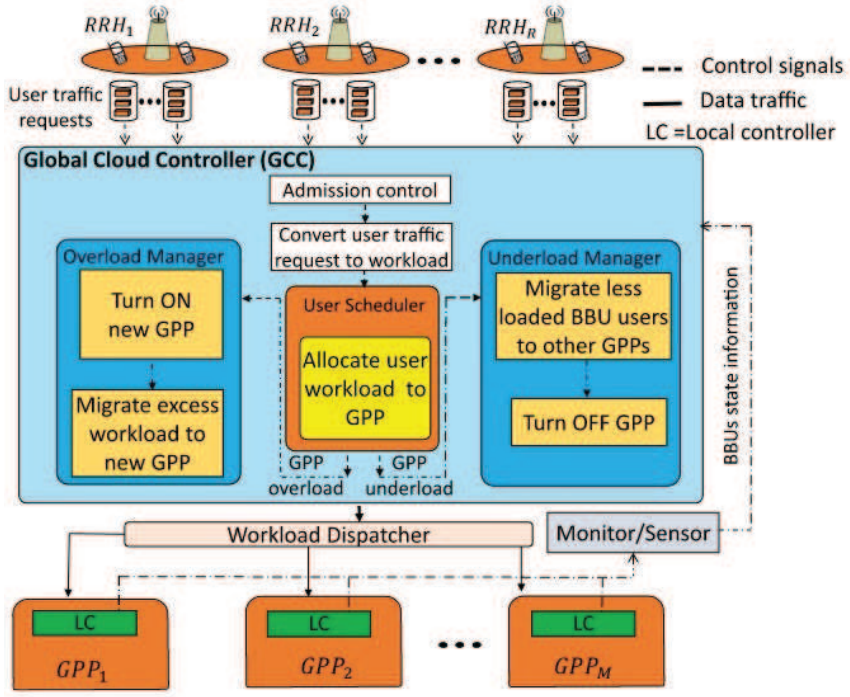
Como forma de avaliar o *framework*, elaborou-se um cenário de teste que considera uma rede de celular formada por RRHs e usuários distribuídos de forma independente. Nesse cenário, observou-se o desempenho do *framework* através da flutuação do tráfego em um típico dia operacional, onde foi contatou-se que o Elastic-Net adapta dinamicamente a densidade RRH, a potência de transmissão e o tamanho das VMs dos VBS para minimizar o consumo de energia, enquanto ao mesmo tempo, mantem as restrições de rede.

3.2.11 Energy-efficient Cloud Radio Access Networks by Cloud Based Workload Consolidation for 5G (SIGWELE et al., 2017)

Este artigo propõe um modelo de redução de energia para arquitetura C-RAN que se baseia na consolidação da carga de trabalho dos BBUs localizados na nuvem. A ideia do modelo é atuar com uma quantidade fixa de BBUs e conforme as demandas ativar ou desativar BBUs, sobrecarregados ou ociosos, reduzindo a baixa utilização dos recursos ativos e o consumo total de energia. Na Figura 9 pode-se observar o modelo proposto pelos autores, onde destaca-se os componentes: (i) controlador da nuvem, o qual é responsável por coletar o estado dos BBUs, através dos controladores locais, e tomar as decisões para encaminhar as requisições dos usuários; (ii) despachador de requisições, que é responsável por receber as decisões do controlador e encaminhar as requisições para o destino correto; e (iii) controlador local, que é responsável por monitorar os recursos do BBU Pool e expor as informações para o controlador da nuvem.

Para realizar a consolidação das cargas de forma eficiente, é apresentado um algoritmo para distribuir as cargas de trabalho recebidas entre os servidores que hospedam os BBUs de

Figura 9 – Modelo proposto para a consolidação da carga na arquitetura C-RAN.



Fonte: Sigwele et al. (2017).

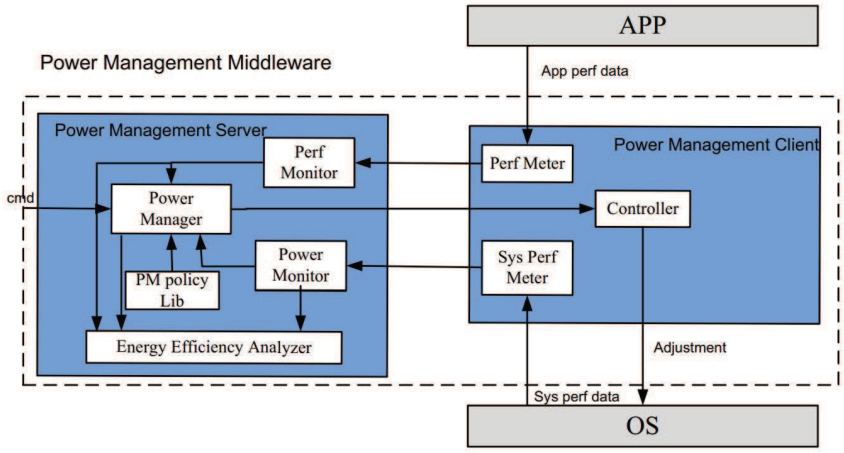
modo que cada servidor atue à plena utilização. Já os servidores de BBU ociosos acabam sendo desligados para economia energética. Em sua arquitetura ainda propõem um controle de admissão que deve descartar requisições quando todos os BBUs estiverem em plena utilização, evitando que não seja cumprida a QoS. Através de simulações foi constatado que o esquema de consolidação de carga de trabalho projetado obtém um desempenho de energia superior ao sistema LTE tradicional. Mais especificamente, a estrutura proposta pode economizar até 80% de energia em comparação com a abordagem tradicional.

3.2.12 Energy Efficiency using Cloud Management of LTE Networks Employing Fronthaul and Virtualized Baseband Processing Pool (AL-DULAIMI; AL-RUBAYE; NI, 2016)

Al-Dulaimi, Al-Rubaye e Ni (2016) propuseram um modelo para gerenciamento do provisionamento de recursos na nuvem capaz de adaptar a estrutura de links em uma rede LTE formada por *macro* e *femtocells* levando em consideração métricas de energia e desempenho. Para tal, o artigo se concentrou no desafio de adaptar as interfaces de rede de BBUs virtuais na nuvem em resposta à variação da carga de tráfego nos links do *fronthaul* e desenvolveu, assim, algoritmos que formam uma topologia através de um esquema de coloração de grafos e com base nessa topologia são criados os mapeamentos da rede considerando os recursos correntes disponíveis. Esse esquema de coloração é utilizado para rotular novos *clusters* de *femtocells* e realizar alocações eficientes dos BBUs, considerando a taxa de chegada das tarefas para processamento.

Após testes no modelo proposto, foi possível verificar três contribuições principais, são elas:

Figura 10 – Arquitetura do *middleware* de gerenciamento de energia.



Fonte: Gong et al. (2013).

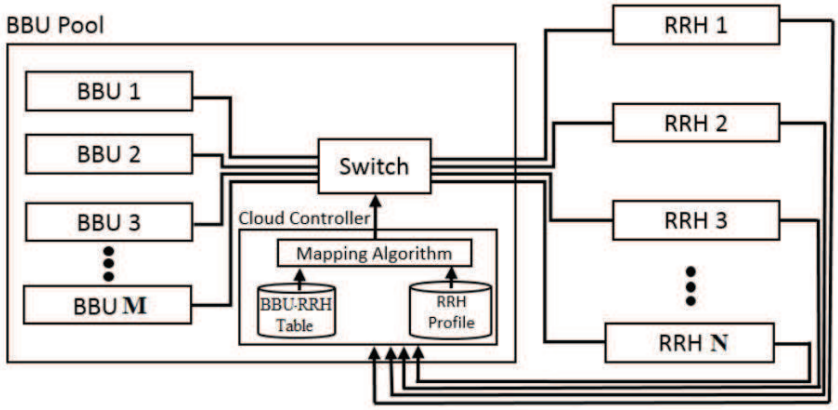
(i) um modelo de coloração de gráfico que desativa os *clusters* com pouco tráfego, permitindo economia de energia significativa; (ii) um estudo do conceito de configuração do pool BBU em resposta ao tráfego e os atrasos de tempo de processamento necessários para realizar adaptações da rede; (iii) e, por fim, o modelo C-RAN proposto foi ampliado para acoplar o Wi-Fi ao sistema em nuvem para obter ainda mais economia de energia por meio do conhecimento prévio da carga de tráfego transmitida por vários canais não licenciados.

3.2.13 GreenBase: An Energy-Efficient Middleware for Baseband Units in Radio Access Networks (GONG et al., 2013)

Gong et al. (2013) elaboraram um *middleware*, denominado GreeBase, para gerenciamento de energia no intuito de aumentar a eficiência energética dos BBUs. Como principais contribuições do artigo, destaca-se a criação do *middleware*, se baseando em um paradigma cliente-servidor, para fornecer ao usuário a capacidade de definir pesos de troca entre energia e desempenho através da escolha de políticas previamente cadastradas. Na Figura 10 é apresentada a arquitetura do *middleware*.

Para controlar a utilização de energia, utiliza uma ferramenta baseada em um modelo de energia linear com capacidade de auto calibração. Em posse desta informação de uso energético, observa a política previamente informada para avaliar a possibilidade de ativar, desativar ou incrementar o poder computacional de BBUs já existentes. Esse incremento em recursos já existentes é realizado com a exploração de uma abordagem de elasticidade vertical. Para avaliação do *middleware*, foi construído um *testbed* para executar uma aplicação de *streaming* de vídeo. Os resultados constataram que o consumo de energia pode ser reduzido em cerca de 13%, mantendo a QoS selecionado nas políticas cadastradas. Como trabalho futuro propõem explorar o *middleware* em um *cluster* para explorar a elasticidade dos recursos cientes da energia.

Figura 11 – Estrutura do modelo iTREE.



Fonte: (SIGWELE; PILLAI; HU, 2015).

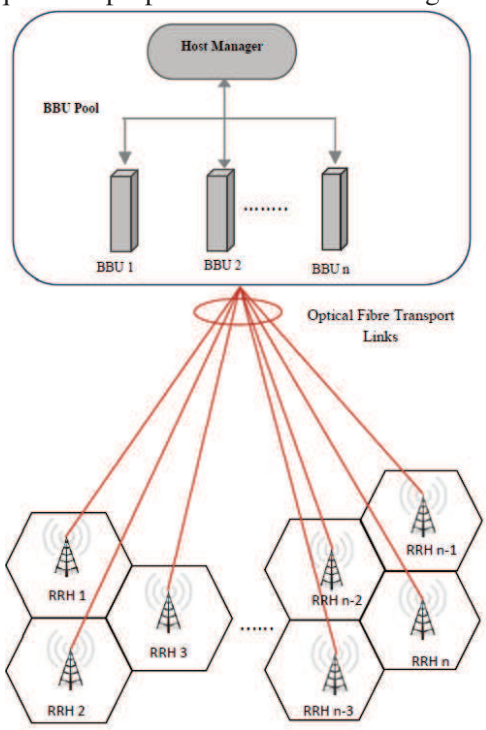
3.2.14 iTREE: Intelligent Traffic and Resource Elastic Energy Scheme for Cloud-RAN (SIGWELE; PILLAI; HU, 2015)

Sigwele, Pillai e Hu (2015) propuseram um modelo para tráfego inteligente e recursos elásticos para Cloud-RAN, denominado iTREE, com o objetivo de minimizar o consumo de energia em C-RAN através da redução do número de BBUs utilizados na nuvem. No iTREE, a quantidade de BBUs ativos é dinâmica conforme o volume de tráfego corrente. Dessa maneira, os recursos de energia podem ser utilizados de forma eficiente nos BBUs. Para tal, foram propostos algoritmos de aproximação heurística, traçando uma analogia com o problema do empacotamento (KELLERER; PFERSCHY; PISINGER, 2004), onde N itens (demandas de recursos dos RRHs), precisam ser alocados em M caixas (BBUs), de forma que a quantidade de caixas seja minimizada ao máximo. Na Figura 11 é apresentada a estrutura do modelo proposto.

Pode-se elencar como principal contribuição desse artigo a criação do algoritmo para redução da quantidade de BBUs na C-RAN, através de um mapeamento de M BBUs para N RRHs. Assim, o processamento de sinal dos RRHs é encaminhado dinamicamente para cada BBU do pool, sob demanda, de acordo com as necessidades de tráfego. Através dessa abordagem, as dependências dos RRHs com BBUs ociosos é removida e os mesmos podem ser desativados após o um novo mapeamento do RRH para outro BBU.

Os resultados da simulação mostram que o iTREE pode reduzir BBUs em até 97% durante os horários de baixa utilização e em até 66% nos horários alta utilização, com reduções da energia utilizada na RAN de 27% e 18%, respectivamente, em comparação com implementações convencionais. Como oportunidades futuras, vislumbra-se utilizar no algoritmo de gerenciamento dos BBUs para tomada de decisão dos recursos de computação e espectro.

Figura 12 – Arquitetura proposta com o módulo do gerenciador de *hosts*.



Fonte: (KHAN; ALHUMAIMA; AL-RAWESHIDY, 2016).

3.2.15 Quality of Service Aware Dynamic BBU-RRH Mapping in Cloud Radio Access Network (KHAN; ALHUMAIMA; AL-RAWESHIDY, 2016)

O artigo explora o roteamento dos dados em uma arquitetura C-RAN para permitir o balanceamento de carga na rede. Khan, Alhumaima e Al-Raweshidy (2016) propuseram uma arquitetura que se auto organiza, automaticamente, para equilibrar o tráfego dos dados na rede ao reduzir o número de chamadas bloqueadas e como consequência melhorando a qualidade de entrega do serviço. A arquitetura pode ser observada na Figura 12, onde todos os BBUs são conectados a um mesmo gerenciador de *hosts*. Esse gerenciador monitora a carga de cada BBU e é responsável por realizar a conexão entre BBU e RRH mais adequada, através do uso de um algoritmo genético. A métrica de carga em cada BBU para a tomada de decisões é a quantidade total de usuários ativos, simultaneamente conectados, em cada um dos BBUs.

Através de testes, constatou-se que o Algoritmo Genético converge para uma solução ótima, com uma configuração adequada de RRHs e BBUs, reduzindo as chamadas bloqueadas de 80 para 0. Ainda, os *links* da rede recebem o tráfego de dados de modo equilibrado, impactando na melhora da qualidade na entrega do serviço.

3.2.16 Reallocation Strategies for User Processing Tasks in Future Cloud-RAN Architectures (SCHOLZ; GROB-LIPSKI, 2016)

O artigo avaliou diferentes estratégias para reduzir a capacidade de processamento requerida em uma arquitetura C-RAN através da alocação eficiente de tarefas para serem processamento nos BBUs. No presente artigo, a atribuição de tarefas de computação em um conjunto de recursos de computação é baseada em tarefas com um grão fino, onde se realiza apenas uma tarefa de computação por usuário.

Foram comparadas diferentes estratégias para equilibrar a carga entre as unidades de processamento do BBU Pool e reduzir os recursos de forma a operar com a quantidade mínima de recursos necessários para o processamento. Então foi elaborado um algoritmo para atribuições dinâmicas das tarefas através de uma abordagem heurística com capacidade de evitar situações de sobrecarga de curto prazo através de um controle de admissão de tarefas.

Os resultados das simulações efetuadas constatam que a combinação de distribuição de tarefas e o mecanismo de controle de admissão, que previne sobrecargas, permitiu avaliar a relação entre recursos de computação e Quality of Experience (QoE), que se refere a percepção do usuário sobre a qualidade de um serviço (CHEN; CHATZIMISIOS; TASOS DAGIUKLAS, 2015). No melhor dos casos, foi possível economizar até 40% dos recursos de processamento sem qualquer degradação da QoE.

3.2.17 Service Scheduling Scheme Based Load Balancing for 5G/HetNets Cloud RAN (CHABBOUH et al., 2017)

Chabbouh et al. (2017) propuseram um novo modelo para arquitetura C-RAN, conhecido por Cloud-RRH, onde é proposta uma nuvem na borda da rede. Os autores desenvolveram um algoritmo dinâmico e centralizado de agendamento de tarefas no Cloud-RRH que considera como parâmetros de análise para tomada de decisões o tempo de execução e a utilização de recursos das máquinas virtuais. O algoritmo seleciona de forma dinâmica as máquinas virtuais apropriadas para execução de cada tarefa, maximizando a utilização dos recursos, melhorando o desempenho da aplicação e aumentando a satisfação do usuário. Caso todas as máquinas estejam sobrecarregadas, uma nova é criada de forma horizontal para efetuar processamento. O artigo não apresenta detalhes sobre como é realizado o processo de elasticidade horizontal para criação e gerenciamento dos recursos.

Os resultados da simulação demonstraram que através do esquema proposto é possível reduzir o tempo total de processamento, em comparação com os algoritmos ACO e Round Robin, bem como melhorar o balanceamento de carga entre elas. Como trabalhos futuros, sugere-se um estudo mais profundo nas maneiras de realizar o gerenciamento dos recursos computacionais considerando a interferência e mobilidade dos usuários entre diferentes antenas.

Tabela 1 – Comparação entre as principais características dos trabalhos relacionados. ✓: Possui; ×: não possui; NE: não especificado.

Publicação	Balanceador de Carga	Arquitetura	Gerenciamento de Recursos	Método de Elasticidade	Modelo de Elasticidade	Parâmetros de Ação
(QIAN et al., 2015)	Dinâmico	Centralizada	×	NE	NE	Uso dos BBUs
(LI et al., 2016)	Dinâmico	Centralizada	×	×	×	Uso da rede
(RAKOVIC et al., 2017)	Estático	Centralizada	✓	Horizontal	Manual	NE
(SIGWELE; PILLAI; HU, 2014)	Dinâmico	Centralizada	✓	NE	NE	Uso dos BBUs
(TALEB et al., 2017)	Dinâmico	Distribuída	×	×	×	Uso da rede energia e *handover*,
(BHAUMIK et al., 2012)	Dinâmico	Centralizada	×	×	×	Uso dos BBUs
(WANG et al., 2016)	Dinâmico	Centralizada	✓	NE	NE	Quantidade de requisições
(XU; WANG, 2016)	Dinâmico	Distribuída	×	×	×	Uso dos BBUs e da rede
(SAHU et al., 2017)	Dinâmico	Centralizada	×	×	×	Uso da rede
(HAJISAMI; TRAN; POMPILI, 2017)	Dinâmico	Centralizada	✓	Horizontal	Automático e reativo	Quantidade de requisições
(SIGWELE et al., 2017)	Dinâmico	Distribuída	✓	×	×	Uso dos BBUs
(AL-DULAIMI; AL-RUBAYE; NI, 2016)	Dinâmico	Centralizada	✓	NE	NE	Uso dos BBUs
(GONG et al., 2013)	Dinâmico	Centralizada	✓	Vertical	Automático e reativo	Energia
(SIGWELE; PILLAI; HU, 2015)	Dinâmico	Centralizada	✓	NE	NE	Uso dos BBUs
(KHAN; ALHUMAIMA; AL-RAWESHIDY, 2016)	Dinâmico	Centralizada	×	×	×	Quantidade de usuários
(SCHOLZ; GROB-LIPSKI, 2016)	Dinâmico	Centralizada	×	×	×	Uso das VMs
(CHABBOUH et al., 2017)	Dinâmico	Centralizada	✓	Horizontal	Automático e reativo	Uso das VMs e da rede

Fonte: Elaborado pela autor.

3.3 Análise dos Trabalhos Relacionados

Com o objetivo apresentar uma síntese dos trabalhos estudados nas seções anteriores, elaborou-se a Tabela 1, que, compilando os principais dados sobre os modelos analisados, elencou seis características consideradas fundamentais para a análise, a saber:

- Balanceamento de Carga: política utilizada para realizar o balanceamento de carga de tarefas provenientes das antenas;
- Arquitetura: forma pela qual os componentes interagem e são mapeados na rede;
- Gerenciamento de Recursos: explora o gerenciamento de recursos em uma nuvem computacional;
- Método de Elasticidade: método de elasticidade fornecido;
- Modelo de Elasticidade: estratégia de elasticidade utilizada com relação ao modelo de elasticidade definido;
- Parâmetros de Ação: métricas consideradas para a realização de ações;

3.3.1 Oportunidades de pesquisa

Atualmente, diversas frentes de pesquisas buscam formas de atingir alto desempenho de processamento em arquitetura C-RAN. Com base na Tabela 1 e na análise realizada a partir

dos trabalhos relacionados, observa-se que a maioria dos estudos se concentra na criação e na combinação de algoritmos e métodos para mapear dinamicamente os RRHs nos BBUs. Uma importante tarefa, dado que as cargas dos RRHs variam durante o dia e, dessa forma, é possível ativar ou desativar os BBUs poucos utilizados, realizando um novo mapeamento dos RRHs para os BBUs ainda ativos. Algumas frentes focam no balanceamento da carga tomando por base diferentes métricas, para realizar o escalonamento das tarefas de forma estática ou dinâmica. Por fim, alguns trabalhos abordam a elasticidade como um meio para tornar a infraestrutura da C-RAN mais flexível e adaptativa às mudanças de carga do dia. Nesse contexto, identificam-se lacunas relacionadas com a pesquisa, as quais são pouco exploradas ou não abordadas pelos autores nas respetivas áreas de interesse. Esses aspectos foram agrupados e descritos da seguinte forma:

- A carga de tarefas recebida é processada em um único BBU Pool centralizado, criando um gargalo para uma alta quantidade de mensagens recebidas através da rede (QIAN et al., 2015; LI et al., 2016; RAKOVIC et al., 2017; SIGWELE; PILLAI; HU, 2014; TALEB et al., 2017; BHAUMIK et al., 2012; WANG et al., 2016; SAHU et al., 2017; HAJISAMI; TRAN; POMPILI, 2017; AL-DULAIMI; AL-RUBAYE; NI, 2016; GONG et al., 2013; SIGWELE; PILLAI; HU, 2015; KHAN; ALHUMAIMA; AL-RAWESHIDY, 2016; SCHOLZ; GROB-LIPSKI, 2016; CHABBOUH et al., 2017);

- A elasticidade é explorada apenas em nível de máquinas virtuais, sem uma abordagem para prover elasticidade multinível, onde ações elásticas são realizadas para BBU Polls (máquinas físicas) e para os BBUs (máquinas virtuais) de forma individual de forma automática e transparente para os provedores, considerando as principais métricas de recursos computacionais e de rede (QIAN et al., 2015; LI et al., 2016; RAKOVIC et al., 2017; SIGWELE; PILLAI; HU, 2014; TALEB et al., 2017; BHAUMIK et al., 2012; WANG et al., 2016; SAHU et al., 2017; HAJISAMI; TRAN; POMPILI, 2017; AL-DULAIMI; AL-RUBAYE; NI, 2016; GONG et al., 2013; SIGWELE; PILLAI; HU, 2015; KHAN; ALHUMAIMA; AL-RAWESHIDY, 2016; SCHOLZ; GROB-LIPSKI, 2016; CHABBOUH et al., 2017; XU; WANG, 2016; SIGWELE et al., 2017);

- O grão de elasticidade não é adaptativo para aumentar a reatividade do sistema de forma que o mesmo responda mais rapidamente às variações de carga (QIAN et al., 2015; LI et al., 2016; RAKOVIC et al., 2017; SIGWELE; PILLAI; HU, 2014; TALEB et al., 2017; BHAUMIK et al., 2012; WANG et al., 2016; SAHU et al., 2017; HAJISAMI; TRAN; POMPILI, 2017; AL-DULAIMI; AL-RUBAYE; NI, 2016; GONG et al., 2013; SIGWELE; PILLAI; HU, 2015; KHAN; ALHUMAIMA; AL-RAWESHIDY, 2016; SCHOLZ; GROB-LIPSKI, 2016; CHABBOUH et al., 2017; XU; WANG, 2016; SIGWELE et al., 2017);

De posse das informações relatadas, surgiu a oportunidade de pesquisa para preenchimento das lacunas identificadas, qual seja, a criação de um modelo de elasticidade multinível não

bloqueante para C-RAN, que oferece elasticidade tanto no nível de BBU Pool (máquina física), devido ao alto volume de tráfego e a distância máxima indicada entre os RRHs e o pool (CHANCLOU et al., 2013), quanto no nível de BBU (máquina virtual), em razão do processamento de CPU e memória necessário nas requisições. Essas operações devem ocorrer de forma não bloqueante para não penalizar o processamento das tarefas durante a reorganização de recursos.

