



Universidade Estadual de Campinas
Instituto de Computação



Diogo Machado Gonçalves

Migração Proativa de Máquinas Virtuais em
Computação em Névoa Para Usuários Móveis

CAMPINAS
2018

Diogo Machado Gonçalves

**Migração Proativa de Máquinas Virtuais em Computação em
Névoa Para Usuários Móveis**

Dissertação apresentada ao Instituto de
Computação da Universidade Estadual de
Campinas como parte dos requisitos para a
obtenção do título de Mestre em Ciência da
Computação.

Orientador: Prof. Dr. Edmundo Roberto Mauro Madeira
Coorientador: Prof. Dr. Luiz Fernando Bittencourt

Este exemplar corresponde à versão final da
Dissertação defendida por Diogo Machado
Gonçalves e orientada pelo Prof. Dr.
Edmundo Roberto Mauro Madeira.

CAMPINAS
2018

Agência(s) de fomento e nº(s) de processo(s): CAPES

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

G586m Gonçalves, Diogo Machado, 1993-
Migração proativa de máquinas virtuais em computação em névoa para usuários móveis / Diogo Machado Gonçalves. – Campinas, SP : [s.n.], 2018.

Orientador: Edmundo Roberto Mauro Madeira.
Coorientador: Luiz Fernando Bittencourt.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Computação em nuvem. 2. Máquinas virtuais. 3. Internet das coisas. 4. Dispositivos móveis. 5. Recursos de redes de computadores. I. Madeira, Edmundo Roberto Mauro, 1958-. II. Bittencourt, Luiz Fernando, 1981-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Proactive virtual machine migration in fog computing environments for mobile users

Palavras-chave em inglês:

Cloud computing

Virtual machines

Internet of things

Mobile devices

Computer network resources

Área de concentração: Ciência da Computação

Titulação: Mestre em Ciência da Computação

Banca examinadora:

Edmundo Roberto Mauro Madeira [Orientador]

Alfredo Goldman Vel Lejbman

Leandro Aparecido Villas

Data de defesa: 10-10-2018

Programa de Pós-Graduação: Ciência da Computação



Universidade Estadual de Campinas
Instituto de Computação



Diogo Machado Gonçalves

Migração Proativa de Máquinas Virtuais em Computação em Névoa Para Usuários Móveis

Banca Examinadora:

- Prof. Dr. Edmundo Roberto Mauro Madeira
IC/UNICAMP
- Prof. Dr. Alfredo Goldman Vel Lejbman
IME/USP
- Prof. Dr. Leandro Aparecido Villas
IC/UNICAMP

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 10 de outubro de 2018

*É fazendo qualquer coisa que você se torna
qualquer um*
(Remi Gaillard)

Agradecimentos

Desde que desembarquei em Campinas há dois anos minha vida mudou completamente. Fui apresentado a pessoas, situações e sentimentos que nunca imaginei conhecer. O que a Unicamp me ensinou, dentro e fora do campus, não tenho como descrever.

Muitas pessoas incríveis entraram na minha vida nesse período. Algumas vêm me acompanhando desde o início, outras apareceram no meio do caminho, algumas chegaram e logo se foram e outras acabaram de chegar prometendo ficar por muito tempo. A todas elas, meus mais sinceros agradecimentos.

Nem tudo são novidades nessa jornada. Algumas relações se mantiveram intactas mesmo com toda essa minha fase de mudança. Muitas pessoas continuaram próximas mesmo com a distância física.

Agradeço primeiramente a Deus pela companhia e pela paz nos momentos difíceis.

Agradeço imensamente à minha família e, em especial, aos meus pais, pelo apoio durante toda essa minha nova fase de pós-graduando. Agradeço pelo carinho mesmo eu não retribuindo a atenção que vocês mereciam.

Não poderia encerrar meus agradecimentos sem antes mencionar as pessoas com as quais mais dividi meu tempo durante meu mestrado.

Sou imensamente grato aos meus orientadores, Edmundo e Luiz, pelos ensinamentos passados, pelos conselhos mas, principalmente, pela paciência e pelas palavras de motivação. Não foram poucas as vezes em que eu saí de uma reunião melhor do que como entrei.

Agradeço, em especial, aos amigos do meu laboratório de pesquisa, o laboratório mais silencioso que esse Instituto de Computação já viu. Agradeço a companhia durante as madrugadas, pelas risadas e pelos conselhos, por compartilharem o café e os desabafos. Vocês fizeram os momentos ruins se tornarem muito mais leves.

Por fim, mas não menos importante, agradeço ao projeto InterSCity pelo suporte para realizar minha pesquisa, e à CAPES pelo apoio financeiro.

Este trabalho faz parte do INCT sobre Internet do Futuro para Cidades Inteligentes (CNPq 465446/2014-0, CAPES 88887.136422/2017-00 e FAPESP 2014/50937-1).