Vislumbra-se, ainda, a criação de um mecanismo de elasticidade com um conceito de grão adaptativo com capacidade de provisionar e mapear os recursos sob demanda e em tempo de execução, processando as tarefas com maior reatividade, agilidade e o menor custo de infraestrutura possível. Esse mecanismo visa realizar baixa quantidade de ações elásticas com um tamanho de grão mais adequado para as necessidades correntes de recursos, aumentando o desempenho e a reatividade do sistema ao responder mais rapidamente às variações de carga.

4 MODELO ELASTIC-RAN

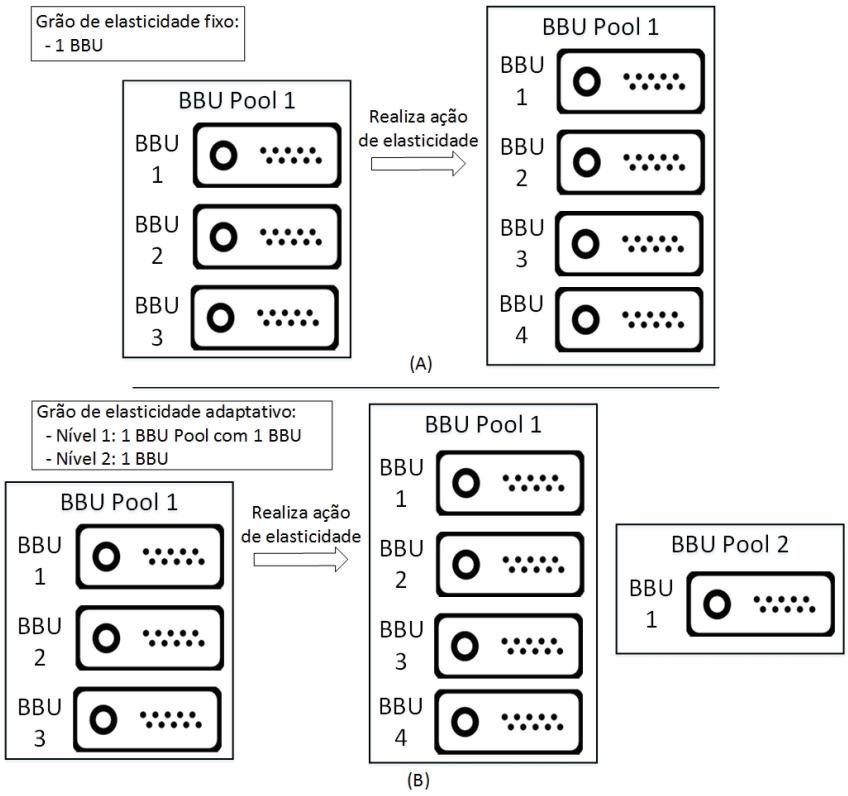
Partindo das lacunas identificadas nos trabalhos relacionados e das diferentes técnicas apresentadas para solucionar a questão de pesquisa, este capítulo, dividido em seis seções, tem por objetivo descrever o modelo Elastic- RAN. Na primeira seção, busca-se explicitar as decisões de projeto para o desenvolvimento do modelo. Em seguida, na Seção 4.2, a arquitetura do modelo Elastic-RAN e o seu fluxo de execuções são explicados. Já nas Seções 4.3 e 4.4, aborda-se em detalhes os principais componentes do modelo. Por fim, na Seção 4.5, apresenta-se o funcionamento do mecanismo de elasticidade multinível, junto dos métodos e algoritmos utilizados para o cálculo de cargas e tomadas de decisões elásticas.

4.1 Decisões de Projeto

O modelo Elastic-RAN propõem um conceito de elasticidade multinível não bloqueante, com orquestração automática de recursos através da coordenação de BBUs (máquinas virtuais) presentes em cada um dos BBUs Pools (máquinas físicas) disponíveis, junto a um mecanismo de grão elástico adaptativo, que considera o uso corrente dos recursos para provisionar e mapear diferentes quantidades de recursos em tempo de execução para cada ação elástica. A elasticidade multinível não bloqueante permite atuar tanto no nível de BBU Pool, devido ao alto volume de tráfego e a distância máxima sugerida entre os RRHs e os pools (CHANCLOU et al., 2013), quanto no nível de BBU, tendo em vista o alto processamento de CPU e memória necessária para as requisições. Como um dos desafios é não causar nenhum impacto nos processamentos correntes durante a reorganização de recursos, o processamento de requisições nos BBUs ocorre em paralelo e separado da reconfiguração de recursos, assim os recursos apenas são mapeados para iniciar processamento quando estiverem inicializados e preparados para tal. O mecanismo de elasticidade com grão elástico adaptativo, de seu turno, permite provisionar e mapear os recursos sob demanda e em tempo de execução. Com a utilização dessa técnica, demanda-se uma menor quantidade de ações elásticas com um tamanho de grão mais adequado para as necessidades correntes de recursos, aumentando o desempenho e a reatividade do sistema, de modo a responder mais rapidamente as variações de carga.

O Elastic-RAN utiliza elasticidade horizontal e reativa de forma automática para os provedores do serviço da C-RAN, sem a necessidade de intervenção humana para definição de regras complexas ou realização de ações manuais. Em vista disso, o modelo tem autonomia para acrescentar ou remover recursos considerando o uso corrente de CPU, de memória e de rede, criando um ambiente de nuvem dinâmico para o processamento das tarefas. Cabe ressaltar que a elasticidade horizontal foi adotada porque a elasticidade vertical limita os recursos disponíveis em uma única máquina física, sem contar o fato de que na maioria dos sistemas operacionais comuns não é permitido que sejam adicionados recursos em tempo de execução (LORIDO-BOTRAN; MIGUEL-ALONSO; LOZANO, 2014; DUTTA et al., 2012).

Figura 13 – Mecanismo de elasticidade. (A) abordagem tradicional em arquitetura C-RAN, onde a elasticidade é aplicada apenas no nível dos BBUs (máquinas virtuais) de um mesmo BBU Pool com o grão de elasticidade fixo (B) elasticidade multinível do Elastic-RAN, onde a elasticidade não é apenas aplicada no nível dos BBUs (máquinas virtuais), mas também no nível dos BBU Pools (máquinas físicas), junto a um grão de elasticidade adaptativo.



Fonte: (SIGWELE; PILLAI; HU, 2015).

A elasticidade automática e reativa é baseada em *thresholds*, tendo como ponto de partida regras previamente definidas, as quais devem ser analisadas, uma vez que envolvem uma série de decisões sobre configuração de limites de processamento, de tempo, de tipos de recursos e de frequência de monitoramento. Segundo Rosa Righi et al. (2016a), a elasticidade dispara alocações de recursos que apresentam impacto direto no custo de utilização da nuvem, de modo que a relação custo benefício deve ser cuidadosamente avaliada, bem como a ocorrência de falso-positivo e falso-negativo na tomada de decisões.

Como se pode observar na Figura 13, na abordagem tradicional (a), a elasticidade é aplicada apenas no nível de BBUs de um mesmo BBU Pool, utilizando como grão de elasticidade um valor estático, o qual é aplicado em todas as operações. Já no Elastic-RAN (b), a elasticidade acontece em dois níveis distintos, não sendo apenas realizada para o nível dos BBUs de forma individual, mas também no nível de BBU Pools.

Para o desenvolvimento do modelo, foram tomadas as seguintes premissas e decisões de projeto:

- O mecanismo de orquestração de máquinas físicas para os BBU Pools e máquinas virtuais para os BBUs deve ser capaz de coletar dados do uso corrente dos recursos computacio-

nais (CPU, memória e rede);

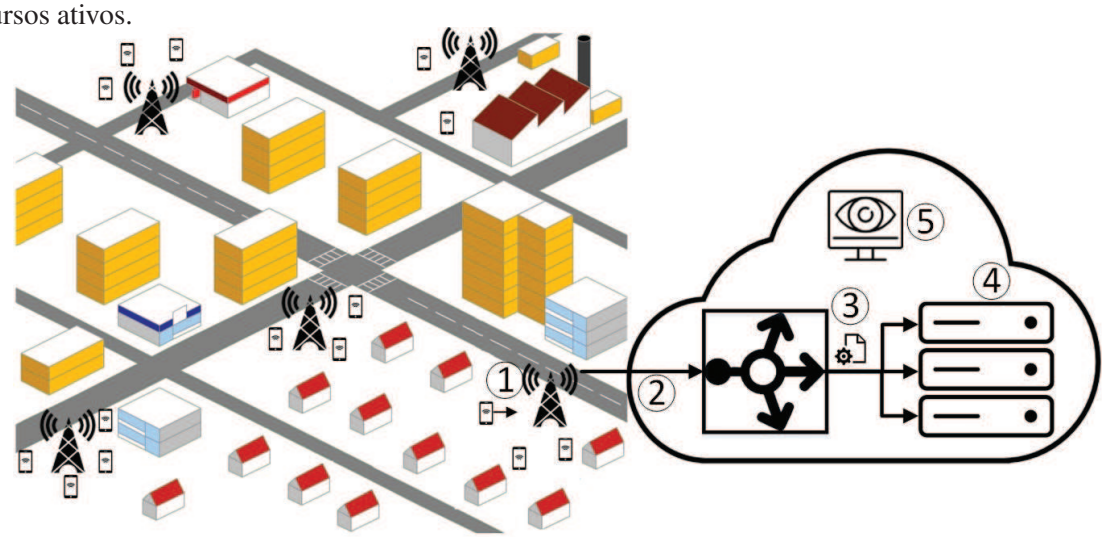
- Estratégia de orquestração dos recursos elásticos em dois níveis, para BBU Pool (máquinas físicas) e BBU (máquinas virtuais);
- A elasticidade adotada será de forma horizontal, automática e reativa utilizando a alocação, replicação e consolidação de máquinas virtuais;
- As operações de elasticidade não devem onerar o processamento das requisições, dessa forma deve ser provido um mecanismo de elasticidade não bloqueante;
- O grão de elasticidade a ser utilizado nas operações deve ser adaptativo, ou seja, em cada iteração esse grão deve ser dimensionado considerando a utilização corrente dos recursos disponíveis, para aumentar ou reduzir os recursos computacionais de forma mais adequado para as necessidades correntes de recursos;
- Capacidade de análise de picos e quedas repentinas na carga de trabalho a fim de evitar operações desnecessárias para evitar o efeito conhecido por *thrashing* (BERSANI et al., 2014);
- Algoritmo para realizar a distribuição das tarefas de forma que os BBU Pools e os BBUs fiquem balanceados e não sejam sobrecarregadas;
- A arquitetura em questão pode ser implementada em qualquer *middleware* de nuvem, deixando-o livre de restrições em relação à plataforma utilizada;

4.2 Arquitetura

O Elastic-RAN é um modelo desenvolvido para arquitetura C-RAN com o objetivo de possibilitar a exploração dos benefícios da elasticidade em nuvem computacional sem a necessidade de intervenção manual dos provedores da C-RAN. Para prover a orquestração da elasticidade, o modelo atua com operações de alocação e consolidação de instâncias de máquinas virtuais e máquinas físicas.

Na Figura 14, pode-se observar, em alto nível, o fluxo do processamento de uma requisição. Cada requisição solicitada por um usuário é capturada por uma antena próxima (1) que encaminha essas tarefas para o Orquestrador do Pool responsável por processar as tarefas da região (2). Essa requisição chega para uma fila de tarefas do tipo FIFO, que se encontra no Orquestrador do Pool. Essa tarefa é então repassada pelo orquestrador para um de seus BBUs Pools disponíveis (3) para, então, ser realizado o processamento por uma das unidades (4). Em paralelo, o Orquestrador de Elasticidade do Elastic-RAN monitora de forma periódica diferentes métricas em cada BBU Pools e suas unidades de processamento (5). Com base nos dados coletados e com suas regras estabelecidas, esse orquestrador realiza ações elásticas para a criação de novos

Figura 14 – Fluxo em alto nível do processamento de uma requisição no modelo Elastic-RAN, destacando-se 5 pontos: (1) o usuário realiza uma requisição que é capturada por uma antena; (2) essa antena repassa essa requisição para o Orquestrador do Pool responsável pela região; (3) posteriormente, o orquestrador distribuí para um de seus BBU Pools (4) realizarem o processamento; (6) em paralelo, ocorre o monitoramento periódico das diferentes métricas de uso dos recursos e o gerenciamento dos recursos ativos.



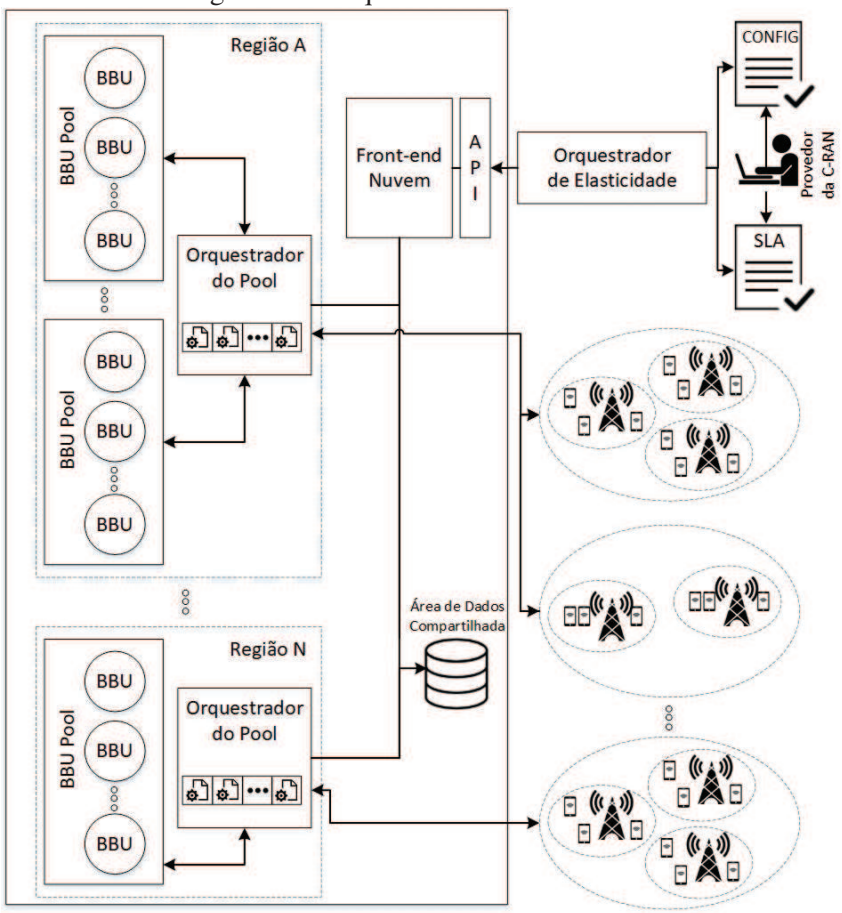
Fonte: Elaborado pelo autor.

BBU Pools através de novas máquinas físicas ou novos BBUs dentro de Pools já existentes com novas máquinas virtuais.

Na Figura 15, por sua vez, é apresentada a arquitetura do modelo Elastic-RAN, com a presença de seus principais componentes em um ambiente de nuvem composto por máquinas físicas homogêneas. O Orquestrador de Elasticidade não possui dependência de localização para sua execução, de modo que ele pode ser executado tanto dentro do ambiente de nuvem como fora, em uma máquina totalmente independente do ambiente. Isso é possível através do uso da API fornecida pela plataforma de infraestrutura da nuvem utilizada, a qual necessita apenas uma conexão de rede para iteração. O modelo em questão prevê uma área de dados compartilhada para comunicação entre todos os componentes envolvidos. Essa área de dados é um componente importante, pois viabiliza a implementação de políticas de comunicação entre os componentes e processos do ambiente. Já o Orquestrador do Pool possui a importante função de controlar e distribuir as tarefas para os seus pools processarem, ou seja, ele recebe as tarefas provenientes das antenas e as encaminha para processamento nos seus BBUs disponíveis.

O ambiente do Elastic-RAN é formado por m BBU Pools (máquinas físicas) homogêneos. Em cada uma dessas máquinas físicas, existem n BBUs (máquinas virtuais), sendo que n representa a quantidade de núcleos de processamento dentro da máquina física do pool. Como o processamento das tarefas nos BBUs deve ser realizado de forma intensiva, elas são distribuídas uma por vez. Essa abordagem é baseada no trabalho de Lee et al. (2011), no qual o autor relata que a melhor abordagem para desempenho e eficiência em aplicações que exigem alto poder de processamento é a utilização de uma máquina virtual em cada núcleo do processador,

Figura 15 – Arquitetura do Elastic-RAN.



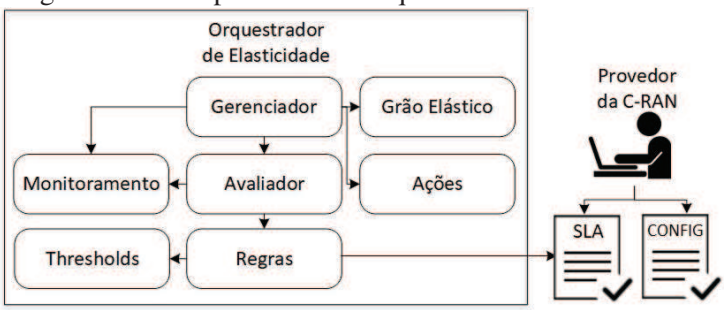
Fonte: Elaborado pelo autor.

executando apenas um processo por vez em cada máquina. Considerando esse estudo, no modelo proposto nesta dissertação, cada máquina virtual é instanciada com apenas um núcleo de processador e o Orquestrador do Pool apenas encaminha uma tarefa por vez para os seus BBUs processarem. Nesse contexto, Elastic-RAN sempre terá ativo ao menos um BBU Pool com um BBU para processamento.

4.3 Orquestrador de Elasticidade

A Figura 16 ilustra os principais componentes do Orquestrador de Elasticidade, que tem a responsabilidade de monitorar o ambiente e gerenciar as ações elásticas. O componente do monitoramento é responsável por realizar a coleta de estatística dos recursos do ambiente e realizar a análise dessas informações. Os dados são coletados através de uma API exposta pelo provedor de infraestrutura da nuvem, no qual são extraídas informações de uso de CPU, de memória e de rede dos recursos ativos. Com base nesses dados, o componente do avaliador calcula um score para cada uma dessas métricas, o qual define as cargas finais no ciclo corrente de monitoramento. Com isso, o score calculado passa a ser comparado com os *thresholds* supe-

Figura 16 – Componentes do Orquestrador de Elasticidade.



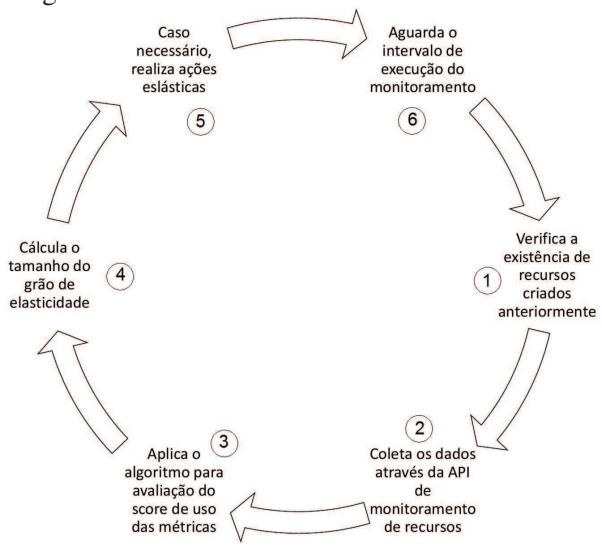
Fonte: Elaborado pelo autor.

riores e inferiores do componente de Regras no intuito de identificar situações que possam gerar ações elásticas. Além desses *thresholds*, o orquestrador ainda leva em consideração um SLA e um arquivo de configuração que deve ser informado pelo provedor do serviço para sinalizar ao módulo de Ações a necessidade de realização de ações elásticas. Esses arquivos recebem as informações de *thresholds* manuais, bem como a quantidade mínima e máxima de máquinas virtuais para a execução. Se o SLA não é informado, essas restrições são definidas com base na quantidade de máquinas virtuais existentes no ponto inicial de execução da aplicação. Por fim, o componente de Ações recebe informações do Gerenciador para realizar ações elásticas e o tamanho do grão a ser utilizado em suas ações é dimensionado dentro do componente de Grão Elástico. Mais detalhes sobre as técnicas de dimensionamento do grão serão apresentados na seção 4.5.1.

Assim como proposto por Imai, Chestna e Varela (2012); Chiu e Agrawal (2010); Rosa Righi et al. (2016a), o monitoramento dos recursos é realizado de forma periódica e é definido manualmente no momento de inicialização do orquestrador em seu arquivo de configuração. Para facilitar a compreensão, como se pode observar na Figura 17, a cada ciclo de monitoramento o orquestrador verifica se existem recursos inicializados anteriormente que podem estar disponíveis para a utilização (1). Isso ocorre pois novos recursos que são instanciados no ambiente só estarão disponíveis para realizar processamento após um determinado tempo de inicialização. Superada essa etapa, coletam-se as estatísticas de uso corrente dos recursos ativos (2), as quais são aplicadas no cálculo do score de uso de CPU, de memória e de rede (3). Em seguida é calculado o tamanho do grão a ser utilizado nas operações elásticas(4) e, caso necessário, realizam-se ações de elasticidade para aumentar ou reduzir os recursos (5). Por fim, o orquestrador aguarda novamente o intervalo de tempo para iniciar um novo ciclo de monitoramento (6).

O modelo utiliza uma forma de elasticidade não-bloqueante para não penalizar o processamento das requisições correntes. Para tal, explora a técnica de elasticidade horizontal, através de um orquestrador de elasticidade que é onipresente da aplicação que processa as requisições. Dessa maneira, a execução da aplicação e as ações de elasticidade ocorrem em paralelo, não penalizando a aplicação com a sobrecarga da reconfiguração de recursos. Para notificar os componentes envolvidos sobre a reconfiguração de recursos, o Elastic-RAN disponibiliza uma

Figura 17 – Ciclo de monitoramento de recursos.



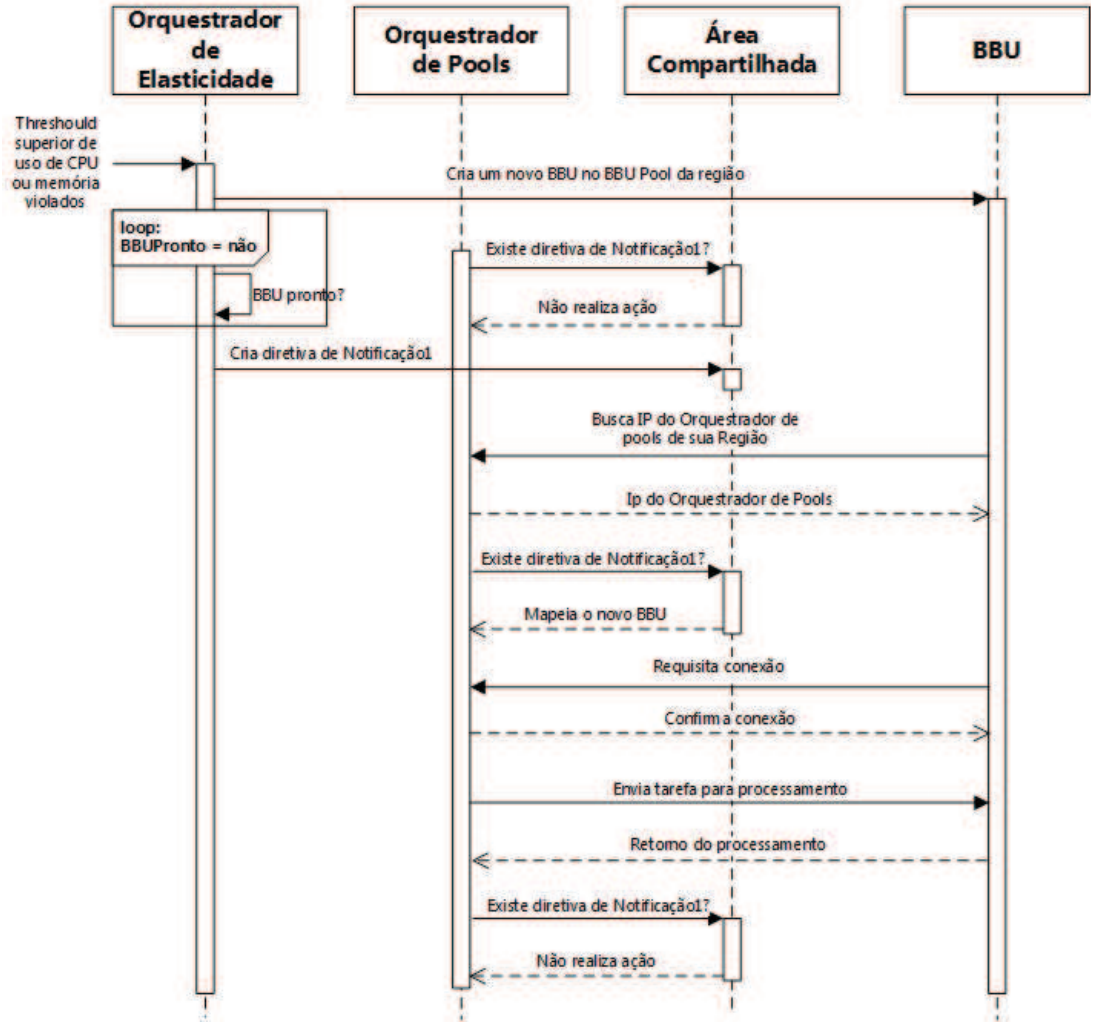
Fonte: Elaborado pelo autor.

área de dados compartilhada para comunicação entre BBU, BBU Pool, Orquestrador do Pool e Orquestrador de Elasticidade. Trata-se de uma área privada para as máquinas virtuais e físicas trocarem informação dentro do ambiente. A utilização desse método para interação entre máquinas é uma abordagem comum em nuvens privadas (MILOJIČIĆ; LLORENTE; MONTERO, 2011; WEN et al., 2012; CAI et al., 2012) e foi comprovado ser uma maneira viável de comunicação entre os componentes de uma arquitetura C-RAN (RAKOVIC et al., 2017).

Na área de dados compartilhada, a troca de mensagens é realizada através de três diretivas de notificação que viabilizam ações de elasticidade de maneira não bloqueante, são elas: (i) notificação 1, realizada pelo Orquestrador de Elasticidade para notificar que novos recursos foram alocados e estão disponíveis para o Orquestrador de Pool; (ii) notificação 2, realizada pelo Orquestrador de Elasticidade para evitar finalizar um processo que ainda esteja executando alguma tarefa, com o intuito de impedir perda de dados; (iii) notificação 3, realizada pelo Orquestrador do Pool após tratar uma notificação 2 recebida anteriormente, garantindo, assim, que a aplicação está em um estado global consistente em que processos podem ser desconectados apropriadamente. Em outras palavras, uma vez recebida uma notificação 2, o Orquestrador do Pool só finaliza sua comunicação com a máquina virtual após finalizar o processamento da tarefa corrente, não enviando mais tarefas para ela. Na sequência, ele envia a Notificação 3, liberando o Orquestrador de Elasticidade para realizar a remoção do recurso. É notória a importância do papel desempenhado pela área de dados compartilhada nesse processo, já que ela viabiliza a notificação e sincronização de todos os componentes e seus processos envolvidos, permitindo assim uma adaptação segura dos recursos.

Através do uso dessas diretivas de notificação, as ações de elasticidade podem ser realizadas sempre com: (i) a adição de f BBU Pools (máquinas físicas) com v BBUs (máquinas virtuais); (ii) adição de v BBUs; (iii) a remoção de f BBU Pools e seus BBUs; ou (iv) a remoção de v

Figura 18 – Fluxo para realizar a alocação de um novo BBU após a violação de uma das *thresholds* superiores de CPU ou memória.

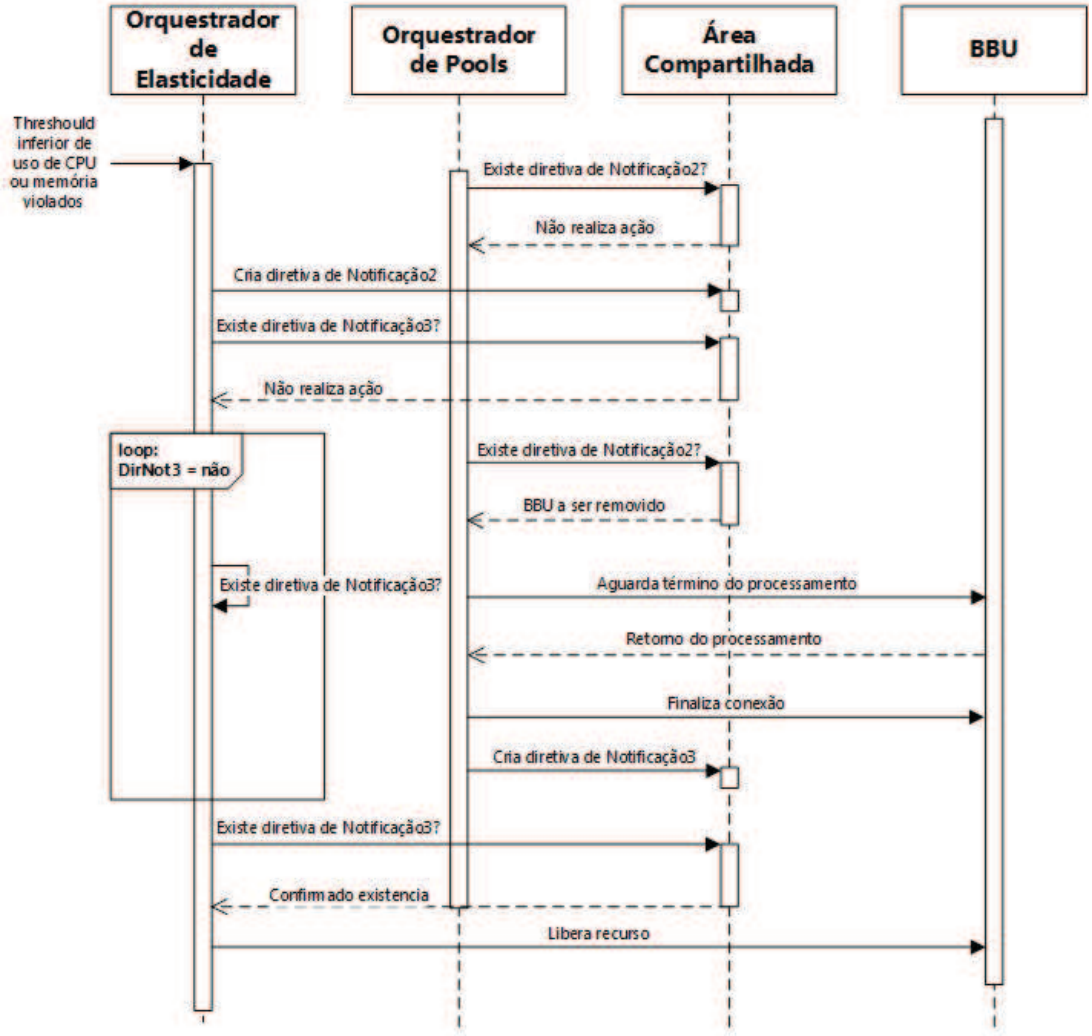


Fonte: Elaborado pelo autor.

BBUs. A Figura 18 demonstra o diagrama de sequência de uma operação de adição BBU após a violação do *threshold* superior de CPU ou memória com a utilização de um grão de elasticidade no tamanho de um BBU. No momento em que o orquestrador de elasticidade observa que o *threshold* superior é violado e existem recursos disponíveis, ele instancia um novo BBU através da API da plataforma da nuvem em um dos BBU Pools da região. Após o BBU inicializar e atingir um estado consistente para processamento, o Orquestrador de Elasticidade cria uma diretiva de Notificação 1. Por sua vez, a cada ciclo do Orquestrador do Pool, antes de distribuir as tarefas de sua fila para os BBUs, o orquestrador verifica na área compartilhada se existe uma diretiva de notificação do tipo 1, e quando encontra alguma, mapeia o novo BBU para processar tarefas juntamente com os demais. Com esse mapeamento, o novo BBU solicita uma conexão no Orquestrador do Pool, para então obter acesso e começar a receber tarefas para processamento.

O fluxo para remoção de recurso pode ser examinado na Figura 19, a qual apresenta um dia-

Figura 19 – Fluxo para realizar a liberação de um BBU após a violação de uma das *thresholds* inferiores de CPU ou memória.



Fonte: Elaborado pelo autor.

grama de sequência de uma operação de remoção de BBU após a violação do *threshold* inferior de CPU ou de memória, com a utilização de um grão de elasticidade no tamanho de um BBU. No momento em que o Orquestrador de Elasticidade verificar uma violação da *threshold* inferior, ele envia uma diretiva de notificação do tipo 2 para o Orquestrador de Pool através da área de dados compartilhada. Essa é uma operação bloqueante no Orquestrador de Elasticidade, pois ele aguarda até receber uma notificação do tipo 3 na área compartilhada. A cada ciclo do Orquestrador do Pool, antes de distribuir as tarefas de sua fila para processamento nos seus BBUs, ele verifica a área de dados compartilhada em busca de notificação do tipo 2. Ao encontrar, ele captura o resultado pendente de um BBU a ser removido e encerra a conexão com ele para que não lhe sejam mais encaminhadas tarefas. Findo esse processo, envia uma notificação do tipo 3. Essa notificação é, então, lida pelo Orquestrador de Elasticidade, que agora pode realizar a remoção do BBU.

No modelo Elastic-RAN, o Orquestrador do Pool possui uma lógica de escalonamento au-

tomático para mapear as máquina virtuais, a fim de torna-las BBUs aptos para processamento. Como forma de evitar que ocorram ações elásticas desnecessárias, após cada operação de elasticidade, ocorre um período de *cooldown*, o qual refere-se ao período de tempo onde não é permitido a realização de novas operações de elasticidade (JAMSHIDI; AHMAD; PAHL, 2014). Tal técnica visa evitar que ocorram operações elásticas desnecessárias, pois uma nova instância pode estar criada pelo Orquestrador de Elasticidade, mas ainda não estar efetivamente mapeada e processando tarefas no Orquestrador do Pool.

No presente modelo, adota-se o período de *cooldown* de dois ciclos de monitoramento, onde após o Orquestrador de Elasticidade entregar um novo recurso, é aguardado dois ciclos para permitir novas ações elásticas.

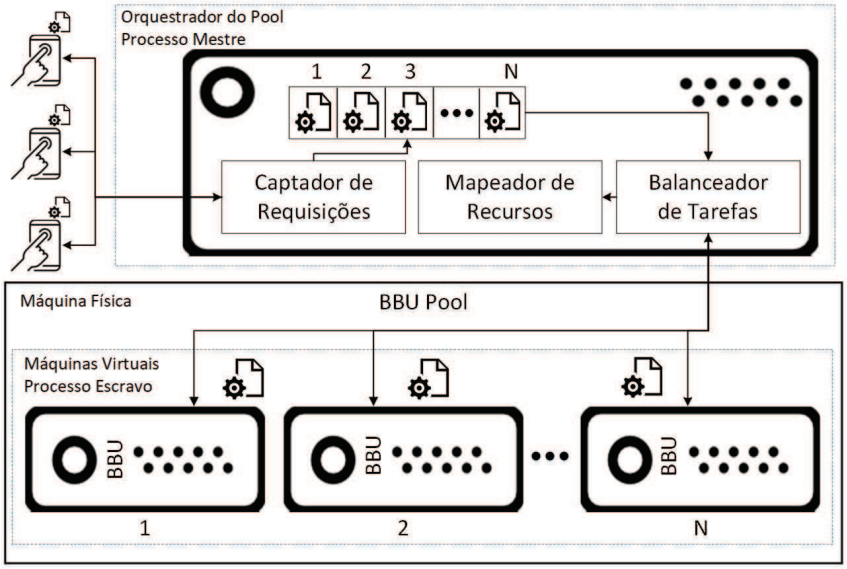
4.4 Orquestrador do Pool

Responsável pelo gerenciamento, mapeamento e distribuição dos trabalhos entre os pools da sua região, o Orquestrador do Pool é um componente que possui uma fila, cujo preenchimento se dá com tarefas provenientes das antenas, as quais serão, individualmente, encaminhadas para os respectivos BBU Pools e, então, processadas por BBUs. Após as tarefas processadas serem recebidas, o Pool retorna o resultado para a antena que originou a requisição. A Figura 20 apresenta a arquitetura do Orquestrador do Pool e seus componentes no modelo, onde é explorado o modelo de computação Mestre/Escravo (BUYYA, 1999), o qual é utilizado em diferentes trabalhos para ambientes de computação em nuvem (TRAN et al., 2015; KUMAR et al., 2014; ROSA RIGHI et al., 2016a). No modelo proposto nesta dissertação, o Orquestrador do Pool atua como processo mestre e distribui as tarefas para processamento em seus BBUs, que atuam como processos escravos.

O Orquestrador do Pool é executado de forma iterativa para explorar as diretivas de comunicação apresentadas na Seção 4.3, com as quais a cada Notificação 1 recebida, o processo mestre lê as informações referentes aos novos BBUs e então aguarda a conexão desse novo recurso para reorganizar a distribuição das tarefas. Caso venha a receber uma Notificação 2, o processo mestre remove do seu grupo de recursos aqueles aos BBUs que serão desativados. Após, o processo gera uma Notificação 3, liberando a remoção de recursos desses recursos para o Orquestrador de Elasticidade. Busca-se com essa abordagem facilitar a reconfiguração de recursos sem alterar a semântica da aplicação (SPINNER et al., 2014), além de garantir um estado global consistente para a aplicação.

O fluxo de processamento de requisições tem início com o mapeamento inicial de recursos do pool, no qual são definidas as primeiras configurações para cada um dos BBUs com os seus endereços de IP para conexão e comunicação. Todas as trocas de mensagens entre o Orquestrador e seus recursos ocorrem de forma assíncrona, de modo que o mestre transmite informações para os recursos de forma não bloqueante, mas recebe os dados de resultado de maneira síncrona. O orquestrador vai, então, distribuindo as requisições da sua fila, de uma em

Figura 20 – Arquitetura do Orquestrador do Pool com seus principais componentes.



Fonte: Elaborado pelo autor.

uma, para ser processada em cada um de seus BBUs. Essa abordagem segue o estudo de Lee et al. (2011), onde o autor relata que a melhor abordagem para desempenho e eficiência em aplicações que exigem alto poder de processamento é a utilização de uma máquina virtual em cada núcleo do processador, executando apenas uma tarefa por vez em cada máquina.

4.5 Modelo de Elasticidade

A presente Seção apresenta os métodos e as técnicas de monitoramento e elasticidade para a tomada de decisões utilizados pelo Elastic-RAN. A Subseção 4.5.1 apresenta o método utilizado para avaliação da carga no ambiente para a tomada de ações elásticas, bem como a técnica do grão de elasticidade adaptativo. Já na Subseção 4.5.2, são expostas as regras e o funcionamento dos *thresholds* para a elasticidade multinível.

4.5.1 Cálculo de carga e tomada de decisões para ações elásticas

Nos moldes do que foi proposto na Seção 4.3, o Orquestrador de Elasticidade extraí a carga de utilização de CPU, de memória e de taxa de transferência de dados dos BBU Pools e de seus BBUs. Para cada uma dessas métricas, aplica-se um cálculo de séries temporais, que leva em consideração o histórico dos valores calculados anteriormente para obter a carga geral do sistema, aqui denominada de score de processamento (SP). Por essa razão, cada ciclo de monitoramento consiste em uma observação em que os valores são coletados e armazenados para serem utilizados posteriormente. Salienta-se que as ações de elasticidade só começam a ser tomadas a partir do momento em que existe uma quantidade mínima de observações gravadas, que é informada previamente no momento de execução do Elastic-RAN.

Assim como utilizado por plataformas comerciais como Amazon AWS[1] (CHIU; AGRAWAL, 2010), Microsoft Azure[2] (ROLOFF et al., 2012) e Nimbus[3] (MARSHALL; KEAHEY; FREEMAN, 2010), o Elastic-RAN faz o uso de $thresholds$ superiores e inferiores, que devem ser informados no momento da inicialização através do arquivo de configuração do Orquestrador de Elasticidade. Destarte, a cada ciclo de monitoramento, essas *thresholds* são comparadas com o score de utilização dos recursos calculados e, caso alguma delas seja ultrapassada, o orquestrador realiza uma ação elástica para alterar os recursos do ambiente.

Como o modelo Elastic-RAN oferece orquestração de elasticidade multinível, tanto no nível de BBU Pool (máquina física) quanto de BBU (máquina virtual) de forma individual, leva-se em consideração, para o cálculo do score de processamento do sistema, métricas pertinentes para C-RAN (CPU, memória e rede). O score de processamento pode ser obtido através da equação 4.1, na qual a função SP (o, bp) resulta na média aritmética de todas as cargas de processamento dos BBUs na observação o para o BBU Pool bp. Onde qb representa a quantidade de BBUs ativos no pool bp e a função SES(o, bp), representada pela Equação 4.2, obtém a carga de uma das métricas avaliadas para a tomada de decisões elásticas.

$$SP(o, bp) = \frac{\sum_{i=0}^{qb} SES(o, bp)}{qb} \tag{4.1}$$

$$SES(o, bp) = \begin{cases} \frac{Carga(o,bp)}{2}, & \text{Se } o = 0 \\ \frac{SES(o-1,bp)}{2} + \frac{Carga(o,bp)}{2}, & \text{caso contrário} \end{cases} \tag{4.2}$$

Na linha dos trabalhos de Rosa Righi et al. (2016a); Facco Rodrigues et al. (2017), aqui o método de Simple Exponential Smoothing (SES) é utilizado no cálculo de cada uma das métricas para aplicar uma suavização exponencial simples sobre elas. Esse cálculo ocorre em cada um dos BBU Pools, onde o é a observação mais recente e o - 1 a média aritmética das observações anteriores. É possível, dessa forma, obter o valor de SP, o qual é confrontado com os *thresholds* superior e inferior de cada uma das métricas, podendo acarretar em ações de elasticidade. A função SES opera com o conceito de *Aging* (TANENBAUM; WETHERALL, 2011), através da implementação do método apresentado na seção 2.4, o qual emprega uma suavização exponencial simples em que a última medida tem a maior influência sobre o índice de carga. Esse conceito é aplicado no modelo Elastic-RAN no intuito de tratar situações de picos de carga, prevenindo a realização de ações elásticas para falsos-positivos ou falsos-negativos (TANENBAUM; WETHERALL, 2011). Basicamente, essa técnica atribui um peso maior para a observação mais recente, dividindo por uma potência de 2 cada elemento subsequente da série histórica. Por exemplo, considerando um *threshold* superior de 80% e observações como 72, 70, 78, 71 e 81, sendo essa última a mais recente, a previsão de carga lp irá informar o valor 74,62 como resultado da seguinte expressão: SES(o,p) = $\frac{81}{2}$ + $\frac{71}{4}$ + $\frac{78}{8}$ + $\frac{70}{16}$ + $\frac{72}{32}$. Dessa maneira,

[1] https://aws.amazon.com
[2] https://azure.microsoft.com
[3] http://www.nimbusproject.org

Algoritmo 1: Cálculo do tamanho do grão para ações de elasticidade.

```
   Entrada: percentualAumentoParaGrao
   Saída: novoTamanhoDoGrao
1  se Condição1(percentualAumentoParaGrao) E Condição10(qtdMaquinasVirtuaisDisponiveis) então
2      se Condição11(percentualAumentoParaGrao) então
3          novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasVirtuais, adição, Funcao.Linear)
4      fim
5
6      se Condição12percentualAumentoParaGrao) então
7          novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasVirtuais, adição, Funcao.Exponencial)
8      fim
9
10 fim
11 se Condição5(percentualAumentoParaGrao) E Condição9(qtdMaquinasFisicasDisponiveis) então
12     se Condição11(percentualAumentoParaGrao) então
13         novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasFisicas, adição, Funcao.Linear)
14     fim
15
16     se Condição12(percentualAumentoParaGrao) então
17         novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasFisicas, adição, Funcao.Exponencial)
18     fim
19
20 fim
21 se Condição2(percentualAumentoParaGrao) E Condição8(qtdMaquinasVirtuaisAtivas) então
22     se Condição11(percentualAumentoParaGrao) então
23         novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasVirtuais, redução, Funcao.Linear)
24     fim
25
26     se Condição12(percentualAumentoParaGrao) então
27         novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasVirtuais, redução, Funcao.Exponencial)
28     fim
29
30 fim
31 se Condição6(percentualAumentoParaGrao) E Condição7(qtdMaquinasFisicasAtivas) então
32     se Condição11(percentualAumentoParaGrao) então
33         novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasFisicas, redução, Funcao.Linear)
34     fim
35
36     se Condição12(percentualAumentoParaGrao) então
37         novoTamanhoDoGrao = CalculaTamanhoDoGrao(graoDeMaquinasFisicas, redução, Funcao.Exponencial)
38     fim
39
40 fim
```

o último valor "81"não irá disparar nenhuma ação de elasticidade, uma vez que é avaliado como falso-positivo. Através dessa estratégia é possível amortizar a importância dos picos, uma vez que são considerados os dados históricos do sistema para evitar alocações provenientes de ruídos nas observações.

Como diferencial frente às abordagens tradicionais de elasticidade para C-RAN, o Elastic-RAN explora um mecanismo elástico que atua com um grão de elasticidade adaptativo. Assim, durante a execução da aplicação, o tamanho do grão a ser utilizado em cada uma das operações é redimensionado com base na utilização corrente dos recursos disponíveis. A cada operação de adição ou remoção de recursos detectada, realiza-se uma operação para calcular o novo tamanho do grão, de maneira que esse grão se aproxime das necessidades de recursos correntes. A lógica utilizada para esse cálculo se encontra no Algoritmo 1, no qual ocorre um comparativo entre a carga do sistema calculada no ciclo corrente, com a carga do sistema calculada no ciclo anterior. Para facilitar a compreensão, as suas condições e ações são apresentadas na Tabela 2.