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Resumo

A Computação em Névoa provê a dispositivos IoT acesso com baixa latência a recursos computacionais e de rede. Neste ambiente porém, a alta mobilidade de alguns desses dispositivos, como *wearables* ou dispositivos embarcados em veículos, traz grandes desafios para a alocação e gerência de recursos nesse ambiente. Manter a aplicação o mais próximo possível do usuário pode ser uma maneira de oferecer as condições necessárias para uma boa execução. A partir das diferentes demandas por recursos por parte das aplicações e da alta mobilidade dos usuários no cenário de cidades inteligentes, otimizar o processo de migração da máquina virtual que executa a aplicação do usuário nesse ambiente pode ser uma maneira de assegurar a qualidade de serviço demandada pelas aplicações. Trabalhos recentes em variados contextos têm demonstrado que adicionar dados de previsão de mobilidade do usuário pode melhorar o gerenciamento de recursos computacionais. Baseado no estado atual da arte com relação à migração de máquinas virtuais em um ambiente de Computação em Névoa para usuários móveis, este trabalho propõe mecanismos baseados na previsão de mobilidade para otimizar a escolha do destino da máquina virtual durante o seu processo. Simulações de cenários compatíveis com o contexto de uma Cidade Inteligente sugerem que a utilização de informações sobre o trajeto futuro do usuário pode melhorar o gerenciamento de recursos da Névoa, mantendo a máquina virtual do usuário em dispositivos de Névoa tão próximos quanto possível do trajeto do usuário. As soluções apresentadas reduzem o número de migrações necessárias durante o trajeto do usuário sem prejudicar o tempo de resposta da máquina virtual alocada na Névoa.

Abstract

Fog computing provides a low latency access to computational and network resources at the edge of the network for the resource-constrained Internet of Things devices. The high mobility of some of these devices, such as wearables or embedded ones in vehicles, bikes or trains, brings great challenges related to resource allocation and management. Keeping the users' applications as close as possible is a way to improve the Quality of Service desired by the users. Based on the wide different applications and high mobility of the users in a scenario of smart cities, optimizing the Virtual Machine migration process may be a way to ensure such Quality of Service requirements. Recent work in a variety of contexts has shown that adding user mobility prediction data can improve the management of computing resources. Based on the current state of the art about the virtual machine migration in a Fog Computing environment for mobile users, this work proposes mechanisms based on the user mobility prediction to optimize the choice of the destination node of the virtual machine process. Simulations of smart cities scenarios suggest that using knowledge about the future user's path can improve the resource management of Fog ecosystems, maintaining user's virtual machine in Fog devices as close as possible to the user's path. Simulations suggest that the presented policy reduces the total of migrations along the user's path without affecting the quality response time of virtual machines allocated to the Fog.

Lista de Figuras

1.1	Fluxo de execução de uma aplicação para assistência cognitiva realizando <i>offloading</i> do processamento para uma <i>Cloudlet</i> . Fonte: Adaptado de Jiang, Huang e Tsang (2018) [22]	15
1.2	Exemplo de cenário de migração em Névoa de um guia turístico virtual	16
2.1	Crescimento global do número de dispositivos móveis e conexões. Fonte: Adaptado de V. N. I. Cisco (2016) [10]	22
2.2	Ilustração do conceito de virtualização em servidores. Fonte: Lopes (2017) [26]	23
2.3	Arquitetura de um ambiente de Computação em Névoa para atender usuários móveis. Fonte: Bittencourt <i>et al.</i> (2015) [6].	25
2.4	Latência obtida por um usuário móvel em cenários de migração e não migração de sua máquina virtual ao longo de seu trajeto. Fonte: Adaptado de Lopes (2017) [26]	26
2.5	Cenários de migração de máquinas virtuais em um ambiente com usuários móveis. Fonte: Adaptado de Yu <i>et al.</i> (2013) [47].	27
4.1	Ilustração da sequência de migrações em um cenário de migrações reativas e proativas. Fonte: Adaptado de Yu <i>et. al</i> (2013) [46].	36
4.2	Ilustração do processo de migração proativa entre os nós da Névoa.	37
4.3	Arquitetura lógica de um ambiente de Computação em Névoa com suporte à mobilidade dos dispositivos IoT.	40
4.4	Integração entre os simuladores SUMO e MyIFogSim	44
4.5	Sequência de passos do processo de migração. Adaptado de Lopes <i>et al.</i> (2017) [27]	46
4.6	Processo de migração a partir da área de cobertura do ponto de acesso. Adaptado de Lopes <i>et al.</i> (2017) [27]	47
4.7	Número de migrações durante o trajeto	48
4.8	Intervalo de tempo entre migrações	49
4.9	Tempo médio em que o veículo ficou sem acesso à máquina virtual a cada migração.	50
4.10	Latência	51
4.11	Período proporcional de indisponibilidade de acesso à máquina virtual	52
4.12	Cenário de migração de máquinas virtuais com concorrência por recursos.	53
4.13	Número médio de migrações da VM durante o trajeto do usuário	58
4.14	Período proporcional de indisponibilidade de acesso à VM	58
4.15	Latência média	59
4.16	Latência em pior caso	60

Lista de Tabelas

3.1	Trabalhos relacionados à migração de máquinas virtuais em Névoa ou migração de conteúdo baseado no deslocamento futuro do usuário.	29
3.2	Trabalhos relacionados à alocação e distribuição de aplicações (<i>Service Placement</i>) em Névoa.	33
4.1	Características do conjunto de dados utilizado	48
4.2	Variáveis do modelo linear proposto	55

Lista de Acrônimos

D2D	Device to Device
D2I	Device to Infrastructure
FMC	Follow Me Cloud
I2I	Infrastructure to Infrastructure
IoT	Internet of Things
IoMT	Internet of Mobile Things
MANET	Mobile Ad-hoc Network
MCC	Mobile Cloud Computing
MIPS	Milhões de Instruções Por Segundo
PLI	Programação Linear Inteira
RSU	Roadside Unit
SUMO	Simulation of Urban Mobility
VANET	Vehicular Ad-hoc Network
VCC	Vehicular Cloud Computing
VM	Virtual Machine