Caso seja observado um aumento X percentual no ciclo corrente $SP(o, bp)_{cor}$ em relação

Tabela 2 – Descrição das condições e ações utilizadas no cálculo do grão de elasticidade.

Declaração	Descrição
Condição1(x)	O uso corrente de CPU ou memória é x% maior ou igual ao histórico de uso dessas métricas
Condição2(x)	O uso corrente de CPU ou memória é x% menor que o histórico de uso dessas métricas
Condição3(x)	O uso corrente de memória é x% maior ou igual ao histórico de uso da memória
Condição4(x)	O uso corrente de memória é x% menor que o histórico de uso de memória
Condição5(x)	O uso corrente da rede é x% maior ou igual ao histórico de uso da rede
Condição6(x)	O uso corrente da rede é x percentual menor que o histórico de uso de uso da rede
Condição7(f)	Existe um mínimo de f máquinas físicas ativas
Condição8(v)	Existe um mínimo de v máquinas virtuais ativas
Condição9(f)	Existe um mínimo de f máquinas físicas disponíveis
Condição10(v)	Existe um mínimo de v máquinas virtuais disponíveis
Condição11(x)	A variação de uso corrente em relação ao histórico é maior que o percentual linear e menor que o percentual exponencial informados no arquivo de configuração
Condição12(x)	A variação de uso corrente em relação ao histórico é maior que o percentual exponencial informados no arquivo de configuração

Fonte: Elaborado pela autor.

Tabela 3 – Funções disponíveis para cálculo do tamanho do grão de elasticidade.

Tipo	Definição da Função	Parâmetros	
		a	x
Linear	a $\in \mathbb{R}$ tal que f(x) = a + x para todo x $\in \mathbb{R}$	1	tamanhoAtualGrao
Exponencial	a $\in \mathbb{R}$ tal que f(x) = a^a + x para todo x $\in \mathbb{R}$	2	tamanhoAtualGrao

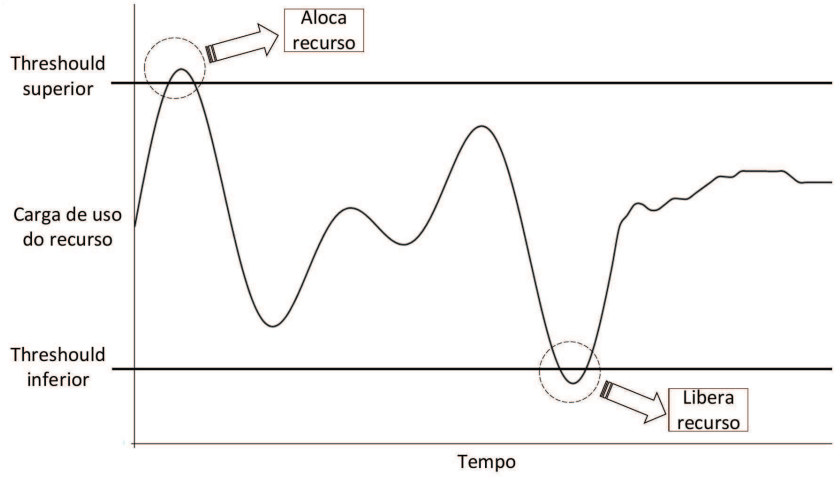
Fonte: Elaborado pela autor.

ao ciclo passado $SP(o, bp)_{pas}$, realiza-se um incremento no tamanho do grão de elasticidade conforme uma função matemática informada no arquivo de configuração do Orquestrador de Elasticidade. Esse percentual X de aumento também é informado nesse mesmo arquivo de configuração. Em contrapartida, caso seja verificada uma redução de X percentual no ciclo corrente, quando comparado ao ciclo passado, é realizado um decremento no tamanho do grão conforme a função matemática informada. As funções matemáticas disponíveis para dimensionamento do grão se encontram na Tabela 3. Na Seção 6.1, encontram-se testes realizados com essas funções combinadas, objetivando-se atingir a melhor abordagem final para o cálculo do grão. É de se salientar, ademais, que é premissa do modelo que o grão de elasticidade seja de, no mínimo, uma máquina virtual, para o nível de BBU, ou de uma máquina física, para o nível de BBU Pool.

4.5.2 *Thresholds*

Para realizar ações elásticas, o modelo Elastic-RAN utiliza *thresholds* fixos de maneira reativa. Essa abordagem baseia-se em regras previamente definidas por limiares que serão aplicados em cada uma das métricas. Para tanto, o modelo considera três diferentes métricas (CPU, memória e rede), devendo informar, diretamente no arquivo de configuração que é consumido pelo Orquestrador de Elasticidade, o percentual de uso limite superior e inferior para cada uma das métricas. No início de cada ciclo de monitoramento, em cada BBU Pool e seus BBUs,

Figura 21 – Elasticidade reativa baseada em *thresholds* do Elastic-RAN.



Fonte: Elaborado pelo autor.

ocorre a aplicação da técnica de *Aging* sobre os valores de uso corrente dessas métricas, a qual tem por objetivo obter os seus scores de uso. Esses scores serão comparados com o percentual de uso informado para detectar violações nas *thresholds* e realizar ações de elasticidade. A Figura 21 demonstra o modelo de elasticidade reativo baseado em *thresholds* empregado no Elastic-RAN, com um exemplo no qual ocorrem dois momentos de violação dos *thresholds* e que, como consequência, ocasionam ações elásticas.

Como o Elastic-RAN atua com elasticidade multinível, são consideradas diferentes métricas para efetuar ações elásticas em cada um dos níveis. No primeiro nível, orquestrador de elasticidade tem por objetivo realizar ações elásticas para BBU Pool (máquina física), tendo como métrica a rede (volume de dados trafegado), de modo a evitar que ocorra um grande volume de requisições na interface da rede de um mesmo pool. Assim, ao atingir a *threshold* da rede, ele executa operações para ativar ou desativar f BBU Pools. Já no segundo nível, o orquestrador realiza ações elásticas nos BBUs (máquinas virtuais) de forma individual. No caso, como os BBUs precisam realizar o processamento das tarefas de forma intensiva, o orquestrador considera o uso de CPU e memória como métrica e, ao atingir o *threshold*, efetua ações elásticas para adicionar ou remover v BBUs.

Tendo em vista que intuito do Elastic-RAN é processar as tarefas com desempenho e menor custo de infraestrutura possível, o modelo realiza o monitoramento e o balanceamento dos recursos de infraestrutura de forma constante. Assim, a lógica e os principais procedimentos para a orquestração da elasticidade são encontradas no Algoritmo 2, enquanto que as suas condições e ações são apresentadas na Tabela 4.

Algoritmo 2: Ciclo de monitoramento do orquestrador de elasticidade

```
   Entrada: observacao, bbuPools
1  enquanto (sistemaAtivado) faça
2      ColetaInformacoesMonitoramento(bbuPools, observacao);
3      CalculaScoresDeProcessamento(observacao);
4      se Condicao1(observacao) então
5          se Condicao2(observacao) então
6              se Condicao7(observacao) então
7                  Ação1(CalcularTamanhoDoGrao());
8              fim
9          fim
10         se Condicao4(observacao) então
11             se Condição6() então
12                 Ação3(CalcularTamanhoDoGrao());
13             fim
14         fim
15         se Condicao3(observacao) então
16             se Condicao9(observacao) então
17                 Ação2(CalcularTamanhoDoGrao());
18             fim
19         fim
20         se Condicao5(observacao) então
21             se Condição8() então
22                 Ação4(CalcularTamanhoDoGrao());
23             fim
24         fim
25     fim
26 fim
```

Tabela 4 – Descrição das condições e ações utilizadas no algoritmo de orquestração de elasticidade.

Declaração	Descrição
Condição1(i)	i é maior que quantidade mínima de monitoramentos para calcular o score de uso.
Condição2(o)	Média do uso de CPU ou memória de todos os BBUs do Orquestrador do Pool for maior que o *threshold* superior de CPU ou memória.
Condição3(o)	Média do uso de CPU ou memória de todos os BBUs do Orquestrador do Pool for menor que o *threshold* inferior de CPU ou memória.
Condição4(o)	Média de uso da rede dos BBU Pools for maior que o *threshold* superior de uso da rede.
Condição5(o)	Média de uso da rede dos BBU Pools for menor que o *threshold* inferior de uso da rede.
Condição6()	Existem máquinas físicas disponíveis.
Condição7()	Existem máquinas virtuais disponíveis.
Condição8()	Existem máquinas físicas ativas.
Condição9()	Existem máquinas virtuais ativas.
Ação1(v)	Adiciona v máquinas virtuais.
Ação2(v)	Remove v máquinas virtuais.
Ação3(f)	Adiciona f máquinas físicas.
Ação4(f)	Remove f máquinas físicas.

Fonte: Elaborado pela autor.

5 METODOLOGIA DE AVALIAÇÃO

Apresentaremos, neste capítulo, a metodologia adotada para avaliação do modelo Elastic-RAN, sendo que a Seção 5.1 tratará das métricas para avaliação do modelo, enquanto que na Seção 5.2 serão expostos os detalhes de implementação do protótipo. A infraestrutura utilizada para execução dos experimentos será apresentada na Seção 5.3 e, após, na Seção 5.4, a aplicação intensiva de CPU e de rede desenvolvida para execução dos testes é descrita. Na última Seção – 5.5 – são detalhados os parâmetros e cenários de testes propostos, bem como os comportamentos dos diferentes perfis de cargas.

5.1 Métricas de Avaliação

Para avaliar o desempenho da elasticidade do modelo proposto, serão abordados alguns métodos que se aplicam a elasticidade para arquitetura C-RAN, observando o desempenho e a eficiência na utilização dos recursos. Além disso, também são adotadas métricas para mensurar o custo e consumo energético dos recursos utilizados durante os experimentos.

5.1.1 Eficiência

Para observar os resultados da elasticidade multinível do modelo, serão utilizadas duas métricas propostas no trabalho de Rosa Righi et al. (2016a): Speedup Elástico (SP) e Eficiência Elástica (EF). Tais métricas são extensão dos conceitos de Speedup e Eficiência tradicionais (KUMAR et al., 2002) e foram propostas com o objetivo de avaliar elasticidade horizontal, na qual instâncias de máquinas virtuais são removidas, adicionadas ou consolidadas durante o processamento da aplicação, alterando a quantidade de recursos disponíveis. Para avaliar a elasticidade do modelo, considera-se um ambiente homogêneo, no qual cada instância de máquina virtual consegue executar 100% de um núcleo do processador da máquina física. SP é calculado pela função SP(q, i, s) conforme Equação 5.1, na qual q representa a quantidade inicial de máquinas virtuais enquanto s e i representam os limites superior e inferior para a quantidade de máquinas virtuais definidas no SLA. t_{ne} e t_e fazem referência aos tempos de execução da aplicação executada em cenários não elásticos e elásticos. Dessa forma, $t_{ne}(i)$ é interpretado com o menor número de máquinas virtuais possíveis.

$$SP(q,i,s) = \frac{t_{\text{ne}}(i)}{t_{\text{e}}(q,i,s)} \tag{5.1}$$

A função EF(q, i, s), representada pela Equação 5.2, corresponde ao cálculo da eficiência elástica. Os parâmetros dessa função são os mesmos da função SP. A eficiência representa o quanto eficaz é o uso dos recursos, sendo esse posicionado como denominador da fórmula. Diferentemente da equação tradicional de cálculo da Eficiência, na equação da Eficiência Elástica

os recursos são flexíveis. Razão pela qual criou-se um mecanismo para alcançar um único valor de forma coerente. Para tanto, EF assume a execução de um sistema de monitoramento que captura o tempo gasto em cada configuração de máquinas virtuais. A Equação 5.3 apresenta a métrica Recursos utilizada no cálculo de EF, indicando o uso de recursos da aplicação em que $p_{te}(j)$ é o intervalo de tempo em que a aplicação foi executada com j máquinas virtuais. Caso não ocorram operações de elasticidade, $p_{te}(j)$ e $t_e(q, i, s)$ resultarão no mesmo valor para j=q. A Equação 5.2 apresenta o parâmetro q no numerador, que está fazendo a multiplicação com o Speedup Elástico.

$$EF(q, i, s) = \frac{SE(q, i, s) \times q}{Recursos(q, i, s)} \tag{5.2}$$

$$Recursos(o, bp) = \sum_{j=i}^{s} (j \times \frac{p_{\text{te}}(j)}{t_{\text{e}}(q, i, s)}) \tag{5.3}$$

5.1.2 Energia e Custo

Não sendo possível inferir de forma precisa o consumo de recursos através da eficiência elástica, buscou-se uma forma de complemento para a análise do consumo, aqui chamada energia. A Energia está definida na Equação 5.4, sendo que neste trabalho ela emprega a mesma ideia do modelo utilizado pela Amazon e Microsoft, no qual considera-se a quantidade de máquinas virtuais em cada unidade de tempo, que, no caso dessas empresas, é de normalmente uma hora. A variável $p_{te}(j)$ representa o intervalo de tempo em que a aplicação foi executada com j instâncias de máquinas virtuais. Dessa maneira, esse intervalo depende do valor de p_{te}(segundos, minutos ou horas) em que a estratégia é somar a quantidade de máquinas virtuais utilizadas em cada unidade de tempo. Sendo assim, a Energia mensura o uso de i até s instâncias de máquinas virtuais, considerando o período de execução parcial em cada reorganização dos recursos utilizados. O uso de energia se torna pertinente para realizar comparações entre diferentes aplicações que utilizam elasticidade variando de i até s.

$$Energia(i, s) = \sum_{j=l}^{s} (j \times p_{\text{te}}(j)) \tag{5.4}$$

No intuito de estimar a viabilidade da elasticidade em diversas situações, adotou-se uma métrica que calcula o custo da execução a partir da multiplicação do tempo total de execução das tarefas pela energia utilizada para a execução, conforme se pode ver na Equação 5.5. Essa ideia é uma adaptação do custo de processamento com uso em cenários elásticos. Alguns autores, como Rosa Righi et al. (2016a) e Baliga et al. (2011), apontam que o consumo de energia é proporcional ao uso de recursos. Dessa maneira, o objetivo é obter o menor custo possível durante os processamentos, pois isso retrata a relação entre o tempo de execução e o consumo de energia.

$$Custo = TempoExecucaoTarefas \times Energia \tag{5.5}$$

5.1.3 Tráfego da Rede

Para a métrica de rede, avaliou-se a vazão das tarefas e o tempo de respostas para seu processamento. A vazão expressa a quantidade máxima de dados que podem ser transportados e processados em uma determinada unidade de tempo. Dessa forma, definiu-se vazão através da Equação 5.6, onde t_{bp} refere-se ao total de tarefas processados em cada um dos BBU Pools bp. Esse valor é, então, divido pelo tempo total de execução das tarefas. Além disso, observou-se o tempo de resposta das requisições.

$$Vazao(bp) = \frac{\sum_{i=0}^{qb} t_{\text{bp}}}{TempoExecucaoTarefas} \tag{5.6}$$

5.2 Implementação do protótipo

Para avaliar o modelo Elastic-RAN, foi desenvolvido um protótipo que abarca os componentes necessários para implementação do Orquestrador de Elasticidade e Orquestrador de Pool. Enquanto o primeiro realiza operações de monitoramento e ações de elasticidade sobre uma infraestrutura de nuvem privada OpenNebula[1], as quais são levadas a efeito por intermédio de uma API Java, fornecida pelo próprio OpenNebula, e que viabiliza a obtenção das informações de uso corrente dos recursos da plataforma e dos recursos ativos, o segundo efetua as operações de distribuição de tarefas, a reestruturação de processos e a atualização da topologia da rede para os seus BBUs processarem e retornarem os resultados.

A linguagem de programação utilizada no desenvolvimento foi Java[2], devido a compatibilidade com o protocolo XML-RPC incorporado pelo *middleware* do OpenNebula. XML-RPC é uma API que permite a interoperabilidade entre softwares executados em diferentes sistemas operacionais, rodando em ambientes diferentes para fazer chamadas de procedimento através do protocolo HTTP.

No *middleware* da nuvem, além de dois *templates* de máquinas virtuais para diferenciar as funções que o processo da aplicação empenhará no início de sua execução, também configurou-se máquinas físicas. Criou-se, assim, um modelo para o Orquestrador de Pool e outro para o BBU, cujas identificações são passadas para o Orquestrador de Elasticidade, que utiliza os modelos no momento de instanciar novos recursos. No ambiente do OpenNebula, configurou-se uma área de dados compartilhada, a qual todos os recursos (máquinas físicas e virtuais) possuem acesso, utilizando NFS (*Network File System*). Cita área de dados possibilita a comunicação entre o Orquestrador de Elasticidade e o Orquestrador do Pool. Nessa região também

[1] https://opennebula.org/
[2] http://www.oracle.com/java

são armazenadas as informações do histórico da execução da aplicação e as informações do SLA e do arquivo de configuração. O SLA do modelo proposto nesta dissertação contém: (i) a quantidade mínima e máxima de máquinas físicas a serem utilizadas; e (ii) a quantidade mínima e máxima de máquinas virtuais a serem utilizadas. Já o arquivo de configuração contém: (i) o percentual de aumento no uso das métricas para alteração do tamanho do grão; (ii) a função matemática utilizada para cálculo do tamanho do grão; (iii) os valores das *thresholds* superiores e inferiores de uso de CPU, memória e tráfego na rede; (iV) o tempo do intervalo do ciclo de monitoramento dos recursos; e (v) a quantidade de ciclos de monitoramento para o período de *cooldown*.

5.3 Infraestrutura

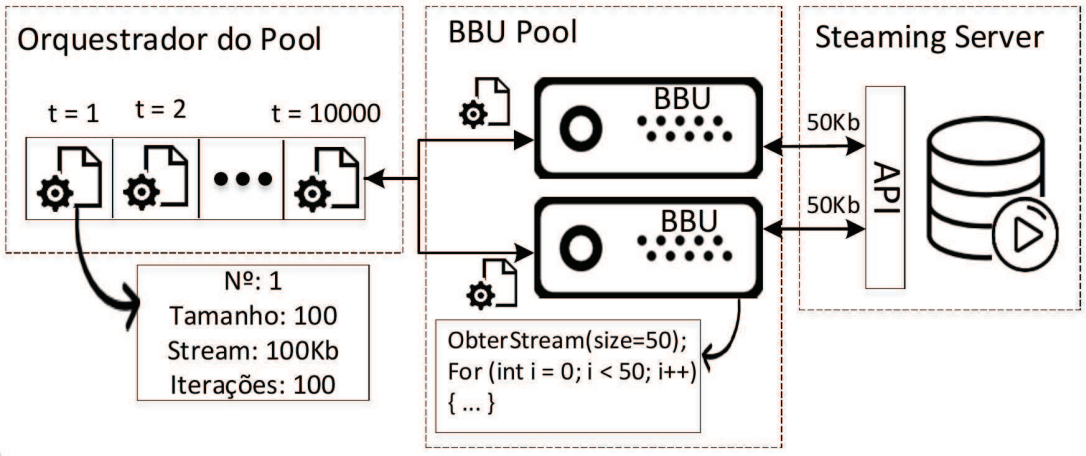
O presente modelo foi executado no laboratório C01 413 da Universidade do Vale do Rio dos Sinos. Para os experimentos, foram utilizados doze equipamentos homogêneos com processadores de dois núcleos de 2.9 GHz e 4 GB de memória RAM, interconectados através de uma rede 100 Mbps. Onze equipamentos foram configurados com a plataforma de nuvem OpenNebula Sunstone versão 4.12.1. Além disso, em um desses equipamentos foi instalado o servidor OpenNebula para servir como o Front-End da nuvem. Nesse equipamento, configurou-se o servidor SSH e a área de dados compartilhada. Em um outro equipamento, realizou-se a execução do Orquestrador de Elasticidade do Elastic-RAN. Já os outros dez equipamentos foram configurados como nós OpenNebula, para receber e executar máquinas virtuais. Dessa maneira, o orquestrador tem a capacidade de manipular simultaneamente até 16 instâncias de máquinas virtuais. Lembrando que, para a execução mínima do Elastic-RAN, é necessário pelo menos uma máquina física com uma instância do tipo Orquestrador de Pool e outra do tipo de BBU.

5.4 Aplicação para avaliações

Fitando avaliar o modelo Elastic-RAN, criou-se uma aplicação intensiva de CPU e de rede, que tem por finalidade simular os diferentes perfis de carga. Essa aplicação realiza a requisição de *streamings* de diferentes tamanhos para um servidor que expõem uma API para distribuir esses dados. Em paralelo, a aplicação também efetua processamento intensivo de CPU para simular a decodificação de blocos realizada em um BBU, onde a quantidade de blocos para cálculo é o mesmo tamanho da quantidade de bytes da *streaming* requisitada na tarefa.

A Figura 22 apresenta, em alto nível, o funcionamento da aplicação de testes desenvolvida, a qual é processada pelo protótipo do modelo, sendo que as cargas de trabalhos recebidas no Orquestrador do Pool de determinada região são um lote de requisições com diferentes parâmetros que são convertidos para diferentes requisitos de processamento, de modo que seja possível reproduzir os perfis de cargas propostos. Assim, o Orquestrador do Pool, que atua como um processo mestre do BBU Pool, distribui t + 1 tarefas referentes a cada uma das requisições de

Figura 22 – Estrutura da aplicação de testes.



Fonte: Elaborado pelo autor.

sua fila entre os seus BBUs para processamento. Para cada bateria de execução, são carregadas 10000 tarefas que são distribuídas e processadas. Dado que t possue as propriedades que definem o tamanho da *streaming* e a quantidade de iterações para a decodificação de blocos, quanto maior forem essas propriedades, maior é a carga computacional e carga na rede.

5.5 Parâmetros e cenários de testes

O Elastic-RAN, assim como a maioria dos trabalhos que exploram elasticidade, utiliza a técnica de elasticidade reativa baseada em *thresholds*, que são configuradas com valores que influenciam os algoritmos de tomada de decisão. Para eleger a *threshold* mais apropriada para o algoritmo de elasticidade, essas configurações foram levadas em consideração no modelo proposto. Assim, após contemplar os trabalhos na literatura, a *threshold* superior escolhida foi de 70% (DAWOUD; TAKOUNA; MEINEL, 2011; AL-HAIDARI; SQALLI; SALAH, 2013; ROSA RIGHI et al., 2016b,a) e a *threshold* inferior foi de 30% (AL-HAIDARI; SQALLI; SALAH, 2013; ROSA RIGHI et al., 2016b,a).

Nada obstante, visando avaliar o impacto de diferentes configurações de *thresholds* no desempenho e consumo de recursos do modelo, foram definidos quatro diferentes cenários para a execução, a saber:

- C1: Execução da aplicação com a quantidade mínima de recursos (uma máquina física com duas máquinas virtuais) e sem o uso da elasticidade. Dessa forma, as alocações e as consolidações dos recursos não são habilitados, executando todo o processo com essa mesma quantidade de máquinas.

- C2: Execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade apenas no nível de BBU (máquina virtual) e com o grão de elasticidade estático;

- C3: Execução da aplicação iniciando com a quantidade mínima de recursos, com elasti-

Tabela 5 – Funções para geração dos diferentes perfis de carga. Para carga(x), x representa o índice da requisição corrente.

Carga	Função	Parâmetros		
		t	v	r
Crescente	carga(x) = x * v * r	-	0,2	500
Decrescente	carga(x) = t - (x * v * r)	1000000	0,2	500
Constante	carga(x) = t	500000	-	-
Oscilante	carga(x) = (r * seno(v * x)) + (v * r) + t	500	0,00125	500000
Exponencial	carga(x) = $e^{((x/t)*v)+r}$	1000	1,5885	1000

Fonte: Elaborado pela autor.