Sumário

1	Introdução	13
1.1	Problema	17
1.2	Contribuições	18
2	Fundamentação Teórica	21
2.1	Internet das Coisas	21
2.2	Computação em Nuvem e em Névoa	23
2.3	Migração de Máquinas Virtuais	25
3	Trabalhos Relacionados	28
3.1	Migração de máquinas virtuais e predição de mobilidade	28
3.2	Localização de aplicações	32
4	Migração Proativa de Máquinas Virtuais	34
4.1	Política de migração proposta	35
4.2	Arquitetura	38
4.3	Cenário 1 - Avaliação em cenário otimista	43
4.3.1	Metodologia	43
4.3.2	Resultados	48
4.4	Cenário 2 - Concorrência por recursos	52
4.4.1	Modelo de alocação proposto	54
4.4.2	Metodologia	56
4.4.3	Resultados	57
5	Conclusão	62
	Referências Bibliográficas	66

Capítulo 1

Introdução

A popularização de paradigmas como o da Internet das Coisas (Internet of Things - IoT), no qual diversos objetos como celulares, relógios, veículos e máquinas industriais passam a ter conexão com a Internet, tem provocado mudanças na maneira de se prover recursos de rede suficientes para atender a demanda desse novo cenário. Estima-se que o número de dispositivos IoT chegue a cerca de 50 bilhões em 2020 [9], sendo em 2021, 929 milhões de dispositivos *wearables* (dispositivos tecnológicos que podem ser utilizados pelos usuários como peças do vestuário, como óculos, relógios ou pulseiras) [10].

Atualmente, esse volume de dispositivos acaba gerando uma grande demanda por armazenamento e processamento, além de uma intensa transmissão de dados entre eles. Esses dispositivos, no entanto, geralmente possuem poucos recursos computacionais disponíveis, não oferecendo recursos suficientes para a execução adequada de aplicações mais complexas que poderiam ser úteis neste contexto. Câmeras de vídeo posicionadas próximas aos semáforos poderiam, por exemplo, com acesso a mais recursos computacionais, auxiliar um sistema que identifica e informa aos motoristas vagas de estacionamento disponíveis nas ruas ou ainda, acionar os hospitais caso algum acidente de trânsito ocorra no local.

Uma alternativa proposta para este problema foi possibilitar que os dispositivos IoT acessem recursos disponíveis remotamente. Esta técnica, conhecida como *offloading*, permite a execução parcial ou integral desses aplicativos em um ambiente remoto ao dispositivo de origem. Neste cenário, caso o dispositivo não ofereça recursos suficientes para a execução do aplicativo, os dados necessários para esta execução seriam coletados pelo dispositivo e enviados para uma fonte de recursos adequada que realiza o processamento dessas informações. Após o processamento remoto dos dados coletados, o resultado pode ser devolvido para o dispositivo original. Estudos têm mostrado que esse processo de *offloading* do processamento de aplicativos não só apresenta uma diminuição no tempo de processamento se comparada com a execução utilizando os recursos locais do dispositivos, como também tem permitido a execução de aplicações mais complexas em contextos antes não explorados [30, 23].

Aplicativos mais robustos podem utilizar conceitos da Computação em Nuvem (*Cloud Computing*) ao utilizarem fontes mais ricas em recursos para executar a aplicação. A conexão com esses grandes conjuntos de servidores, característicos da Nuvem, porém, pode comprometer o funcionamento de aplicações que demandam uma resposta rápida para

o seu pedido de processamento. Visto que geralmente os recursos físicos oferecidos pela Nuvem estão concentrados em grandes volumes no núcleo da rede, o tempo de comunicação entre os dispositivos IoT e a fonte de recursos da Nuvem pode comprometer a execução da aplicação.

Para amenizar esse cenário, conceitos de Computação em Névoa (*Fog Computing*) têm sido incorporados neste contexto. Servidores menores, denominados *Cloudlets* ou nós da Névoa, têm sido posicionados de forma distribuída por toda a extensão de uma área de interesse a fim de oferecer um acesso a recursos computacionais mais próximos aos usuários, diminuindo a latência na comunicação.

Os recursos físicos disponíveis tanto por servidores na Nuvem quanto pelas *Cloudlets* são particionados em máquinas virtuais (*Virtual Machine - VM*), plataformas que abstraem os recursos físicos do hardware, permitindo a personalização dos recursos oferecidos ao cliente e a migração do conteúdo lógico da VM de uma máquina física para outra [32].

Potencialmente, qualquer dispositivo IoT pode usufruir dos recursos disponíveis na Névoa. O foco deste trabalho no entanto, se dará para um subconjunto dessa categoria: os dispositivos IoT móveis (*Mobile IoT* ou *Internet of Mobile Things*), tais como *smartphones*, dispositivos *wearables* ou embarcados a objetos tradicionalmente móveis, como bicicletas e veículos.

Dada a mobilidade destes dispositivos, desenvolver sistemas para transferir de forma eficiente os dados e aplicações dos usuários entre os servidores da Nuvem e da Névoa de acordo com a localização dos dispositivos se torna uma tarefa desejável tanto pelos provedores da infraestrutura quanto pelos usuários. Em algumas situações, alguns grupos de usuários, como usuários de ônibus urbanos, metrô e trem, podem apresentar um padrão de mobilidade altamente previsível. O uso dessas informações pode auxiliar este processo de decisão.

Dados recentes [10] têm mostrado que em 2016 o número global de dispositivos IoT móveis já alcançava 8 bilhões. Desses dispositivos, 46% eram classificados como dispositivos inteligentes, ou seja, executam aplicações que complementam a experiência do usuário ao utilizar estes dispositivos e, em média, geravam 13 vezes mais dados que os demais dispositivos.

Sistemas relacionados ao aumento de segurança no trânsito como, por exemplo, aplicativos que envolvam visão computacional para identificação de placas de trânsito, poderão se beneficiar desse acesso a uma maior fonte de recursos com uma baixa latência na conexão, requisito necessário devido à frequente atualização dos dados e à necessidade de respostas rápidas às requisições. Aplicativos executados em um *smartphone*, por exemplo, relacionados a jogos ou realidade virtual/aumentada que envolvam uma interação com o usuário também podem obter vantagens ao preferirem executar seu processamento na Névoa.

Aplicativos que não necessitam de uma baixa latência mas demandam um grande poder de processamento, como processamento de vídeo ou aplicações relacionadas à mineração de dados, poderão realizar o *offloading* desse processamento para uma *Cloudlet* ao identificarem essa demanda. Nesse contexto, a prioridade na migração para uma *Cloudlet* em relação a servidores remotos na Nuvem é a possibilidade de diminuir o tráfego de dados no núcleo da rede.

O foco deste trabalho, porém, se dará a casos de uso em que baixa latência e alta demanda de processamento se fazem necessárias. Aplicações baseadas em realidade aumentada sendo executadas em *smartphones* ou dispositivos específicos para estas funções, como óculos interativos que filmam o ambiente e sobrepõem informações relevantes sobre o que está sendo visto pelo usuário. Alguns casos de uso baseados nesse cenário são descritos a seguir:

- *Assistente cognitivo*. Baseado em [21, 22]: Problemas de saúde como Alzheimer, Acidente Vascular Cerebral - AVC, ou lesões causadas por impacto podem comprometer o funcionamento do cérebro para reconhecimento de pessoas, lugares e objetos, ou ainda a recuperação de memória de curto e longo prazo. Dispositivos de realidade aumentada, como proposto por [21], podem auxiliar pessoas portadoras desses problemas e melhorar seu convívio em sociedade de forma independente. Óculos de realidade virtual, por exemplo, poderiam capturar imagens a partir da mesma perspectiva de visão do usuário, interpretá-las, e apresentar para o usuário informações relevantes e em tempo real do que se está sendo observado. Reconhecimento de objetos, textos e faces de pessoas conhecidas pelo usuário, além de uma incorporação de outros possíveis lembretes, como agenda de compromissos, são exemplos de aplicações que auxiliam a interação do usuário com o ambiente ao seu redor. Neste cenário, o dispositivo envia as imagens capturadas para serem processadas na Névoa e, após o tratamento adequado, são enviadas de volta para o dispositivo e apresentadas para o usuário. A Figura 1.1 ilustra um exemplo de funcionamento deste cenário.

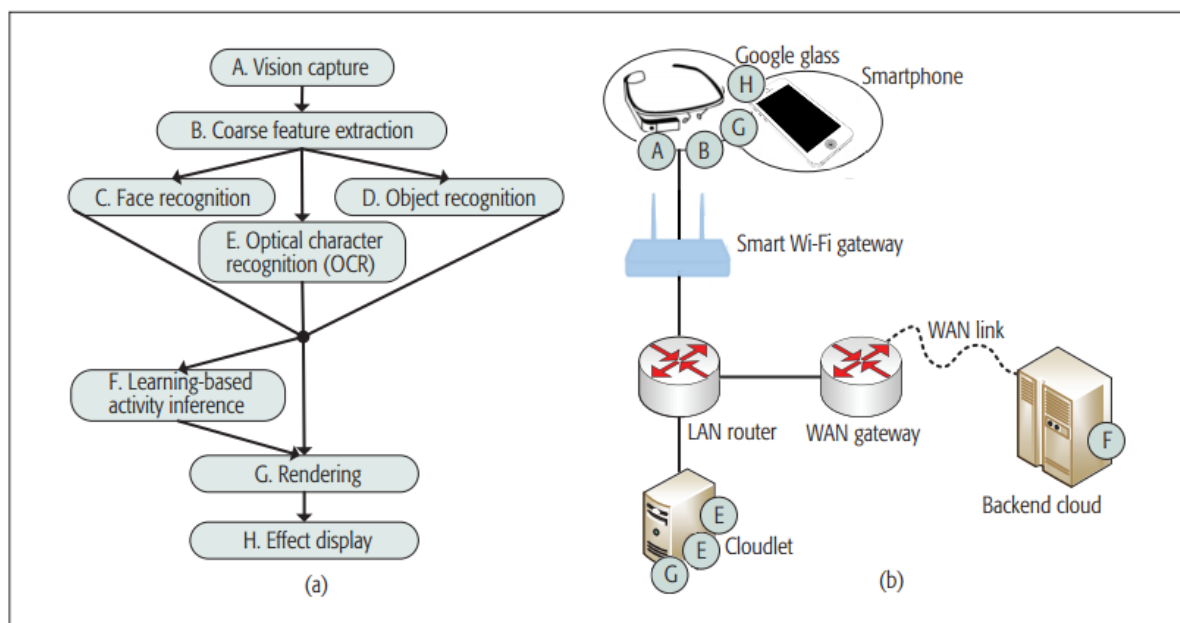


Figura 1.1: Fluxo de execução de uma aplicação para assistência cognitiva realizando *offloading* do processamento para uma *Cloudlet*. Fonte: Adaptado de Jiang, Huang e Tsang (2018) [22]

- *Guia turístico virtual*. Baseado em [31, 39]: Comumente cidades turísticas oferecem roteiros que apresentam monumentos e locais históricos para seus visitantes.