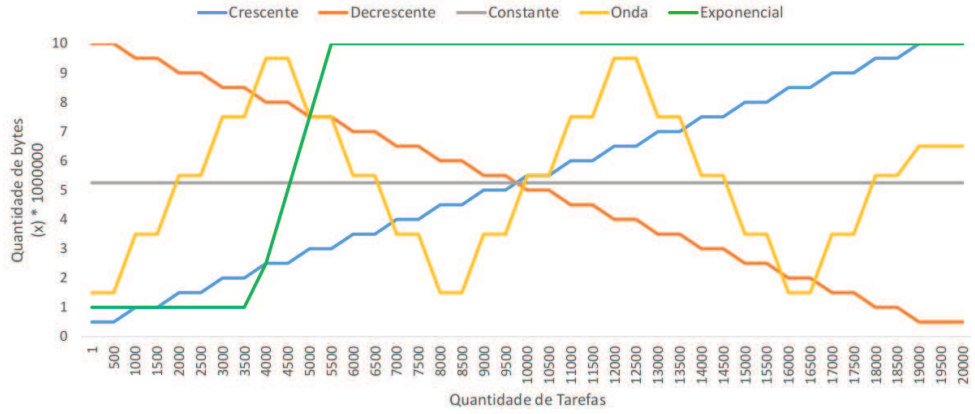
cidade multinível, para BBU Pool (máquina física) e BBU de forma individual, além de utilizar um grão de elasticidade estático;

- C4: Execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade multinível, para BBU Pool e BBU de forma individual, com o grão de elasticidade adaptativo. Objetivando uma abordagem final para adaptação do grão, nesse cenário são propostos três subcenários que exploram diferentes técnicas para calcular o tamanho do grão, são eles: (i) C4.1: de forma linear; (ii) C4.2: de forma exponencial; (iii) C4.3: de forma linear e exponencial, considerando o nível da alteração da carga.

Sabendo que o fluxo de entrada em arquiteturas C-RANs pode ser intenso, inconstante ou irregular, foram executadas cargas de processamento com diferentes intensidades – Crescente, Decrescente, Constante, Oscilante e Exponencial. Essas cargas foram geradas através das funções apresentadas na Tabela 5 e foram escolhidas em razão de serem consideradas uma forma representativa de avaliar a elasticidade em nuvens computacionais (ISLAM et al., 2012). É de se salientar que o uso de diferentes comportamentos para a mesma aplicação é uma abordagem empregada com frequência para observar como a carga de entrada pode impactar em gargalos e na elasticidade dos recursos (MAO; HUMPHREY, 2011; ISLAM et al., 2012). A Figura 23 apresenta os padrões de carga gerados pelas funções, os quais serão utilizados para avaliar cada um dos cenários de teste.

Adotou-se, para o intervalo entre as observações de monitoramento do Orquestrador de Elasticidade, o tempo de 15s, valor que leva em consideração o tempo total que o *middleware* da nuvem OpenNebula gasta para atualizar as métricas de todas as máquinas, que pode atingir o tempo de até um segundo por máquina. Cada comportamento de carga deve ser executado em cada um dos quatro cenários com as *thresholds* e os intervalos definidos, o que gera as 75 possíveis execuções que podem ser observadas da leitura da Tabela 6.

Figura 23 – Padrões de cargas de processamento para avaliação dos cenários de testes.



Fonte: Elaborado pelo autor.

Tabela 6 – Combinações das configurações para os cenários de testes. Ts = *threshold* superior; Ti = *threshold* inferior; im = intervalo do monitoramento; mfi = quantidade de máquinas físicas iniciais; mvi = quantidade de máquinas virtuais iniciais; pagl = percentual para alteração do grão de forma linear; page = percentual para alteração do grão de forma exponencial.

Cenário		Parâmetros								Cargas				
		Ts	Ti	im	mfi	mvi	pagl	page	func	Crescente	Decrescente	Constante	Oscilante	Exponencial
C4	C1	70	30	15	1	2	-	-	-	Execução 1	Execução 2	Execução 3	Execução 4	Execução 5
	C2	70	30	15	1	2	-	-	-	Execução 6	Execução 7	Execução 8	Execução 9	Execução 10
	C3	70	30	15	1	2	-	-	-	Execução 11	Execução 12	Execução 13	Execução 14	Execução 15
	C4.1	70	30	15	1	2	6	12	linear	Execução 16	Execução 17	Execução 18	Execução 19	Execução 20
		70	30	15	1	2	3	6	linear	Execução 21	Execução 22	Execução 23	Execução 24	Execução 25
		70	30	15	1	2	5	9	linear	Execução 26	Execução 27	Execução 28	Execução 29	Execução 30
		70	30	15	1	2	8	15	linear	Execução 31	Execução 32	Execução 33	Execução 34	Execução 35
	C4.2	70	30	15	1	2	6	12	exponencial	Execução 36	Execução 37	Execução 38	Execução 39	Execução 40
		70	30	15	1	2	3	6	exponencial	Execução 41	Execução 42	Execução 43	Execução 44	Execução 45
		70	30	15	1	2	5	9	exponencial	Execução 46	Execução 47	Execução 48	Execução 49	Execução 50
		70	30	15	1	2	8	15	exponencial	Execução 51	Execução 52	Execução 53	Execução 54	Execução 55
	C4.3	70	30	15	1	2	6	12	linear+exponencial	Execução 56	Execução 57	Execução 58	Execução 59	Execução 60
		70	30	15	1	2	3	6	linear+exponencial	Execução 61	Execução 62	Execução 63	Execução 64	Execução 65
		70	30	15	1	2	5	9	linear+exponencial	Execução 66	Execução 67	Execução 68	Execução 69	Execução 70
		70	30	15	1	2	8	15	linear+exponencial	Execução 71	Execução 72	Execução 73	Execução 74	Execução 75

Fonte: Elaborado pela autor.

6 RESULTADOS

Neste capítulo, busca-se apresentar a avaliação do modelo considerando a metodologia definida no capítulo anterior. Inicia-se expondo a análise para a abordagem final do algoritmo de grão adaptativo. Posteriormente, na Seção 6.2, é realizada uma análise das métricas de tempo, de energia e de custo. Já na Seção 6.3, as métricas de Speedup Elástico e Eficiência Elástica são avaliadas e na Seção 6.4 observa-se em detalhes o monitoramento dos comportamentos de cada uma das execuções realizadas, explorando o tempo, utilização de recursos e tráfego na rede. As considerações a respeito do consumo e dos perfis de alocação dos recursos e a análise do método de elasticidade não bloqueante com grão adaptativo proposto para o modelo são analisadas, apartadamente, nas últimas duas seções.

6.1 Cenários da Abordagem Final do Grão Elástico Adaptativo

A técnica do grão elástico adaptativo utiliza informações da utilização corrente dos recursos ativos para definição do tamanho do grão a ser utilizado nas operações elásticas. Foram propostas três abordagens distintas para calcular o tamanho desse grão, são elas: (i) de forma linear; (ii) de forma exponencial; (iii) de forma linear e exponencial, considerando o nível da alteração da carga. Para cada abordagem, foram realizados testes nos quais constatou-se que os melhores índices de custo, tempo, energia e eficiência são alcançados na abordagem que realiza alteração de forma linear e exponencial. À vista disso, a Tabela 7 ilustra os resultados obtidos para as quatro diferentes combinações de parâmetros aplicados nessa abordagem que obteve os melhores índices, na qual as regras para alteração do tamanho do grão – linear e exponencial – são realizadas com base no percentual dos diferentes níveis de alteração das cargas. Por exemplo: imagine um cenário onde o percentual de alteração da carga está definido, para aplicar uma função linear e exponencial, em 5% e 10%, respectivamente. Caso se detecte um aumento de 6%, ocorrerá um incremento linear no tamanho do grão, pois a faixa de 5% até 10% está configurada para essa função. Em contrapartida, se for verificado uma alteração superior a 10%, ocorrerá um incremento exponencial no tamanho desse grão.

Com o objetivo de gerar apenas uma abordagem definitiva para guiar o funcionamento da

Tabela 7 – Resultados obtidos nas execuções das quatro diferentes combinações de parâmetros da abordagem que realiza alterações no tamanho do grão de forma linear e exponencial. pagl: percentual para alteração do grão de forma linear; page: percentual para alteração do grão de forma exponencial. Os valores em verde e vermelho representam, respectivamente, o melhor e pior resultado para cada uma das métricas em cada perfil de carga.

Abordagem	Parâmetros		Crescente			Decrescente			Constante			Oscilante			Exponencial		
	pagl	page	Tempo	Energia	Custo	Tempo	Energia	Custo	Tempo	Energia	Custo	Tempo	Energia	Custo	Tempo	Energia	Custo
C4.1	6	12	2172	12330	26780760	2240	14490	32457600	2258	13860	31295880	2414	14055	33928770	2366	10230	24204180
C4.2	3	6	2066	11295	25300800	2145	13065	28024425	2170	13635	29587950	2514	12195	30658230	2261	8490	19195890
C4.3	5	9	2063	11430	25808940	2221	12990	28850790	2193	14055	30822615	2362	14145	33410490	2344	10620	24893280
C4.4	8	15	2110	11385	24022350	2207	13350	29463450	2254	13770	31037580	2492	14130	35211960	2311	10800	24958800

Fonte: Elaborado pela autor.

Tabela 8 – Análise da métrica de Custo para seleção da solução final do grão de elasticidade adaptativo. Os valores em verde e vermelho representam, respectivamente, o melhor e pior resultado para cada uma das métricas em cada perfil de carga.

Abordagem	Crescente	Decrescente	Constante	Oscilante	Exponencial	Total
C4.1	1	1	1	2	3	8
C4.2	3	4	4	4	4	19
C4.3	2	3	3	3	2	13
C4.4	4	2	2	1	1	10

Fonte: Elaborado pela autor.

técnica do grão adaptativo, realizou-se uma avaliação dos resultados através da técnica apresentada por Triantaphyllou (2000) que é conhecida por Modelo de Soma Ponderada. Para tanto, em cada um dos perfis de carga, atribuiu-se um peso para cada posição na lista, baseando-se nos valores dos custos. Assim, considerando as 4 possíveis combinações, a primeira posição na lista recebe o peso 4; a segunda recebe peso 3; a terceira peso 2; e, por fim, a quarta peso 1. Dessa maneira, pode-se observar na Tabela 8 que o C4.2 obteve o maior peso ao somar os valores obtidos para cada comportamento de carga, sendo esta abordagem a adotada como final para a técnica do grão elástico. C4.2 obteve um total de 19, através de uma soma composta por: (i) crescente: 3; (ii) decrescente: 4; (iii) constante: 4; (iv) oscilante: 4; (v) exponencial: 4.

Pode-se observar que essa solução é promissora, uma vez que obteve destaque em relação ao tempo, energia e custo nos cinco diferentes comportamentos de carga avaliadas. De forma inversa, também foi possível evidenciar que C4.1 e C4.4 obtiveram respectivamente os piores resultados, devido a importância de um incremento exponencial em momentos de uma maior variação de carga. Para essas duas abordagens as variações de carga não chegam a ultrapassar o percentual definido para atuar com incremento exponencial do grão, resultando em operações de forma linear, e como consequência levando um tempo maior para atingir uma quantidade de recursos mais adequada para as cargas de trabalho correntes.

6.2 Análise de Tempo, Energia e Custo

A Tabela 9 apresenta uma análise das métricas de Tempo, de Energia e de Custo considerando os cinco comportamentos de carga frente os quatro diferentes cenários: (i) C1: execução da aplicação com a quantidade mínima de recursos (uma máquina física com duas máquinas virtuais) e sem o uso da elasticidade; (ii) C2: execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade apenas no nível de BBU (máquina virtual) e com o grão de elasticidade estático; (iii) C3: execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade multinível, BBU Pool (máquina física) e BBU (máquina virtual) de forma individual, junto ao grão de elasticidade estático; (iv) C4: execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade multinível e a abordagem do grão de elasticidade adaptativo habilitada.

Comparando os diferentes cenários de execução, observa-se que apesar do cenário C1 atin-

Tabela 9 – Análise das métricas Tempo, Energia e Custo para os cinco comportamentos de carga frente os quatro cenários propostos: (i) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (ii) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (iii) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (iv) C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo. Os valores em verde e vermelho representam, respectivamente, o melhor e pior resultado para cada uma das métricas em cada perfil de carga.

Cenário	Crescente			Decrescente			Constante			Oscilante			Exponencial		
	Tempo	Energia	Custo	Tempo	Energia	Custo	Tempo	Energia	Custo	Tempo	Energia	Custo	Tempo	Energia	Custo
C1	5582	11190	62462580	5613	11250	63146250	5595	11220	62775900	5610	11250	63112500	5705	11430	65208150
C2	2114	11490	24289860	2258	13635	30787830	2254	13770	31037580	2415	14145	34160175	2377	10950	26028150
C3	2069	11400	23586600	2272	13965	31728480	2144	13920	29844480	2371	14190	33644490	2356	10890	25656840
C4	2066	11295	23335470	2145	13065	28024425	2170	13635	29587950	2514	12195	30658230	2261	8490	19195890

Fonte: Elaborado pela autor.

gir os melhores índices de Energia em quase todos os tipos de carga, as execuções dos cenários que atuam com a elasticidade habilitada obtiveram índices de tempo muito superiores, resultando em um melhor custo. Esse comportamento se deve ao fato de que ao executar a aplicação sem elasticidade, os recursos permanecem grande parte do tempo sobrecarregados e isso aumenta drasticamente o tempo total de execução da aplicação. Como consequência, ocorre um impacto direto no custo, que é a métrica que reflete essa relação entre tempo e energia. Ainda, destaca-se que na carga exponencial, onde o consumo de recursos inicia baixo e que em pouco tempo ocorre um grande aumento na utilização, o cenário C1 acabou obtendo um gasto energético superior em relação aos demais pois os recursos que estavam sendo alocados nos cenários de elasticidade foram utilizados quase que em sua totalidade devido a alta demanda. Esses pontos demonstram a viabilidade da elasticidade para favorecer o tempo de execução da aplicação e o custo final necessário para a execução das tarefas.

Nota-se que o cenário C3, em relação ao cenário C2, obteve custos inferiores para quatro cargas (crescente, constante, oscilante, exponencial), com uma diferença entre 1.4% e 3.8%. Esses valores ficaram bastante próximos porque C3 tem como diferencial a elasticidade multinível, onde em momentos de alta utilização da rede, realiza a alocação de uma nova máquina física no intuito de dividir o volume de trafego na interface de rede que está sob alta utilização. Dessa forma, essa abordagem tem o objetivo de melhorar o desempenho em relação ao uso da rede. Em relação ao custo, essa diferença não foi expressiva, uma vez que em cada violação do *threshold* de uso da rede foi alocado uma nova máquina física com uma máquina virtual para cada core. Porém, as máquinas utilizadas nos presentes experimentos possuem apenas 2 cores, então o impacto direto no custo não é tão expressivo. Dessa forma, acredita-se que através da utilização de recursos físicos com maior quantidade de cores, essa diferença tende a aumentar devido a maior disponibilização de recursos virtuais ao alocar um novo recurso físico.

Analisando os resultados da perspectiva do cenário C4, o qual atua com a elasticidade multinível e a abordagem do grão adaptativo, é possível evidenciar uma superioridade de custo total em relação aos cenários C1 e C2 para todos os padrões de carga. Em relação ao C1 a diferença em termos de custo é extremamente significativa, ficando esse entre 51,4% e 70,6%. Quanto aos cenários elásticos, no C4 em relação à C2, observa-se que em cargas que possuem

menores índices de variações o ganho foi menor, resultando em 3.9%, 4.67% e 9.0% para as cargas crescente, constante e decrescente, respectivamente. Já nos cenários com maiores variações de carga, observa-se um ganho expressivo, resultando em 10.3% e 26.2%, para os cenários oscilante e exponencial, respectivamente. Isso é ocorre, pois para cargas com variações mais frequentes ou acentuadas, uma adaptação do tamanho do grão elástico leva ao aumento da reatividade, ocasionando adição ou remoção de novos recursos de maneira mais rápida, e oferecendo assim valores mais competitivos quando comparados com o cenário C2 que utiliza a elasticidade com um grão de tamanho fixo, por exemplo.

6.3 Análise de Speedup e Eficiência

A Tabela 10 apresenta a comparação das métricas de Recursos (RE), Speedup Elástico (SP) e Eficiência Elástica (EF). Conforme citado na Seção 5.1.1, o cálculo de RE e SP são utilizados como parâmetros para calcular EF. Salienta-se que no cenário C1 todos os resultados de SE e EE resultam em 1 porque o cálculo considera a relação entre o cenário C1 com o próprio cenário, o que sempre vai resultar em 1.

Da perspectiva da métrica SP, os melhores resultados são observados na carga Crescente, através dos cenários C3 e C4. Em contrapartida, os piores resultados foram encontrados na carga oscilante para os cenários C4 e C2, respectivamente. De posse desses dados, também é possível observar uma relação direta entre as métricas de Speedup Elástico e Tempo, apresentado na Tabela 9, na qual os melhores e piores valores de tempos e de Speedup ocorrem nas mesmas execuções.

Do ponto de vista da métrica EF, os melhores resultados foram na carga Exponencial, através dos cenários C4 e C3, respectivamente, enquanto os piores resultados ocorreram na carga oscilante, nos cenários C2 e C3. Nessa perspectiva, também existe uma relação direta da métrica EF com a Energia, onde as melhores e piores execuções de EF também ocorrem em relação a métrica de Energia. Ou seja, os resultados demonstram que um melhor desempenho equivale a uma maior eficiência, e em contrapartida, um pior desempenho leva à uma menor eficiência.

6.4 Histórico de Comportamento de cargas e alocação de recursos

Para facilitar a compreensão e evidenciar as relações entre as diferentes métricas coletadas, além da Tabela 9, elaborou-se as Tabelas 11, 12, 13, 14 e 15. Além disso, as Figuras 24, 25, 26, 27 e 28 apresentam em detalhes a execução dos comportamentos de cada um dos cenários frente os diferentes tipos de cargas propostos nos experimentos. Todas as Figuras foram geradas utilizando eixos da mesma escala, dessa forma possibilita uma comparação vertical entre cada uma das execuções.

Tabela 10 – Análise das métricas de Recursos (RE), Speedup Elástico (SP) e Eficiência Elástica (EF) nos cinco comportamentos de cargas, para os quatro cenários:(i) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (ii) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (iii) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (iv) C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo. Os valores em verde e vermelho representam, respectivamente, o melhor e pior resultado para cada uma das métricas em cada perfil de carga.

Cenário	Crescente			Decrescente			Constante			Oscilante			Exponencial		
	RE	SP	EF	RE	SP	EF	RE	SP	EF	RE	SP	EF	RE	SP	EF
C1	2,00	1,00	1,00	2,00	1,00	1,00	2,01	1,00	1,00	2,01	1,00	1,00	2,00	1,00	1,00
C2	5,44	2,64	0,97	6,04	2,49	0,82	6,11	2,48	0,81	5,86	2,32	0,79	4,61	2,40	1,04
C3	5,51	2,70	0,98	6,15	2,47	0,80	6,49	2,61	0,80	5,98	2,37	0,79	4,62	2,42	1,05
C4	5,47	2,70	0,99	6,09	2,62	0,86	6,28	2,58	0,82	4,85	2,23	0,92	3,75	2,52	1,34

Fonte: Elaborado pela autor.

Tabela 11 – Comparativo entre os comportamentos da carga crescente.

Média das Métricas	Cenários			
	C1	C2	C3	C4
Tempo de Resposta (ms)	381,89	120,13	118,45	119,11
Vazão (Kbytes/s)	1233,65	3597,61	3730,88	3685,40
Máquinas Virtuais	2,00	6,13	6,28	6,22
Máquinas Físicas	1,00	3,30	3,32	3,30

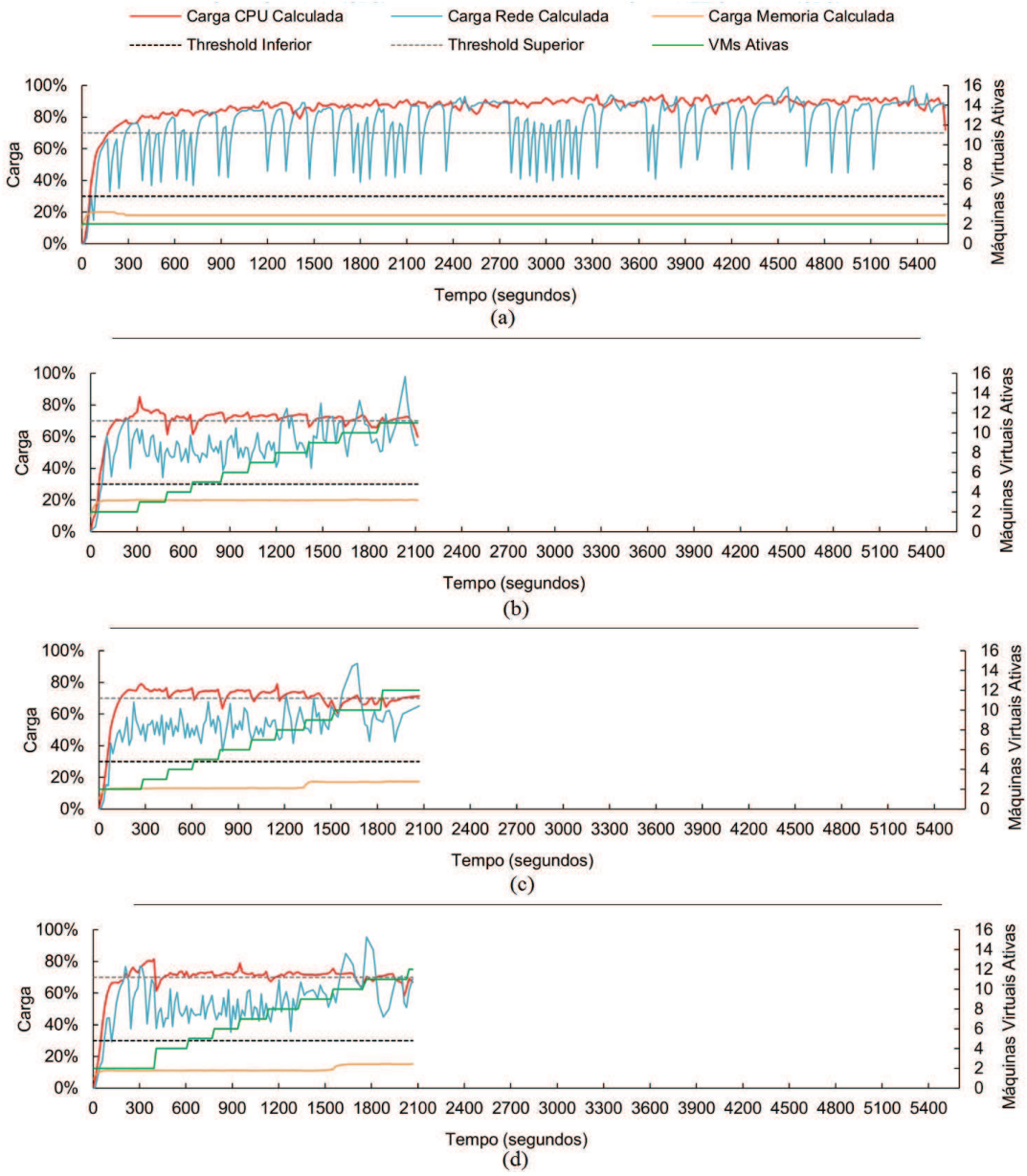
Fonte: Elaborado pela autor.