Dispositivos de realidade aumentada poderiam melhorar a experiência do usuário em seu passeio ao interpretar a visão em tempo real do usuário e apresentar sobre a paisagem observada, informações ou mídias relevantes relacionadas ao local. Um turista ao visitar um monumento histórico pode visualizar, sobreposto a paisagem observada, informações como ano de construção e fundador, obter um áudio ou vídeo introdutório à história do local, ou ainda, visualizar em tempo real, uma projeção de como o monumento foi se modificando ao longo de sua história, permitindo visualizar restaurações ou danos. Dado o poder de processamento demandando por essa aplicação, os dispositivos IoT dificilmente seriam capazes de realizar tal processamento e, dada a necessidade de uma rápida interpretação da visão do usuário, o uso da Névoa se faz necessário. Neste cenário, os usuários tipicamente se locomovem em um ônibus turístico ou em veículos de passeio, fazendo necessário que a aplicação do usuário executada na Névoa o acompanhe durante seu trajeto, realizando uma migração transparente ao usuário para garantir uma execução satisfatória da aplicação. A Figura 1.2 ilustra o caso de uso.

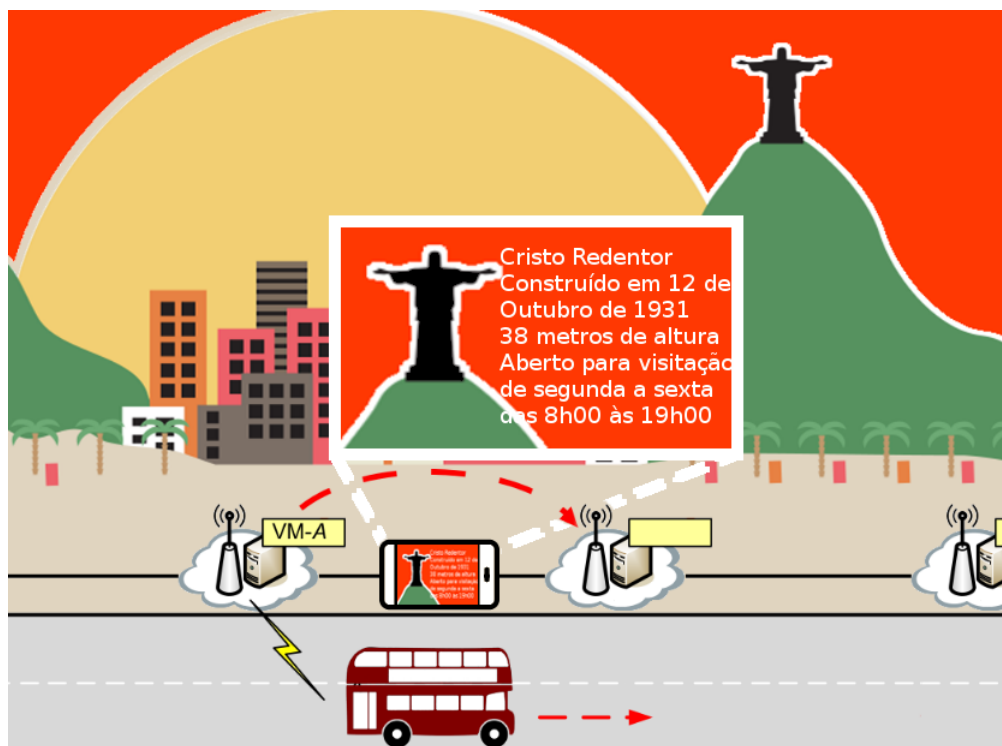


Figura 1.2: Exemplo de cenário de migração em Névoa de um guia turístico virtual

A partir do cenário descrito acima e dos desafios impostos pelos dispositivos IoT móveis, este trabalho tem como objetivo desenvolver mecanismos para otimizar o gerenciamento dos recursos externos utilizados pelos dispositivos IoT móveis. Neste trabalho, o foco de estudo foi o desenvolvimento de políticas e algoritmos para otimizar o processo de migração de máquinas virtuais entre as *Cloudlets* presentes na infraestrutura da Névoa tendo como base usuários com mobilidade altamente predizível. A partir da política de migração proposta baseada no uso da previsão de mobilidade desses usuários, alguns cenários de simulação foram elaborados para a avaliação desta política e do algoritmo de

migração proposto. A Seção 1.1 descreve com detalhes o problema estudado neste trabalho e a Seção 1.2 apresenta as contribuições obtidas durante o desenvolvimento desta dissertação.

1.1 Problema

No contexto de cidades inteligentes, os dispositivos IoT estão inseridos em variados cenários, em alguns deles, o dispositivo pode estar associado a um usuário móvel, como por exemplo, pedestres ou passageiros de carro, metrô ou barco, sendo eles pertencentes aos usuários (celulares, relógios ou óculos) ou embarcados no meio de transporte utilizado (câmeras filmadoras ou sensores de distância). Neste cenário, a realização de um *offloading* para a *cloudlet* mais próxima do usuário em um determinado momento pode não mais atender às suas necessidades após o deslocamento deste usuário durante seu trajeto.

O deslocamento desses usuários introduz uma questão crítica com relação à viabilidade de se executar um *offloading* para servidores da Névoa neste contexto. Para manter a qualidade de serviço oferecida ao usuário, pode-se fazer necessário realizar a migração de sua máquina virtual de um servidor da Névoa para outro melhor localizado. Essa decisão é tomada com base na prioridade entre os denominados custos de migração e custos de execução. O primeiro refere-se ao custo de mover uma VM de uma *Cloudlet* para outra (como por exemplo, volume de dados trafegados entre as *Cloudlets* e processamento necessário por elas para realizar essa migração), enquanto o segundo está relacionado aos custos da comunicação entre o usuário e sua VM (por exemplo, volume de dados trafegados entre o usuário e a *Cloudlet* e a latência oferecida por ela).

Adotar o critério de sempre migrar a VM para a *Cloudlet* mais próxima do usuário a fim de minimizar os custos de execução pode adicionar um custo muito alto à rede caso tal processo necessite transferir um grande volume de dados entre as *Cloudlets*, além de que, apesar de haver técnicas para minimizar o tempo de indisponibilidade de acesso à aplicação durante o processo de migração, a cada vez que esse processo ocorre, o usuário fica sem poder acessar sua aplicação durante um período de tempo. Adotar este critério de muitas migrações aumenta o tempo de indisponibilidade de acesso à aplicação durante o trajeto do usuário.

Por outro lado, adotar outro critério, como não migrar a máquina virtual, removerá todo o custo de migração, porém poderá deixar a VM muito distante do usuário, comprometendo a qualidade da comunicação. Priorizar a minimização de um dos custos em detrimento do outro poderá levar a uma subutilização dos recursos disponibilizados pelas *Cloudlets*. Definir o melhor momento para efetuar a migração da VM, tendo como base as localizações do usuário e de sua máquina virtual e as condições da rede, é uma das principais abordagens para otimizar o uso dos recursos das *Cloudlets*, além de melhorar a experiência do usuário ao utilizar esses aplicativos.

Devido à grande mobilidade característica de alguns dispositivos da Internet das Coisas, como os veículos, bicicletas e trens, a definição de mecanismos para tomada de decisão sobre o momento e o destino ótimos para o processo de migração em servidores na Névoa se torna não trivial, se mantendo como um problema de pesquisa em aberto [31, 6, 34].

O desenvolvimento de novas soluções para gerenciar os recursos computacionais nesse contexto se apresenta necessário [42].

Desenvolver bons mecanismos para o gerenciamento de máquinas virtuais em um ambiente de Computação na Névoa não só tem sido apontado como uma das chaves para possibilitar um "aumento substancial de soluções com suporte à mobilidade, tanto em termos de desempenho quanto de aplicabilidade" [31], como também está entre os desafios para o desenvolvimento da Internet do Futuro para Cidades Inteligentes [4].

Algumas soluções têm sido propostas para manter a máquina virtual no servidor melhor localizado para o usuário a fim de diminuir a latência na comunicação [43, 32, 47]. No entanto, esses trabalhos visam apenas corrigir o posicionamento da máquina virtual após o deslocamento do usuário, atuando de forma reativa.

Considerando os diversos cenários e demandas relacionados a IoT no contexto de Cidades Inteligentes, alguns grupos de usuários, como usuários de ônibus urbanos, metrô e trem, apresentam um padrão de mobilidade altamente previsível. Trabalhos recentes [16, 29] avaliados em outros contextos têm incorporado informações relacionadas a predição de mobilidade desses usuários para otimizar a escolha do destino dos dados dos usuários. Tendo como base dados de mobilidade e informações sobre a localização futura dos usuários, a política de migração de dados proposta por esses autores visa antecipar a migração do conteúdo do usuário para um local que provavelmente fará parte do trajeto do usuário no futuro próximo. Resultados apontaram que as abordagens propostas trouxeram vantagens quanto ao aumento da qualidade de serviço oferecida a esses usuários.

A partir do atual cenário apresentado pelo estado da arte, alguns tópicos de pesquisa se mostram em aberto para o problema de migração de dados e processamento em ambientes de Névoa. Tendo em vista os resultados promissores apresentados em outras áreas ao incorporar informações sobre a localização futura dos usuários, o processo de migração de máquinas virtuais em um ambiente de computação em Névoa também poderia ser avaliado neste contexto.

1.2 Contribuições

Tendo como base o cenário descrito anteriormente, este trabalho propõe uma política de migração de máquinas virtuais baseada no uso de informações sobre o futuro trajeto dos usuários. A migração proativa, como foi definida, baseia-se na utilização de informações sobre a localização futura do usuário para definir o destino de sua máquina virtual ao se requisitar o início do processo de migração. Baseados nessa política, cenários e algoritmos de migração foram propostos e avaliados durante o desenvolvimento do trabalho.

Inicialmente, a literatura não disponibilizava nenhum conjunto de ferramentas para construir e avaliar algoritmos relacionados ao gerenciamento de recursos em um ambiente de Computação em Névoa, como o processo de migração de máquinas virtuais, considerando um cenário de mobilidade compatível com a esperada para usuários em uma cidade inteligente. Tendo em vista este cenário, o desenvolvimento do trabalho deu-se inicialmente pela construção de um ambiente de simulação compatível com o cenário previamente proposto para avaliação.

Este projeto teve, como uma de suas contribuições, uma extensão ao simulador de rede MyIFogSim [27], incorporando o uso de uma rede veicular ao cenário de mobilidade dos usuários. O simulador MyIFogSim disponibiliza um ambiente para avaliar o gerenciamento de recursos computacionais da Computação na Névoa. Baseado no Simulador IFogSim [20], o simulador MyIFogSim estende o simulador anterior suportando usuários móveis no contexto de migração de máquinas virtuais e alocação de recursos nos nós da Névoa.

Baseado no ambiente disponibilizado pelo simulador, foram feitas as modificações necessárias no MyIFogSim a fim de realizar uma integração com o simulador SUMO (Simulation of Urban Mobility) [5]. O SUMO é o simulador responsável pelo gerenciamento da mobilidade de veículos a serem utilizados como usuários no cenário de estudo. A partir de dados de mobilidade disponibilizados na literatura, o simulador SUMO fica responsável por interpretá-los e informar as suas posições para o simulador de rede MyIFogSim. O ambiente construído, bem como as modificações incorporadas ao simulador estão disponíveis para uso da comunidade¹.