6.4.1 Comportamento de Carga Crescente

A carga crescente, tem o objetivo de avaliar o comportamento do modelo em situações de crescimento constante na demanda de recursos computacionais. Observa-se na Figura 24 que para esse tipo de carga o *threshold* inferior acaba não tendo nenhum impacto durante as execuções, uma vez que o mesmo nunca é violado por tratar-se de uma carga com tendência de crescimento. Isso resulta em violações frequentes apenas da *threshold* superior. No cenário C1, fica claro que no início da execução, o volume de tráfego na rede e a utilização de CPU são elevados, fazendo com que os mesmos violem a *threshold* superior durante praticamente toda a execução.

Diferentemente do cenário C1, que não atua com elasticidade, os demais cenários exploram esse recurso, o que possibilita a alteração de recursos ativos para evitar violações das *thresholds* definidas. Isso leva a uma diferença bastante expressiva em termos de tempo de execução, com uma redução do tempo total na execução de aproximadamente 63% e do tempo de resposta de 69% dos cenários C2, C3 e C4 em relação à C1. Agora considerando os cenários de elasticidade entre si, observa-se que C2 e C3 realizaram processamentos bastante semelhas, onde o principal diferencial entre eles é dado no tempo de 1574s, onde C3, com sua abordagem de elasticidade multinível, detecta uma violação na *threshold* superior de rede, ocasionando na alocação de uma nova máquina física com duas máquinas virtuais, e resultando em uma maior redução no volume de tráfego da rede e utilização de CPU, uma vez que as tarefas a serem processadas são

Figura 24 – Carga de processamento do sistema e disponibilidade de recursos nas execuções do comportamento da carga crescente. (a) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (b) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (c) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (d) C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo.



Fonte: Elaborado pelo autor.

Tabela 12 – Comparativo entre os comportamentos da carga decrescente.

Média das Métricas	Cenários			
	C1	C2	C3	C4
Tempo de Resposta (ms)	364,77	129,42	122,99	117,49
Vazão (Kbytes/s)	1228,69	2944,59	3346,80	3499,38
Máquinas Virtuais	2,00	6,73	7,05	7,44
Máquinas Físicas	1,00	3,54	3,70	3,89

Fonte: Elaborado pela autor.

divididas entre mais BBUs.

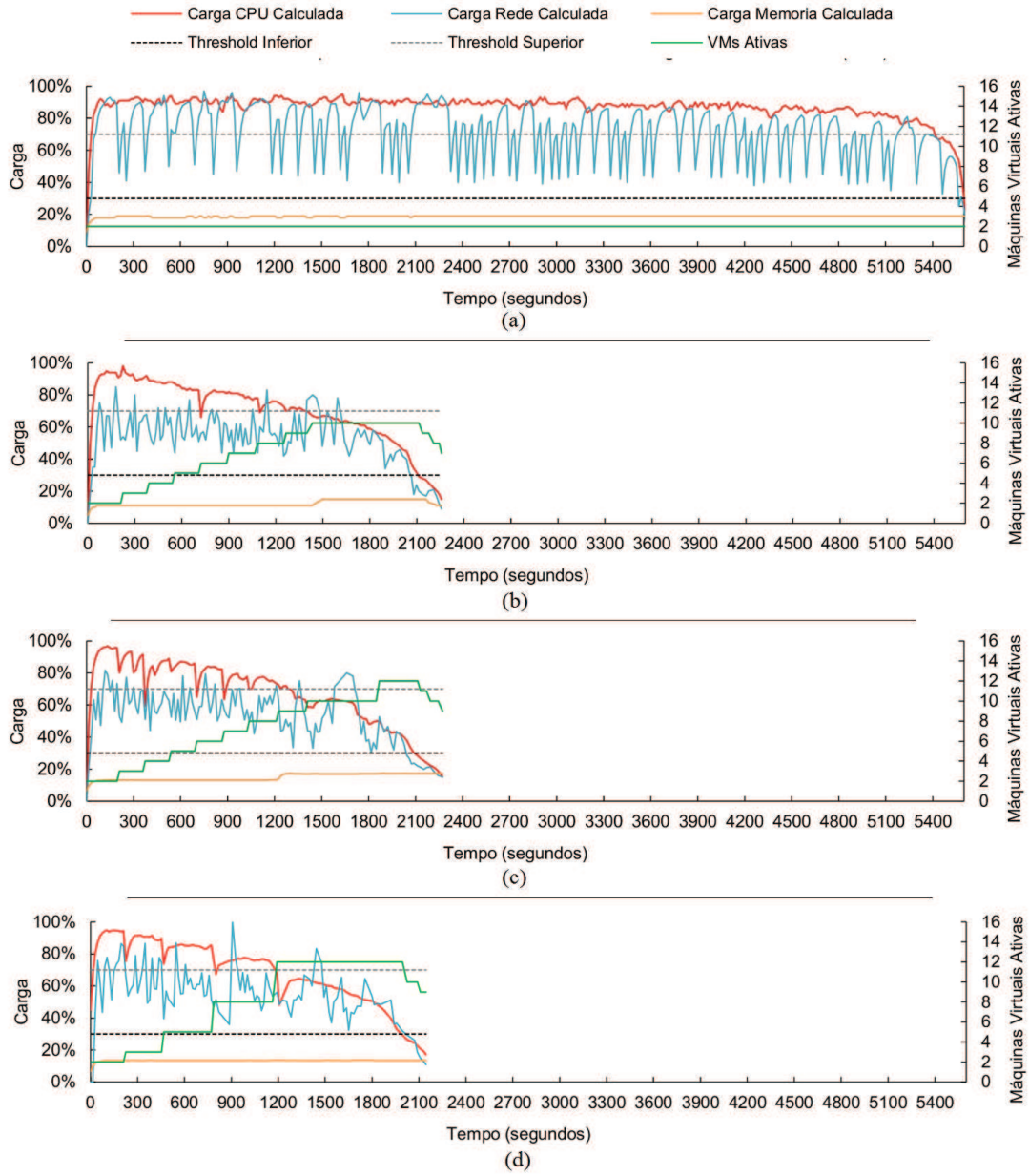
Já no cenário C4, logo no inicio da execução, aos 195s, observou-se um aumento de 4% na carga em relação ao monitoramento anterior, levando a uma adaptação no tamanho do grão de forma linear para alocar diretamente 2 máquinas virtuais, diferentemente de C2, que sempre atua de uma em uma máquina virtual. Após essa ação, o aumento da carga crescente não foi expressivo, fazendo com que o sistema fosse reagindo com um grão de tamanho de apenas 1 máquina virtual. Ainda, evidencia-se que para o uso da rede nos cenários C3 e C4, ambos obtiveram menores quantidade de violações da *threshold* superior, uma vez que C2 violou 9 vezes, enquanto C3 e C4 violaram, respectivamente, 3 e 4 vezes cada u, representando uma redução de mais de 50%.

6.4.2 Comportamento de Carga Decrescente

A carga decrescente, tem o objetivo de avaliar o comportamento do sistema em situações de queda constante da demanda por recursos computacionais. O resultado esperado inicial é uma alocação repentina de várias máquinas virtuais. Em seguida, com a queda da demanda de requisições, é esperado que a demanda diminua, permitindo a liberação gradual de recursos. Na Figura 25 pode-se observar que diferentemente da carga Crescente, nessa carga o *threshold* inferior também acaba tendo impacto durante as execuções, uma vez que ambos os *threshold*, superior e inferior, são violados durante a execução. No cenário C1 observa-se que nos primeiros segundos da execução, o volume de tráfego na rede e utilização de CPU atingem quase os 100% de uso, fazendo com que os mesmos violem a *threshold* superior durante praticamente todo o experimento. Apenas aos 5418s, quando a carga fica baixa, a utilização da rede e CPU ficam abaixo da *threshold* superior, e na sequencia, aos 5598s, violam a *threshold* inferior, onde a cargas de trabalho terminam 15s depois.

Os cenários C2 e C3 tiveram execuções bastante semelhantes do início até os 1582s, uma vez que a utilização de CPU, no geral manteve-se superior a utilização da rede, e em C3 nas janelas de detecções de violações das *thresholds*, a métrica de CPU infringiu a *threshold*, fazendo com que as alocações de elasticidade atuassem apenas em nível de máquinas virtuais. Porém após o segundo 1582 da execução, detectou-se um incremento de uso da rede que passou a utilizar 71%, dessa forma, C3 atuou no nível de máquinas físicas, alocando um novo BBU Pool (máquina

Figura 25 – Carga de processamento do sistema e disponibilidade de recursos nas execuções do comportamento da carga decrescente. (a) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (b) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (c) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (d) C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo.



Fonte: Elaborado pelo autor.

Tabela 13 – Comparativo entre os comportamentos da carga constante.

Média das Métricas	Cenários			
	C1	C2	C3	C4
Tempo de Resposta (ms)	368,09	132,57	130,43	125,09
Vazão (Kbytes/s)	1226,03	3156,99	3287,62	3458,44
Máquinas Virtuais	2,00	6,29	6,63	6,78
Máquinas Físicas	1,00	3,26	3,64	3,67

Fonte: Elaborado pela autor.

física) com dois BBUs (máquinas virtuais), no intuito de dividir o fluxo de *streaming* na rede para uma quantidade maior de máquinas. Posteriormente as cargas continuaram diminuindo, fazendo com que a quantidade de recursos diminuísse de maneira gradual.

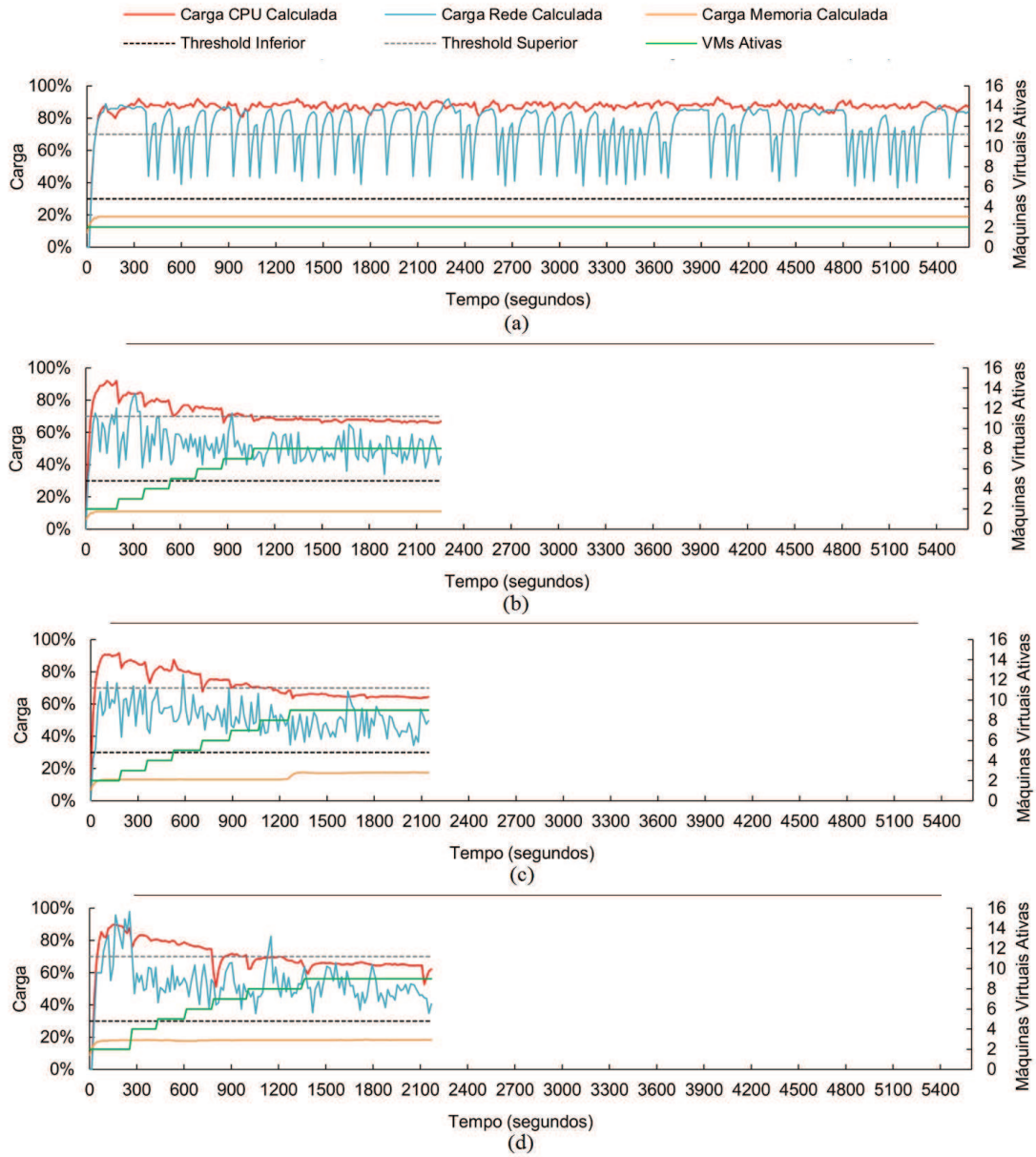
Em C4 obteve-se os melhores resultados para esse perfil de carga, pois diferentemente de C2 e C3, atua com um grão elástico adaptativo, assim foi possível responder com maior reatividade frente as variações de carga. Dessa forma, enquanto C2 e C3 conseguiram atingir condições de uso dentro das *thresholds* superior e inferior no tempo de 1434s e 1317s, respectivamente, C4 atingiu essa região aos 1192s. Isso expressa que C4 teve maior reatividade para o tratamento da carga, uma vez que atingiu a zona entre os *thresholds* 17% e 10% mais rápido do que C2 e C3, respectivamente. Ainda, com base na Tabela 12, observa-se que C4 foi o cenário que obteve a maior vazão de dados, atingindo uma média de 3499,38Kb/s, contra 2944,59Kb/s e 3346,80Kb/s de C2 e C3, tornando-o o cenário com menor tempo médio de resposta (117,49ms).

6.4.3 Comportamento de Carga Constante

Com esse perfil de carga o objetivo é avaliar a estabilidade do sistema com uma carga alta durante um longo período de tempo. Como resultado, observou-se uma sobrecarga inicial, devido ao alto volume da carga inicial. Em seguida, ocorre a alocação de máquinas virtuais que se mantem em uso até o final do experimento. Na Figura 26 pode-se observar que assim como ocorre na carga Crescente, nesse perfil de carga o *threshold* inferior acaba não tendo impacto nas execuções, uma vez que o mesmo nunca é violado pois trata-se de uma carga com tendência uniforme, na qual a partir da correta alocação da quantidade de recursos para a carga que está sendo processada, esses permanecem atuando até o final da execução. O Cenário C1 permanece processando as cargas com seus recursos em uma média de 90% de uso, o que contribui para atingir um alto tempo total de 5595s para realizar todo o processamento das requisições.

Por tratar-se de uma carga uniforme, a quantidade de tráfego na rede não gerou um grande impacto na execução dos cenários C2, C3 e C4, que de maneira geral obtiveram maiores violação em relação ao processamento de CPU. O tamanho do grão se manteve do mesmo tamanho que C2 e C3 (1 máquina virtual) durante toda a execução pois não atingiu notórias alterações entre os ciclos de monitoramento que superassem a barreira de 3%. Para essa bateria de testes,

Figura 26 – Carga de processamento do sistema e disponibilidade de recursos nas execuções do comportamento da carga constante. (a) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (b) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (c) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (d) C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo.



Fonte: Elaborado pelo autor.

Tabela 14 – Comparativo entre os comportamentos da carga oscilante.

Média das Métricas	Cenários			
	C1	C2	C3	C4
Tempo de Resposta (ms)	357,97	134,49	142,29	131,12
Vazão (Kbytes/s)	1224,72	2911,42	2772,92	3391,28
Máquinas Virtuais	2,00	6,25	6,18	6,02
Máquinas Físicas	1,00	3,32	3,31	3,22

Fonte: Elaborado pela autor.

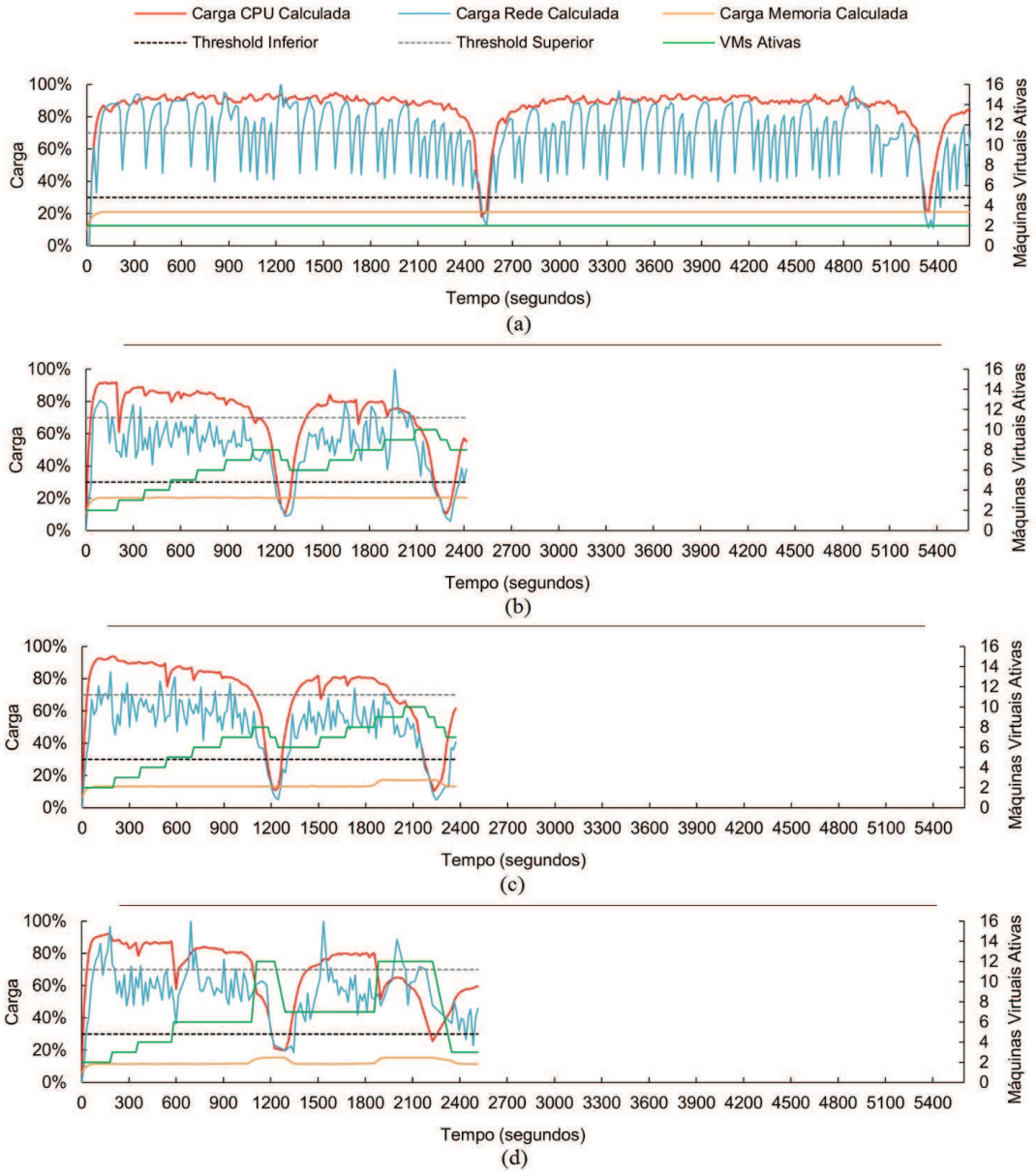
a maior diferença de C4 para C2 e C3, ocorreu aos 75s, com a única violação de tráfego de dados, onde a utilização de rede atingiu 74%, fazendo com que o Orquestrador de Elasticidade alocasse uma nova máquina física, com duas máquinas virtuais. Assim, ao final da execução C4 e C3 completaram o processamento com 9 máquinas virtuais, enquanto C2 finalizou com 8 máquinas virtuais. Conforme a Tabela 13, proporcionalmente, C3 e C4, atingiram maior vazão de dados, 3287,62Kb/s e 3458,44Kb/s, em relação ao C2 que obteve 3156,99Kb/s. Isso influenciou nos tempos médios de respostas de 132,57ms, 130,23ms e 125,09ms para C2, C3 e C4, respectivamente. Destaca-se a relação próxima entre o tempo de resposta e a vazão de dados, onde é possível verificar que maiores médias de vazões de dados contribuem para menores médias no tempo de resposta.

6.4.4 Comportamento de Carga Oscilante

Nesse perfil de carga, a ideia é avaliar o aumento e diminuição da utilização de recursos computacionais conforme a variação da carga no tempo, ou seja, simulando uma execução inconstante e irregular. Conforme observado na Figura 26 assim como ocorre na carga Decrescente, nesse perfil as *thresholds* inferior e superior possuem uma grande influência nas execuções. Como pode-se observar no cenário C1, a sua execução expressa que em determinados momentos as cargas do sistema (CPU e rede) atingem, durante um período curto de tempo, índices inferiores à 25%, como no instante 2505s, em que esse índice é de 23% no uso de rede e 18% no uso de CPU. Porém, durante a maior parte do tempo de execução a média do índice da carga do sistema fica acima da *threshold* superior, sofrendo quedas apenas nos pontos em que a quantidade e tamanho de tarefas atinge a quantidade mínima, diminuindo a carga total do sistema.

Para tal perfil de carga, C2 e C3 reagiram de forma bastante semelhante, sempre atuando com um grão de 1 máquina virtual. Ainda, destaca-se o fato de C2 e C3, processarem suas tarefas respectivamente em 2415s e 2371s, enquanto C4 levou 2514s. Isso faz com que eles tenham processado 4% e 5% mais rápido que C4. Porém, conforme observado na Tabela 9, C4 tem o melhor custo, devido a maior reatividade, ou seja, devido as diferentes variações da carga durante a execução C4 alocou e removeu recursos com maior agilidade e em menor tempo, respondendo mais rapidamente às variações de carga. Isso é evidenciado nos instantes

Figura 27 – Carga de processamento do sistema e disponibilidade de recursos nas execuções do comportamento da carga oscilante. (a) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (b) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (c) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (d) C4: execução da aplicação com elasticidade em multinível e o grão de elasticidade adaptativo.



Fonte: Elaborado pelo autor.

1111s e 1883s, onde realizou-se um aumento exponencial no tamanho do grão, incrementando 5 novos recursos em uma mesma operação elástica. Em contrapartida, nos tempos 1291s e 1883s, realizou-se dois aumentos exponenciais no tamanho do grão para duas operações de decremento, diminuindo na primeira operação 5 e na segunda 9 máquinas virtuais.