A partir do cenário de simulação construído, o trabalho iniciou pela avaliação da política de migração proposta baseada na predição de mobilidade de usuários que apresentavam um padrão de mobilidade altamente predizível. Em um cenário inicial, assegurava-se recursos suficientes para os usuários ao longo de seu trajeto e avaliava o cenário considerando um alto nível de acurácia na predição de mobilidade. Os resultados deste trabalho foram apresentados no Simpósio Brasileiro de Redes de Computadores - SBRC 2018 [17].

Após a construção e avaliação da política proposta, outros cenários mais complexos puderam ser estudados. O processo de migração proativa foi avaliado em um ambiente de concorrência por recursos, no qual um conjunto de usuários se desloca para uma mesma região do mapa. Os recursos computacionais disponíveis nas *Cloudlets* presentes nessa área não são suficientes para atender a demanda dos usuários de maneira ótima, podendo resultar na penalização de alguns usuários ao posicionar suas aplicações em *Cloudlets* mais distantes. Um algoritmo baseado em Programação Linear Inteira foi proposto para otimizar esse processo de distribuição das aplicações dos usuários de forma dinâmica pelas *Cloudlets* considerando o deslocamento dos usuários ao longo do dia. Os resultados foram apresentados no IEEE Symposium of Computers and Communication - ISCC 2018[18]

O restante deste trabalho está organizado da seguinte forma: O Capítulo 2 descreve os conceitos fundamentais para o entendimento do cenário utilizado neste trabalho como os paradigmas de Internet das Coisas, Cidades Inteligentes, Computação em Nuvem e em Névoa. Este capítulo também apresenta mais detalhes acerca do processo de migração de Máquinas Virtuais no contexto de Computação em Névoa. O Capítulo 3 apresenta os principais trabalhos relacionados ao gerenciamento de recursos computacionais em ambientes de Computação em Névoa ou em cenários similares, bem como o atual estado da arte com relação ao processo de migração de máquinas virtuais em Névoa. O Capítulo 4 descreve com mais detalhes o trabalho desenvolvido para a avaliação, em um cenário inicial, da política proativa para a migração de máquinas virtuais proposta neste trabalho. Além disso, este capítulo também detalha o desenvolvimento, bem como a avaliação, do algoritmo proposto para otimizar o processo migração em um ambiente de concorrência

¹<https://github.com/diogomg/MyIFogSim>

de recursos na Névoa baseada na variação dinâmica da demanda dos usuários ao longo do dia. Por fim, o Capítulo 5 sumariza os resultados obtidos durante o desenvolvimento deste trabalho, apresenta as conclusões obtidas e discorre acerca dos tópicos em aberto que podem ser desenvolvidos baseados neste trabalho.

Capítulo 2

Fundamentação Teórica

Esta seção apresenta os conceitos básicos relacionados aos assuntos contidos neste trabalho. A Seção 2.1 descreve o paradigma de Internet das Coisas e sua relação com o desenvolvimento de Cidades Inteligentes. A Subseção 2.1 apresenta os conceitos sobre Computação em Nuvem e Computação em Névoa. A Subseção 2.2 apresenta o conceito de Redes Veiculares e descreve a arquitetura de uma *Vehicular Cloud*. A Subseção 2.3 contém uma visão geral sobre o processo de migração de máquinas virtuais e descreve o funcionamento desse processo em uma *Vehicular Cloud*.

2.1 Internet das Coisas

A incorporação de tecnologias para transmissão de dados nos mais diversos objetos, de celulares e relógios de pulso a veículos e máquinas industriais, tem alterado a experiência de uso desses objetos. Conhecido como Internet das Coisas (Internet of Things - IoT), esse paradigma tem alterado significativamente a utilidade de objetos do cotidiano e permitindo, principalmente, a interação remota entre esses objetos e seus usuários, sejam eles humanos ou outros dispositivos tecnológicos.

Os benefícios trazidos pelas novas funcionalidades incorporadas a esses objetos têm resultado em uma grande popularização desse paradigma, além de impulsionar o desenvolvimento de outros temas como cidades inteligentes. As denominadas cidades inteligentes visam melhorar a qualidade de vida de seus cidadãos principalmente, mas não somente, através de uma infraestrutura tecnológica capaz de oferecer um ambiente integrado, acessível e sustentável para o desenvolvimento das operações do cotidiano de uma cidade [35, 4]. Dentre as aplicações desenvolvidas para cidades inteligentes, pode-se citar controle de tráfego urbano, monitoramento no consumo de água e energia elétrica e deposição de resíduos em lixeiras e aterros, e identificação de rotas para evacuação de pessoas em caso de emergências [1, 35].

A facilidade de incorporação do paradigma de Internet das Coisas no cotidiano das pessoas acaba gerando uma grande inclusão desses dispositivos nos mais diversos contextos. Muitos desses dispositivos podem possuir um contato muito próximo com as pessoas, podendo ser diretamente portados por elas. Os dispositivos *wearables*, como são chamados, possuem diversas formas e aplicações e geralmente são atualizações de objetos já

tradicionalmente portados pelas pessoas, como relógios, pulseiras, óculos, monitores de saúde, entre outros. Estima-se um aumento no número desses dispositivos de 325 milhões em 2016 para 929 milhões em 2021 [10]. Tais dispositivos podem auxiliar as pessoas gerenciando compromissos, ou ainda, contactando de forma autônoma outros agentes, como familiares, polícia ou hospital, em casos de emergência.

Apesar desses dispositivos acompanharem os seres humanos, no contexto de cidades inteligentes, muitos outros dispositivos IoT podem apresentar uma alta mobilidade devido a estarem associados a objetos tipicamente móveis, como em automóveis, bicicletas, trens e navios. Nesse contexto, a Internet das Coisas Móveis (Internet of Mobile Things ou Mobile IoT) restringe o conceito do paradigma de IoT ao referenciar especificamente os dispositivos que não possuem localização fixa.