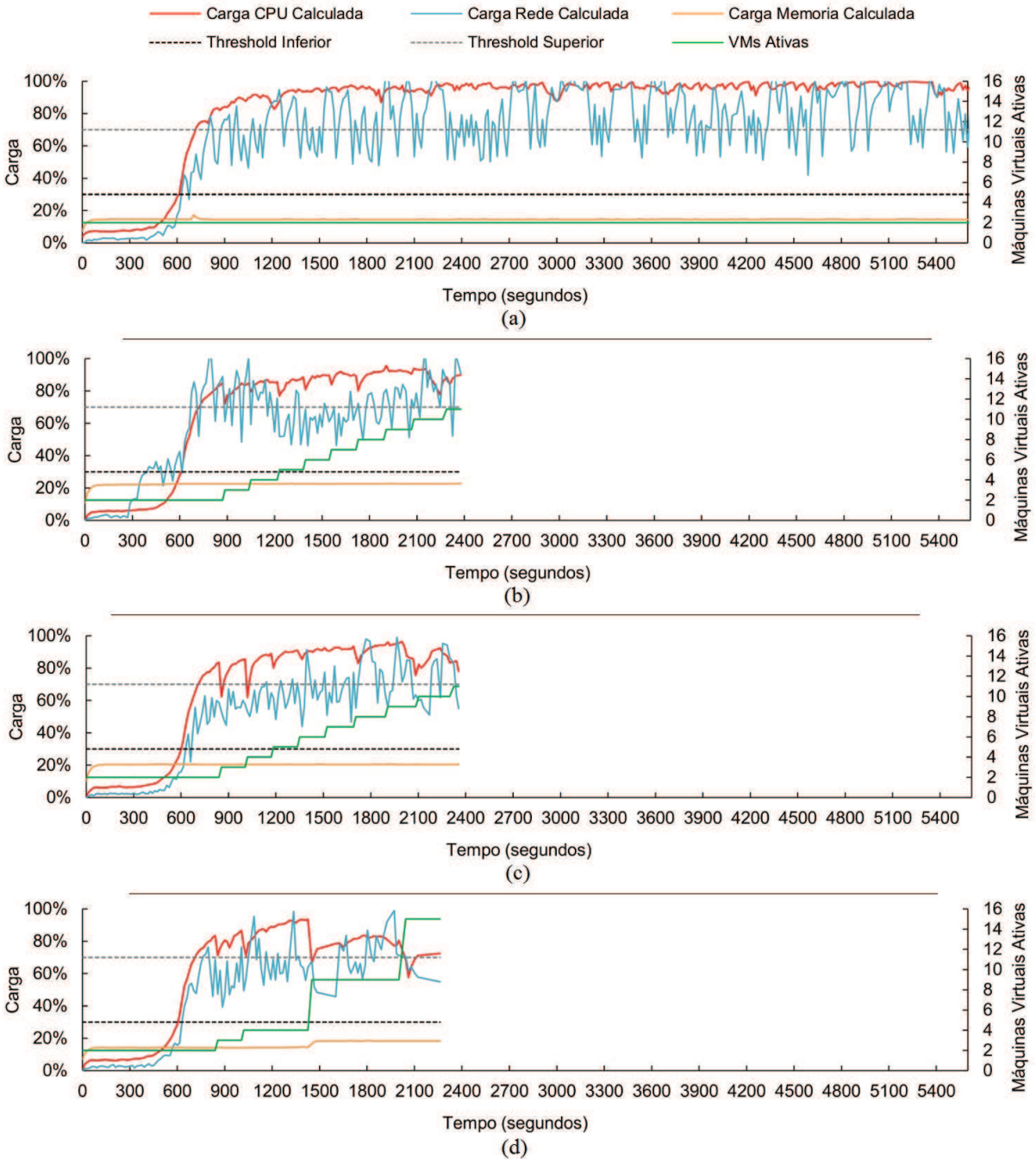
A Tabela 14 mostra que a média de recursos de C4 foi inferior à C2 e C3, e mesmo assim obteve um tempo médio de resposta inferior com 131,12ms contra 134,49ms de C2 e 142,29ms de C3. Essa diferença ocorre, pois, C2 e C3 realizaram 14 e 15 operações de elasticidade, respectivamente, enquanto C4 realizou apenas a metade disso, com 7 operações. Apesar das operações de elasticidade serem executadas de forma não bloqueante para todos os cenários de elasticidade, ou seja, o Orquestrador de Elasticidade instancia os novos recursos e só notifica o Orquestrador do Pool no momento em que eles estiverem prontos para execução, ainda existe um tempo gasto pelo Orquestrador do Pool para mapear, conectar e passar a utilizar esses novos recursos. Para tal é necessário que o Orquestrador do Pool pare a distribuição de tarefas para verificar na região de dados compartilhados a existência de novos recursos, bem como organiza-los para iniciarem processamentos de novas tarefas. Ou seja, para esse perfil de carga, no cenário C4, além do aumento da reatividade, ocorre ganho de mais de 50% de redução da quantidade de operações elásticas realizadas nos cenários de elasticidade. Além disso, a redução da quantidade de operações elásticas gera ganhos de tempo e economia de energia por não precisar esperar tantas vezes pelo período de *cooldown*, o qual representa uma parada de dois ciclos de monitoramento para o sistema atingir um estado global consistente antes de poder efetuar uma nova ação elástica.

6.4.5 Comportamento de Carga Exponencial

Por fim, o perfil de carga Exponencial tem por objetivo avaliar o comportamento do sistema em situações de alteração bastante agressiva por demanda de recursos computacionais. Conforme observado na Figura 28, assim como ocorre na carga Crescente, nesse perfil de carga apenas a *threshold* superior possuem influência nas execuções. Pode-se observar que até os 615s a carga permanece abaixo da *threshold* inferior, porém não são realizadas ações elásticas devido as restrições do modelo de quantidade mínima de recursos para o funcionamento (1 máquina física com duas máquinas virtuais). Assim, a partir de 616s, a carga começa a crescer de forma exponencial, mantendo-se dentro das *thresholds* por apenas 105s, onde ocorre a violação da *threshold* superior para todos os cenários. O cenário C1 permanece com sobrecarga de CPU e rede durante toda a execução, levando-o ao pior tempo entre todos os experimentos, com um total de 5705s e um tempo médio de resposta (1205,34Kb/s) quase três vezes maior que os cenários C2, C3 e C4, que levaram respectivamente 3149,38Kb/s, 3298,00Kb/s e 3420,98Kb/s.

Assim como no comportamento oscilante, para esse perfil de carga, C2 e C3 reagiram de forma bastante semelhante, reagindo às variações de carga sempre com o grão fixo de 1 máquina virtual. Isso pois não foi detectada violações no nível de máquinas físicas, pois o uso de CPU

Figura 28 – Carga de processamento do sistema e disponibilidade de recursos nas execuções do comportamento da carga exponencial. (a) C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (b) C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (c) C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (d) C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo.



Fonte: Elaborado pelo autor.

Tabela 15 – Comparativo entre os comportamentos da carga exponencial.

Média das Métricas	Cenários			
	C1	C2	C3	C4
Tempo de Resposta (ms)	367,03	137,80	132,56	129,23
Vazão (Kbytes/s)	1205,34	3149,38	3298,00	3420,98
Máquinas Virtuais	2,00	4,83	4,87	4,53
Máquinas Físicas	1,00	2,58	2,60	2,43

Fonte: Elaborado pela autor.

sempre se manteve infringindo a *threshold* superior enquanto o uso da rede realizava algumas oscilações entre ultrapassar a *threshold* superior e manter-se dentro de uma região entre as *thresholds* superior e inferior.

Novamente C4 obteve destaque por utilizar a abordagem de grão adaptativo, aumentando a quantidade de recursos com maior velocidade, para assim, responder mais rapidamente às mudanças de carga. À vista disso, C4 processou em um tempo total 60% menor que C1, 5% menor que C2 e 4% menor que C3. Além disso, conforme apresentado na Tabela 9, obteve o menor gasto de energia e melhor custo entre todas as execuções. Outrossim, foi o cenário que atingiu a maior quantidade de máquinas físicas (8) e virtuais (15) alocadas entre todos os cenários, porém devido ao curto tempo que precisou utiliza-las, obteve uma média de máquinas físicas e virtuais inferior à C2 e C3, conforme observado na Tabela 15. Isso deve-se ao fato de que as máquinas que foram alocadas mais rapidamente, foram utilizadas quase que em sua totalidade. Isso contribui para ressaltar os benefícios da utilização da abordagem com o grão elástico adaptativo no intuito de aumentar a reatividade no sistema frente à abordagem de elasticidade tradicional.

6.5 Análise de Entrega e Consumo de Recursos

A Figura 29 demonstra a avaliação da métrica Energia obtida em todos os cenários testados: (i) C1: execução da aplicação com a quantidade mínima de recursos (uma máquina física com duas máquinas virtuais) e sem o uso da elasticidade; (ii) C2: execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (iii) C3: execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade multinível, para BBU Pool e BBUs de forma individual, além de utilizar um grão de elasticidade estático; e (iv) C4: execução da aplicação iniciando com a quantidade mínima de recursos, com elasticidade multinível e com a técnica do grão de elasticidade adaptativo habilitada. Para cada cenário, foram considerados todos os comportamentos de carga testados. Conforme apresentado na seção 5.1.2, a métrica de Energia representa a quantidade total de recursos que foram alocados durante a execução da aplicação. Dessa forma, pode-se observar a quantidade de energia total alocada e a quantidade de energia que efetivamente consumida. Por exemplo, se no monitoramento n está alocado 10 máquinas virtuais, com 50% de carga de CPU, o consumo desses recursos alocados equivale à 5 máquinas virtuais para

o ciclo n do monitoramento. Assim como nas Figuras de comportamento das cargas, na Figura 29, cada um dos comportamentos de carga foi organizado de maneira a utilizar o eixo x com a mesma escala, facilitando assim a comparação vertical entre eles.

Observando-se, então, a Figura citada, verifica-se que o cenário C1 obteve, para quatro dos cincos perfis de carga, os menores valores de energia alocada e os maiores percentuais de utilização dos recursos alocados, superando a marca média de 85% de utilização. Isso ocorre porque C1 sempre processa com a quantidade mínima de recursos (1 máquina física e 2 máquinas virtuais) do início ao fim das execuções. Também é possível notar que há proporcionalidade entre Tempo e Energia, pois, para executar as requisições mais rapidamente, é necessária uma quantidade maior de recursos, aumentando a energia alocada. Dessa forma, ao considerar o desempenho, é possível concluir que priorizar uma economia na alocação dos recursos resulta em penalidades no tempo de execução e entrega das requisições.

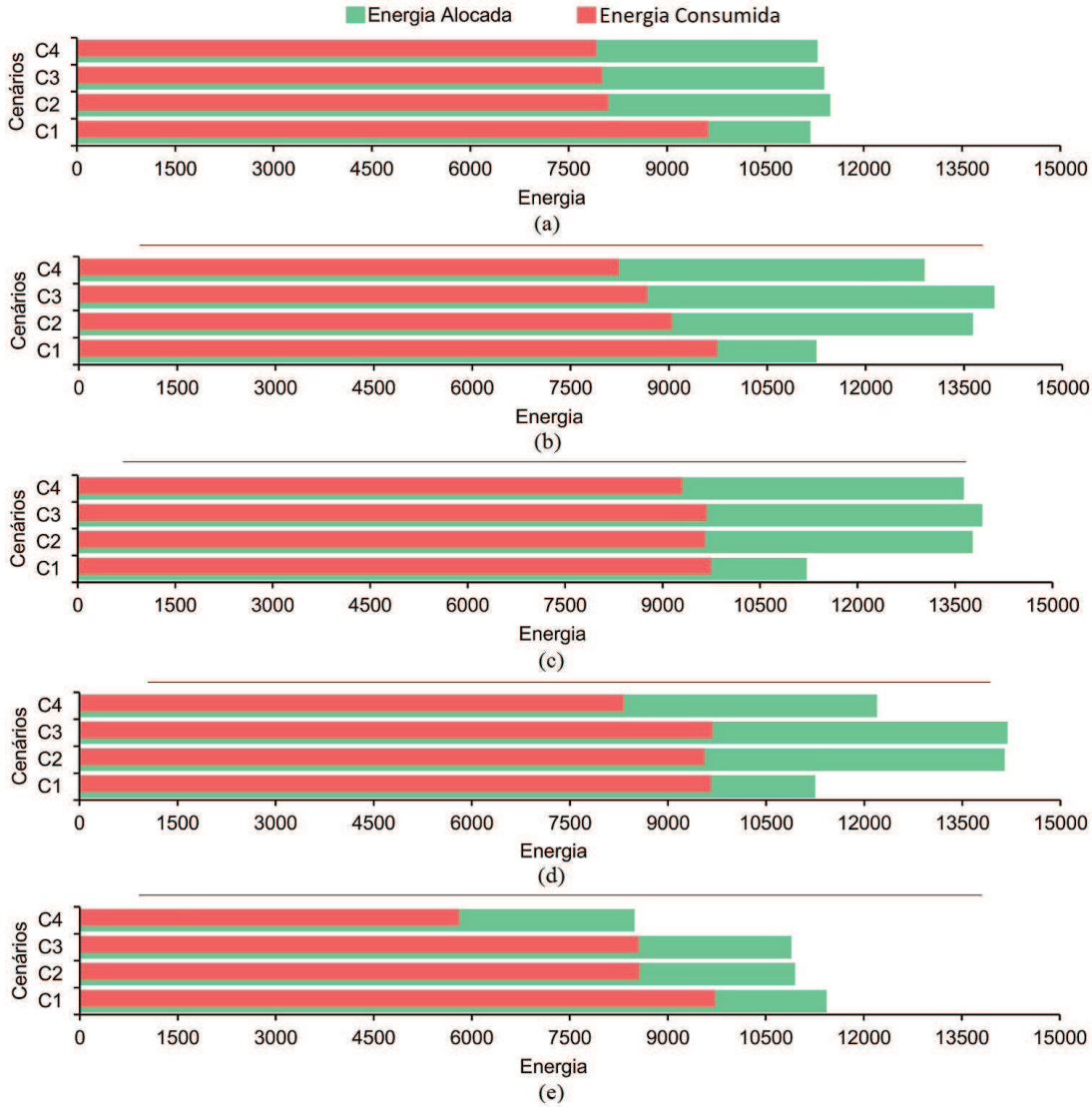
Em relação aos cenários de elasticidade, o C4 obteve os menores valor de Energia alocada, quando comparado com os demais, mesmo tendo valores médios de quantidade de máquinas físicas e virtuais superiores para cenários como Decrescente e Constante. Tal circunstância se deve ao fato de que tal cenário explora o recurso de grão adaptativo, onde as ações elásticas visam responder mais rapidamente às mudanças de carga, ou seja, ao identificar um crescimento da carga superior à 3% e inferior a 6%, ao invés de utilizar um grão fixo de 1 máquina virtual, foi realizado um incremento linear no grão. Em contrapartida, ao detectar variações maiores, superiores à 6%, foi realizado um incremento exponencial no tamanho do grão. Na carga Exponencial, isso é extremamente notório pois o grão elástico cresce duas vezes de forma linear e duas vezes de forma exponencial, ou seja, é possível atingir um total de 15 máquinas virtuais em um tempo inferior à C2, que em um tempo superior conseguiu atingir apenas 11 máquinas.

6.6 Análise do mecanismo de elasticidade com utilização de grão elástico

No escopo de analisar o impacto nos custos das ações elásticas que utilizam o grão adaptativo, formulou-se a Tabela 16. Assim, utilizando-se dos custos coletados pelas informações obtidas nessa tabela, selecionou-se, para os mesmos perfis de carga, a melhor e a pior execução de C4 (carga exponencial e oscilante) e comparou-se com a melhor e pior execução de C2. Como resultado, observou-se que no momento em que o Orquestrador de Elasticidade identifica a violação de *threshold* superior, para efetivamente entregar novos recursos para processamento no Orquestrador do Pool, um período de tempo é necessário para que esses novos recursos sejam instanciados, inicializados e comecem a se comunicar na rede. Assim, a Tabela 16 apresenta o período de tempo em que os recursos são instanciados e o momento em que eles são liberados e comunicados para o Orquestrador do Pool iniciar a distribuição de requisições para processamento.

Notou-se que o perfil de carga oscilante é o que mais necessita de operações de elasticidade durante toda a execução devido ao seu comportamento altamente variante. Observou-se, ade-

Figura 29 – Relação entre energia alocada e consumida para os cenários: C1: execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; C2: execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; C3: execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; C4: execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo. (a) perfil de carga crescente; (b) perfil de carga decrescente; (c) perfil de carga constante; (d) perfil de carga oscilante; (e) perfil de carga exponencial.



Fonte: Elaborado pelo autor.

Tabela 16 – Operações de elasticidade para o melhor e pior custo do cenário C4, juntamente com o melhor e pior custo do cenário C2 nos mesmos perfis de carga.

Carga	Cenário	Número	Detecção		Entrega		Tamanho Grão	OP	Tempo Total
			Monitoramento	Tempo	Monitoramento	Tempo			
Oscilante	2	1	6	75	15	210	1	+	135
		2	17	240	26	376	1	+	136
		3	28	406	37	542	1	+	136
		4	39	572	48	709	1	+	137
		5	50	739	60	890	1	+	151
		6	62	920	71	1061	1	+	141
		7	82	1226	83	1241	1	-	15
		8	85	1278	86	1295	1	-	17
		9	93	1400	102	1597	1	+	197
		10	104	1579	112	1714	1	+	135
		11	114	1746	123	1897	1	+	151
		12	125	1927	133	2093	1	+	166
		13	141	2223	142	2252	1	-	29
		14	144	2289	145	2313	1	-	24
	4	1	6	75	14	195	1	+	120
		2	16	225	25	360	1	+	135
		3	27	390	40	585	2	+	195
		4	42	615	70	1111	6	+	496
		5	75	1224	76	1231	5	-	7
		6	85	1435	110	1883	5	+	448
		7	123	2226	124	2348	9	-	122
Exponencial	2	1	49	720	60	885	1	+	165
		2	62	915	71	1050	1	+	135
		3	73	1080	83	1230	1	+	150
		4	85	1260	94	1395	1	+	135
		5	96	1425	105	1560	1	+	135
		6	107	1590	116	1728	1	+	138
		7	118	1758	127	1906	1	+	148
		8	129	1940	137	2080	1	+	140
		9	139	2118	147	2288	1	+	170
	4	1	49	720	58	855	1	+	135
		2	60	885	69	1020	1	+	135
		3	71	1050	95	1450	5	+	400
		4	97	1483	120	2044	6	+	561

Fonte: Elaborado pela autor.

mais, que, quando comparado com cenário de elasticidade tradicional, a abordagem que utiliza grão adaptativo precisou realizar apenas 50% da quantidade de operações de elasticidade para a carga oscilante e 45% para a carga exponencial. Isso ocorre pois, enquanto em C2 o grão de elasticidade possui um tamanho fixo, com apenas uma máquina virtual, em C4 ocorrem momentos em que as operações são realizadas com o grão de tamanho de uma, duas, cinco, seis ou até nove máquinas, o que faz com que C4 apresente uma reatividade maior, respondendo mais rapidamente as variações de carga.

Veja-se que, para cada nova máquina virtual solicitada, existe um período de tempo para a transferência da imagem da máquina virtual para a máquina física, bem como para a inicialização do sistema operacional e preparação para comunicação na rede. Apenas após as novas máquinas virtuais estarem aptas a se comunicar na rede é que elas são efetivamente liberadas para uso. Essa circunstância evita a parada do Orquestrador do Pool durante o processo de inicialização e tudo ocorre através da estratégia de elasticidade que ocorre de maneira não bloqueante. Além disso, observa-se que conforme o tamanho do grão elástico aumenta, o middleware

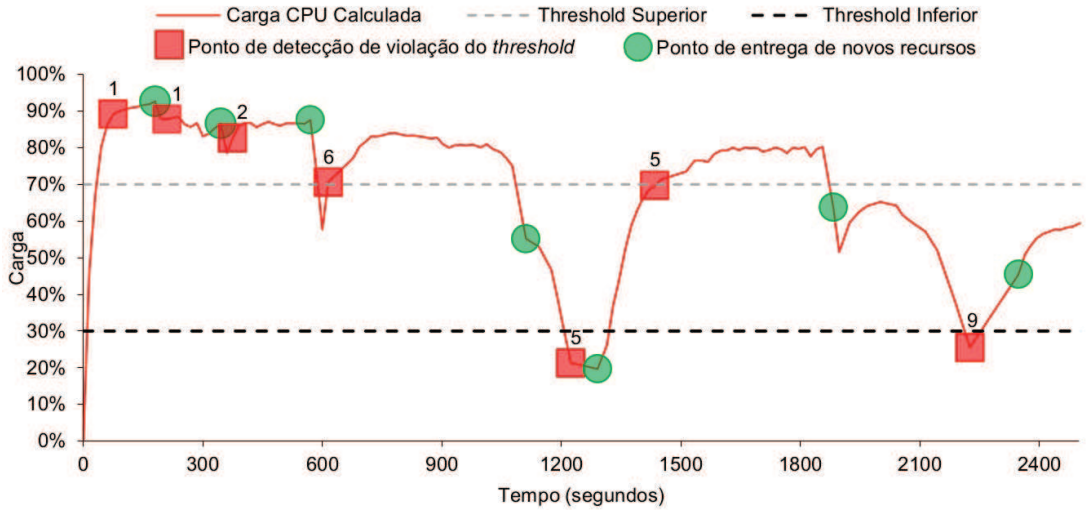
do OpenNebula leva um tempo maior para entregar esses novos recursos. Para rotinas de decremento de recursos o tempo é bastante baixo, de modo que o Orquestrador de Elasticidade solicita a redução de recursos para o Orquestrador do Pool, o qual finaliza os processamentos da quantidade de máquinas a serem reduzidas e notifica o Orquestrador de Elasticidade da liberação de recursos.

Também pode ser observado que conforme o tamanho do grão aumenta, o *middleware* do OpenNebula leva um tempo maior para entregar esses novos recursos. Tomando como exemplo o grão de tamanho 1 e o de tamanho 5, veremos que, enquanto o tempo de solicitação e entrega de um novo recurso leva em média 125s, a entregar cinco novos recursos leva em média 424s. Então, se compararmos o tempo para C2 chegar em 5 máquinas virtuais, tal levaria 625s contra 424s de C4, o que representa um ganho de 32%. Acredita-se que esse valor pode ser ainda superior ao explorar um *middleware* de nuvem capaz de realizar todo o processo de instância e de preparação das máquinas virtuais de forma paralela. Para rotinas de decremento de recursos o tempo é bastante baixo, de modo que o Orquestrador de Elasticidade solicita a redução de recursos para o Orquestrador do Pool, o qual finaliza os processamentos da quantidade de máquinas a serem reduzidas e autoriza o Orquestrador de Elasticidade para remover os recursos.

Outrossim, a cada operação realizada, ocorre o período de *cooldown*, que se refere ao período de tempo após a realização de uma operação, em que novas operações de elasticidade não são permitidas (JAMSHIDI; AHMAD; PAHL, 2014). Isso faz com que leve um tempo ainda maior para C2 atingir a carga de recursos necessária para os processamentos correntes, pois em cada ação elástica ele tem esse tempo de *cooldown* equivalente a dois ciclos de monitoramento para poder iniciar uma nova ação de elasticidade.

Buscando analisar com maior profundidade o impacto das operações de elasticidade com grão adaptativo, a Figura 30 apresenta em maiores detalhes a execução da carga Oscilante de C4, na qual ocorreu a maior quantidade de operações elásticas, que, como visto anteriormente, utiliza elasticidade multinível junto ao grão de elasticidade adaptativo. Observa-se que no início a detecção de variação de carga é menor que 3%, mantendo o grão elástico em uma máquina virtual. Isso ocorre até os 390s, onde se observa uma variação de 4%, resultando em um incremento linear do grão que passa a ter tamanho de duas máquinas virtuais e tem recursos entregues aos 585s. No ponto 615, verifica-se uma variação superior à 6%, o que resulta em um incremento exponencial do grão que passou à ter um tamanho de seis máquinas virtuais. Esses recursos ficaram prontos e foram entregues 496s depois, no momento de 1111s da execução. A carga começa, então, a cair e rapidamente o Gerenciador da Nuvem reduz os recursos em cinco máquinas virtuais, através de uma operação que iniciou aos 1124s com a detecção de infração da *threshold* inferior e que é completada 7s depois. Através desse processo é possível observar, de forma clara, que ao utilizar a elasticidade multinível com grão adaptativo, o tempo para responder mudanças de variações nas cargas é reduzido, levando ao aumento da reatividade, redução do tempo total de processamento e da quantidade de ações elásticas necessárias.

Figura 30 – Análise de detecção e efetivação da operação de elasticidade no perfil de carga oscilante no cenário C4: elasticidade multinível com grão adaptativo.



Fonte: Elaborado pelo autor.

7 CONCLUSÃO

Para atender a crescente demanda de dispositivos móveis conectados as redes sem fio de longa distância, iniciativas científicas e indústria voltaram esforços para o desenvolvimento de uma nova geração de redes móveis, a rede 5G. Propôs-se, então, a arquitetura C-RAN, que tem por escopo viabilizar e atender as especificações da nova geração de redes móveis, permitir que os operadores respondam as demandas através do uso de técnicas de virtualização e recursos de processamento de dados centralizado, explorando principalmente a capacidade de elasticidade da computação em nuvem. Considerando os inúmeros desafios que precisam ser enfrentados em relação a orquestração de recursos, as métricas para tomada de decisão de ações e a distribuição de requisições para processamento, nesta dissertação buscou-se desenvolver o modelo Elastic-RAN para realizar orquestração automática de recursos em arquitetura C-RAN.

Assim, na Seção 1.2, realizou-se a seguinte questão de pesquisa: Como seria um modelo de elasticidade multinível para C-RAN com orquestração automática de recursos capaz de adaptar o grão de elasticidade para provisionar tanto BBU Pools quanto BBUs de forma não bloqueante, com desempenho e o menor custo de infraestrutura possível? Com o objetivo de responder a questão, Elastic-RAN propõe um conceito de elasticidade multinível não bloqueante, com orquestração automática de recursos através da coordenação dos BBUs presentes em cada um dos BBUs Pools disponíveis, junto a um mecanismo de grão elástico adaptativo, que considera o uso corrente dos recursos para provisionar e mapear as diferentes quantidades de recursos em tempo de execução para cada ação elástica. A elasticidade multinível não bloqueante permite atuar tanto no nível de BBU Pool, devido ao alto volume de tráfego e a distância máxima sugerida entre os RRHs e os pools (CHANCLOU et al., 2013), quanto no nível de BBU, tendo em vista o alto processamento de CPU e memória necessária para as requisições, de forma assíncrona para não penalizar os processamentos correntes. O mecanismo de elasticidade com grão elástico adaptativo, de seu turno, permite provisionar e mapear os recursos sob demanda e em tempo de execução, demandando menor quantidade de ações elásticas com um tamanho de grão mais adequado para as necessidades correntes de recursos e aumentando o desempenho e a reatividade do sistema, respondendo mais rapidamente as variações de carga.

Para a realização dos experimentos, explorou-se quatro cenários de testes: (i) execução da aplicação com a quantidade mínima de recursos e sem o uso da elasticidade; (ii) execução da aplicação com elasticidade apenas no nível de BBU e com o grão de elasticidade estático; (iii) execução da aplicação com elasticidade multinível e com o grão de elasticidade estático; (iv) execução da aplicação com elasticidade multinível e o grão de elasticidade adaptativo. Estes cenários foram testados frente cinco diferentes perfis de cargas: (i) carga crescente; (ii) carga decrescente; (iii) carga constante; (iv) carga oscilante e (v) carga exponencial. A simulação desses perfis de carga foi dada através da criação de uma aplicação intensiva de CPU e rede, que explora a transferência de *streamings* e o processamento de decodificação de blocos.

Métricas de avaliação de desempenho e consumo de recursos da aplicação foram apresen-

tadas e, da análise de resultados, foi possível demonstrar o impacto do uso de diferentes cargas de processamentos nos diferentes cenários para executar ações de elasticidade no desempenho e recursos da aplicação. Como principais resultados, observou-se que o Elastic-RAN atingiu ganhos entre 4% e 26%, em relação aos custos de execução, e obteve a maior eficiência para todos os perfis de carga, além de reduzir até 55% a quantidade de operações elásticas necessárias, quando comparado à abordagem de elasticidade tradicional. Ainda, frente a abordagem sem elasticidade, os ganhos de custos foram ainda superiores, ficando entre 51% e 70%.

7.1 Contribuições

O modelo Elastic-RAN foi desenvolvido com o objetivo de preencher as lacunas identificadas a partir da análise de trabalhos que exploram alto desempenho para arquitetura C-RAN. Como resultado, foram obtidas adições ao estado da arte, podendo listar as seguintes contribuições científicas:

- Elasticidade multinível não bloqueante: dado que a elasticidade automática tem como desafio não causar impacto à execução da aplicação durante a reorganização de recursos, o modelo Elastic-RAN apresenta uma forma de orquestração autônoma de recursos que vai além do gerenciamento no nível de máquinas virtuais, pois também considera o volume de tráfego na rede para atuar no nível de máquinas físicas. Nada obstante, tem a capacidade de realizar as operações de elasticidade de maneira não bloqueante, utilizando uma área de dados compartilhada para gerar notificações que, em união com o *middleware* da nuvem, permitem disponibilizar estes recursos para a aplicação, sem penalizar os processamentos das tarefas correntes.

- Grão elástico adaptativo: enquanto as abordagens de elasticidade tradicionais atuam com um grão de tamanho fixo, o Elastic-RAN explora uma abordagem de grão de elasticidade adaptativo, que considera métricas importantes na C-RAN (CPU, memória e rede) para realizar ações elásticas de forma mais precisa, reduzindo a quantidade de operações e aumentando a reatividade ao responder mais rapidamente às variações de carga.

7.2 Trabalhos Futuros

Após a realização deste estudo, vislumbrou-se a possibilidade de exploração de outros pontos em futuros trabalhos, tais como:

- Criação de um módulo para realizar a tomada de decisões elásticas de forma proativa;

- Explorar uma abordagem de elasticidade híbrida através da aplicação da elasticidade horizontal e vertical.

- Adaptação de um *middleware* de infraestrutura na nuvem que realize criação e alocação de recursos de forma paralela, no intuito de reduzir o tempo entre solicitação e entrega de grandes quantidade de recursos.

- Explorar a utilização de virtualização com base em *containers* ao invés da utilização das tradicionais máquinas virtuais.

7.3 Publicações

Ao longo de todo o período desta pesquisa foram produzidos diversos artigos para publicação em revistas e eventos. Além dos artigos referentes a esta dissertação, também foram produzidos outros artigos através da colaboração com outros grupos de pesquisa e universidades. A seguir são listados os artigos publicados durante este período:

- ANDRIOLI, LEANDRO; DA ROSA RIGHI, RODRIGO ; ROSTIROLLA, GUSTAVO ; ANDRE DA COSTA, CRISTIANO . Proposal of a network congestion-aware RFID model for online management of assets. In: 2016 35th International Conference of the Chilean Computer Science Society (SCCC), 2016, Valparaíso. 2016 35th International Conference of the Chilean Computer Science Society (SCCC), 2016. p. 1.

- ANDRIOLI, L.; RIGHI, R. R. . AGANT: Proposta de um Modelo Ciente do Tráfego da Rede para Ambientes Inteligentes. In: Anis da 17a Escola Regional de Alto Desempenho do Estado do Rio Grande do Sul (ERAD/RS 2017), 2017, Ijuí. Sociedade Brasileira de Computação, 2017. v. 17.

- ANDRIOLI, L.; RIGHI, R. R. ; COSTA, C. A. ; GRAEBIN, L. . Observing Network Performance and Congestion on Managing Assets with RFID and Cloud Computing. In: Journal of Computer and Communications, 2017. Journal of Computer and Communications.

- ANDRIOLI, L; RIGHI, R. R. ; AUBIN, M. Analisando Métodos e Oportunidades Em Redes Definidas por Software (SDN) para Otimizações de Tráfego de Dados, 2017 (volume 9, número 4) da Revista Brasileira de Computação Aplicada (RBCA - ISSN 2176-6649).

REFERÊNCIAS

AGRAWAL, D. et al. Database scalability, elasticity, and autonomy in the cloud. In: INTERNATIONAL CONFERENCE ON DATABASE SYSTEMS FOR ADVANCED APPLICATIONS, 2011, Hong Kong. **Anais...** 2011. p. 2–15.

AL-DHURAIBI, Y. et al. Elasticity in cloud computing: state of the art and research challenges. **IEEE Transactions on Services Computing**, p. 1–17, jun 2017.

AL-DULAIMI, A.; AL-RUBAYE, S.; NI, Q. Energy efficiency using cloud management of lte networks employing fronthaul and virtualized baseband processing pool. **IEEE Transactions on Cloud Computing**, v. PP, n. 99, p. 1–12, my 2016.

AL-HAIDARI, F.; SQALLI, M.; SALAH, K. Impact of cpu utilization thresholds and scaling size on autoscaling cloud resources. In: CLOUD COMPUTING TECHNOLOGY AND SCIENCE (CLOUDCOM), 2013 IEEE 5TH INTERNATIONAL CONFERENCE ON, 2013. **Anais...** 2013. v. 2, p. 256–261.

BALIGA, J. et al. Green cloud computing: balancing energy in processing, storage, and transport. **Proceedings of the IEEE**, v. 99, n. 1, p. 149–167, 2011.

BERSANI, M. M. et al. Towards the formalization of properties of cloud-based elastic systems. In: INTERNATIONAL WORKSHOP ON PRINCIPLES OF ENGINEERING SERVICE-ORIENTED AND CLOUD SYSTEMS, 6., 2014. **Proceedings...** 2014. p. 38–47.

BHAUMIK, S. et al. Cloudiq: a framework for processing base stations in a data center. **Mobile computing and networking**, Istanbul, p. 125–136, aug 2012.

BUYYA, R. High performance cluster computing: programming and applications, vol. 2. **Pre ticeHallPTR, NJ**, v. 29, 1999.

CAI, B. et al. Research and application of migrating legacy systems to the private cloud platform with cloudstack. In: AUTOMATION AND LOGISTICS (ICAL), 2012 IEEE INTERNATIONAL CONFERENCE ON, 2012. **Anais...** 2012. p. 400–404.

CHABBOUH, O. et al. Service scheduling scheme based load balancing for 5g/hetnets cloud ran. **Advanced Information Networking and Applications (AINA)**, Taipei, may 2017.

CHANCLOU, P. et al. Optical fiber solution for mobile fronthaul to achieve cloud radio access network. In: FUTURE NETWORK AND MOBILE SUMMIT (FUTURENETWORKSUMMIT), 2013, 2013. **Anais...** 2013. p. 1–11.

CHECKO, A. et al. Cloud ran for mobile networks—a technology overview. **IEEE Communications Surveys & Tutorials**, v. 17, n. 1, sep 2014.

192 p.

CHIU, D.; AGRAWAL, G. Evaluating caching and storage options on the amazon web services cloud. In: GRID COMPUTING (GRID), 2010 11TH IEEE/ACM INTERNATIONAL CONFERENCE ON, 2010. **Anais...** 2010. p. 17–24.

CMRI. C-ran: the road towards green ran. **C-RAn International Workshop**, Beijing, apr 2010.

DATAPIPE. **Gogrid**. Disponível em: <https://www.datapipe.com/gogrid>. Acesso em: 26 outubro 2017.

DAWOUD, W.; TAKOUNA, I.; MEINEL, C. Elastic vm for cloud resources provisioning optimization. **Advances in Computing and Communications**, p. 431–445, 2011.

DUAN, T. et al. Inter-bbu control mechanism for load balancing in c-ran-based bbu pool. **Computer and Communications (ICCC)**, Chengdu, may 2017.

DUTTA, S. et al. Smartscale: automatic application scaling in enterprise clouds. In: CLOUD COMPUTING (CLOUD), 2012 IEEE 5TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** 2012. p. 221–228.

ERICSSON, W. P. **More than 50 billion connected devices**. Disponível em: <http://www.inf.ufpr.br/info/techrep/RT_DINF004_2004.pdf>. Acesso em: 26 outubro 2017.

FACCO RODRIGUES, V. et al. Brokel: towards enabling multi-level cloud elasticity on publish/subscribe brokers. **International Journal of Distributed Sensor Networks**, v. 13, n. 8, p. 1550147717728863, 2017.

GALANTE, G.; BONA, L. C. E. de. A survey on cloud computing elasticity. **International Conference on Utility and Cloud Computing (UCC)**, Chicago, nov 2012.

GIBSON, J. et al. Benefits and challenges of three cloud computing service models. **Computational Aspects of Social Networks (CASoN)**, Sao Carlos, v. 01, p. 198–205, nov 2012.

GONG, J. et al. Greenbase: an energy-efficient middleware for baseband units in radio access networks. **Wireless Communications and Networking Conference (WCNC)**, Shanghai, jul 2013.

HAJISAMI, A.; POMPILI, D. Dynamic joint processing: achieving high spectral efficiency in uplink 5g cellular networks. **Computer Networks**, v. 126, p. 44–56, 2017.

HAJISAMI, A.; TRAN, T. X.; POMPILI, D. Elastic-net: boosting energy efficiency and resource utilization in 5g c-rans. **Networking and Internet Architecture**, Oct. 2017.

HERBST, N. R. et al. Self-adaptive workload classification and forecasting for proactive resource provisioning. **ACM/SPEC International Conference on Performance Engineering**, Beijing, p. 187–198, apr 2013.

HU, Y. et al. A survey on evaluating elasticity of cloud computing platform. **ACM SIGCOMM Computer Communication Review**, Puerto Rico, aug 2016.

IMAI, S.; CHESTNA, T.; VARELA, C. A. Elastic scalable cloud computing using application-level migration. In: UTILITY AND CLOUD COMPUTING (UCC), 2012 IEEE FIFTH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** 2012. p. 91–98.

ISLAM, S. et al. How a consumer can measure elasticity for cloud platforms. In: ACM/SPEC INTERNATIONAL CONFERENCE ON PERFORMANCE ENGINEERING, 3., 2012. **Proceedings...** 2012. p. 85–96.

JAMSHIDI, P.; AHMAD, A.; PAHL, C. Autonomic resource provisioning for cloud-based software. In: INTERNATIONAL SYMPOSIUM ON SOFTWARE ENGINEERING FOR ADAPTIVE AND SELF-MANAGING SYSTEMS, 9., 2014. **Proceedings...** 2014. p. 95–104.

KAUR, S.; KAUR, G. A review of load balancing strategies for distributed systems. **International Journal of Computer Applications**, v. 121, n. 18, p. 45–47, July 2015.

546 p.

KHAN, M.; ALHUMAIMA, R.; AL-RAWESHIDY, H. Quality of service aware dynamic bbu-rrh mapping in cloud radio access network. **Emerging Technologies (ICET)**, Peshawar, jan 2016.

KUMAR, V. et al. A message passing interface to support fast data access in distributed cloud environment along with master and slave communication. In: CURRENT TRENDS IN ENGINEERING AND TECHNOLOGY (ICCTET), 2014 2ND INTERNATIONAL CONFERENCE ON, 2014. **Anais...** 2014. p. 309–312.

LEE, Y. et al. Exploring the tradeoffs between programmability and efficiency in data-parallel accelerators. In: ACM SIGARCH COMPUTER ARCHITECTURE NEWS, 2011. **Anais...** 2011. v. 39, n. 3, p. 129–140.

LI, A. et al. An energy-effective network deployment scheme for 5g cloud radio access networks. **Computer Communications Workshops (INFOCOM WKSHPS)**, San Francisco, sep 2016.

LORIDO-BOTRAN, T.; MIGUEL-ALONSO, J.; LOZANO, J. A. A review of auto-scaling techniques for elastic applications in cloud environments. **Journal of Grid Computing**, v. 12, n. 4, p. 559–592, 2014.

MALATHI, M. Cloud computing concepts. **Electronics Computer Technology (ICECT), 3rd International Conference**, Kanyakumari, v. 06, p. 236–239, nov 2011.

MAO, M.; HUMPHREY, M. Auto-scaling to minimize cost and meet application deadlines in cloud workflows. In: HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS (SC), 2011 INTERNATIONAL CONFERENCE FOR, 2011. **Anais...** 2011. p. 1–12.

MAROTTA, M. A. et al. Resource sharing in heterogeneous cloud radio access networks. **IEEE Wireless Communications**, v. 22, n. 3, p. 74–82, 2015.

MARSHALL, P.; KEAHEY, K.; FREEMAN, T. Elastic site: using clouds to elastically extend site resources. In: CLUSTER, CLOUD AND GRID COMPUTING (CCGRID), 2010 10TH IEEE/ACM INTERNATIONAL CONFERENCE ON, 2010. **Anais...** 2010. p. 43–52.

MELL, P.; GRANCE, T. The nist definition of cloud computing. **NIST Special Publication 800-145**, Gaithersburg, p. 1–3, set 2011.

MILOJIČIĆ, D.; LLORENTE, I. M.; MONTERO, R. S. Opennebula: a cloud management tool. **IEEE Internet Computing**, v. 15, n. 2, p. 11–14, 2011.

MOBILE, C. C-ran: the road towards green ran. **White Paper, ver**, v. 2, 2011.

NIKAEIN, N. et al. Demo – closer to cloud-ran: ran as a service. **ACM SIGCOMM Computer Communication Review**, Paris, p. 193–195, sep 2015.

PARK, S.-H. et al. Robust and efficient distributed compression for cloud radio access networks. **IEEE/OSA Journal of Optical Communications and Networking**, v. 62, n. 2, feb 2013.

QIAN, M. et al. A super base station based centralized network architecture for 5g mobile communication systems. **Digital Communications and Networks**, v. 1, n. 2, p. 152–159, feb 2015.

RAKOVIC, V. et al. Analysis of virtual resource allocation for cloud-ran based systems. **Internet and Networks (ICIN)**, Paris, v. 39, n. 1, apr 2017.

RAZA, M. R. et al. Demonstration of dynamic resource sharing benefits in an optical c-ran. **IEEE/OSA Journal of Optical Communications and Networking**, v. 8, n. 8, aug 2016.

RIMAL, B. P.; CHOI, E.; LUMB, I. A taxonomy and survey of cloud computing systems. **NCM**, v. 9, p. 44–51, 2009.

ROLOFF, E. et al. Evaluating high performance computing on the windows azure platform. In: CLOUD COMPUTING (CLOUD), 2012 IEEE 5TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** 2012. p. 803–810.

ROSA RIGHI, R. da. Elasticidade em cloud computing: conceito, estado da arte e novos desafios. **Revista Brasileira de Computação Aplicada**, v. 5, n. 2, p. 2–17, 2013.

ROSA RIGHI, R. da et al. Autoelastic: automatic resource elasticity for high performance applications in the cloud. **IEEE Transactions on Cloud Computing**, v. 4, n. 1, p. 6–19, 2016.

ROSA RIGHI, R. et al. Joint-analysis of performance and energy consumption when enabling cloud elasticity for synchronous hpc applications. **Concurrency and Computation: Practice and Experience**, v. 28, n. 5, p. 1548–1571, 2016.

SAH, S. K.; JOSHI, S. R. Scalability of efficient and dynamic workload distribution in autonomic cloud computing. **Issues and Challenges in Intelligent Computing Techniques (ICICT)**, Ghaziabad, Apr. 2014.

SAHU, B. J. R. et al. Energy-efficient bbu allocation for green c-ran. **IEEE Communications Letters**, v. 21, n. 7, apr 2017.

SCHOLZ, S.; GROB-LIPSKI, H. Reallocation strategies for user processing tasks in future cloud-ran architectures. **Personal, Indoor, and Mobile Radio Communications (PIMRC)**, Valencia, dec 2016.

SEMCHEDINE, F.; BOUALLOUCHE-MEDJKOUNE, L.; AÏSSANI, D. Task assignment policies in distributed server systems: a survey. **Journal of network and Computer Applications**, London, v. 34, n. 4, p. 1123–1130, 2011.

SIGWELE, T. et al. Energy-efficient cloud radio access networks by cloud based workload consolidation for 5g. **Journal of Network and Computer Applications**, v. 78, p. 1–8, jan 2017.

SIGWELE, T.; PILLAI, P.; HU, Y. F. Call admission control in cloud radio access networks. **International Conference on Future Internet of Things and Cloud (FiCloud)**, Barcelona, aug 2014.

SIGWELE, T.; PILLAI, P.; HU, Y. F. itree: intelligent traffic and resource elastic energy scheme for cloud-ran. **Future Internet of Things and Cloud (FiCloud)**, Rome, oct 2015.

SOSINSKY, B. **Cloud computing bible**. 1. ed. Indianapolis: Wiley Publishing, 2011.

SPINNER, S. et al. Runtime vertical scaling of virtualized applications via online model estimation. In: SELF-ADAPTIVE AND SELF-ORGANIZING SYSTEMS (SASO), 2014 IEEE EIGHTH INTERNATIONAL CONFERENCE ON, 2014. **Anais...** 2014. p. 157–166.

TALEB, H. et al. Centralized and distributed rrh clustering in cloud radio access networks. **Computers and Communications (ISCC)**, Heraklion, sep 2017.

TRAN, G. S. et al. Cooperative resource management in a iaas. In: ADVANCED INFORMATION NETWORKING AND APPLICATIONS (AINA), 2015 IEEE 29TH INTERNATIONAL CONFERENCE ON, 2015. **Anais...** 2015. p. 611–618.

TRAN, T. X.; HAJISAMI, A.; POMPILI, D. Quaro: a queue-aware robust coordinated transmission strategy for downlink c-rans. In: SENSING, COMMUNICATION, AND NETWORKING (SECON), 2016 13TH ANNUAL IEEE INTERNATIONAL CONFERENCE ON, 2016. **Anais...** 2016. p. 1–9.

p. 5–21.

VAQUERO, L. M. et al. A break in the clouds: towards a cloud definition. **ACM SIGCOMM Computer Communication Review**, New York, v. 39, n. 1, p. 50–55, jan 2009.

VNI, C. **Cisco visual networking index**: forecast and methodology, 2016–2021. Disponível em: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>. Acesso em: set. 2017.

WANG, X. et al. Dynamic resource scheduling in cloud radio access network with mobile cloud computing. **Quality of Service (IWQoS)**, Beijing, oct 2016.

WEN, X. et al. Comparison of open-source cloud management platforms: openstack and opennebula. In: FUZZY SYSTEMS AND KNOWLEDGE DISCOVERY (FSKD), 2012 9TH INTERNATIONAL CONFERENCE ON, 2012. **Anais...** 2012. p. 2457–2461.

WINDOWS. **Azure**. Disponível em: <https://azure.microsoft.com>. Acesso em: 26 outubro 2017.

XU, C.; LAU, F. **Load balancing in parallel computers**: theory and practice. Kluwer: Springer US, 1997.

XU, S.; WANG, S. Efficient algorithm for baseband unit pool planning in cloud radio access networks. **Vehicular Technology Conference (VTC Spring)**, Nanjing, jul 2016.

YASEEN, F. A.; AL-KHALIDI, N. A.; AL-RAWESHIDY, H. S. Smart virtual enb (svenb) for 5g mobile communication. **Fog and Mobile Edge Computing (FMEC)**, Valencia, jun 2017.

ZHANG, Q.; CHENG, L.; BOUTABA, R. Cloud computing: state-of-the-art and research challenges. **Journal of internet services and applications**, v. 1, n. 1, p. 7–18, 2010.

ZHANG, Y.; KAMEDA, H.; HUNG, S.-L. Comparison of dynamic and static load-balancing strategies in heterogeneous distributed systems. **IEE Proceedings - Computers and Digital Techniques**, v. 144, n. 2, Mar. 1997.