Pesquisas recentes [10] têm apresentado a grande popularidade desses dispositivos móveis, como ilustrado na Figura 2.1. Em 2016 já havia 8 bilhões de dispositivos móveis conectados e com previsão para superarem a marca de 10 bilhões em 2021. Deste número de dispositivos móveis previsto para 2021, 82% deles podem ser classificados como dispositivos inteligentes que, segundo definição da Cisco [10], possuem capacidades de processamento, multimídia e pelo menos um mecanismo de conectividade para transmissão de dados.

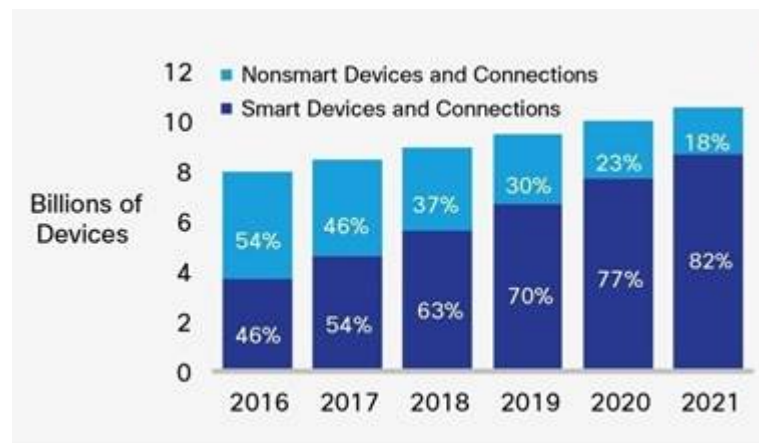


Figura 2.1: Crescimento global do número de dispositivos móveis e conexões. Fonte: Adaptado de V. N. I. Cisco (2016) [10]

Devido aos poucos recursos computacionais oferecidos por tais dispositivos, a execução do processamento, ou pelo menos parte dele, em fontes mais abundantes de recursos como a Nuvem, tem se mostrado uma alternativa nesse contexto. Apesar da facilidade e flexibilidade oferecida por esse ambiente, devido a demandas por baixa latência por parte de algumas aplicações, e a fim de amenizar o grande volume de dados que são transmitidos para o núcleo da rede, alternativas como a Computação em Névoa tem se mostrado como uma opção mais adequada para esse cenário.

2.2 Computação em Nuvem e em Névoa

A demanda por uma otimização do uso de recursos computacionais, a melhoria na qualidade dos meios de transmissão e o barateamento de equipamentos e processamento influenciaram o desenvolvimento de um novo modelo de computação, no qual o acesso a esses recursos é feito remotamente e sob demanda.

O modelo de Computação em Nuvem (*Cloud Computing*) oferece o acesso a um conjunto de recursos, tais como armazenamento, processamento, aplicativos e serviços de forma que a requisição e liberação desses recursos sejam feitas de forma rápida e transparente ao usuário [13].

O acesso ao hardware, geralmente distribuído, oferecido pela *Cloud* é feito principalmente por meio de virtualização. A virtualização é uma tecnologia que abstrai os recursos físicos do hardware e oferece o acesso a eles por meio de máquinas virtuais (*Virtual Machine – VM*). A Figura 2.2 ilustra o conceito de virtualização apresentando a flexibilidade oferecida por esse modelo de abstração para obter acesso aos recursos físicos do *hardware*. A virtualização permite que diferentes sistemas operacionais e processos com características e demandas distintas entre si sejam executadas de forma isolada e independente no *hardware* disponível.

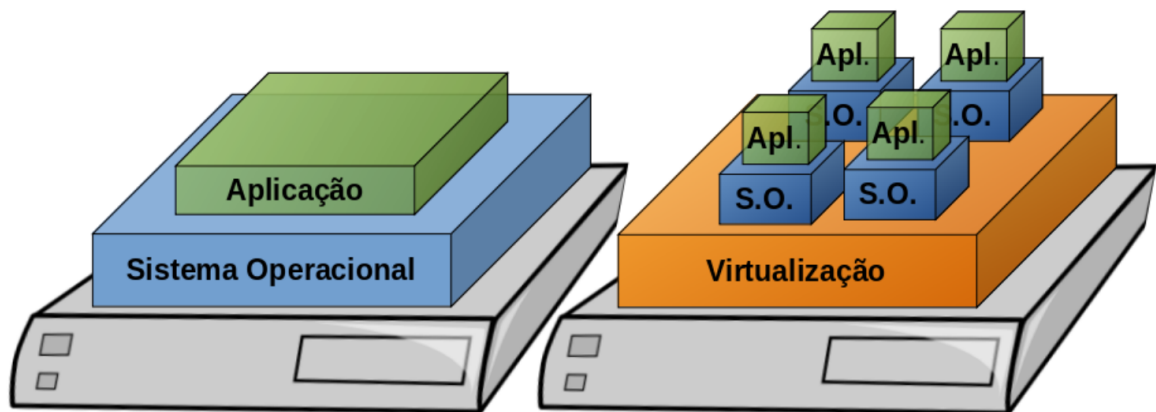


Figura 2.2: Ilustração do conceito de virtualização em servidores. Fonte: Lopes (2017) [26]

Este processo de migração do processamento e armazenamento para servidores na Nuvem permite que aplicações mais robustas possam ser executadas em dispositivos móveis seguindo este mesmo princípio [2], dando origem à Computação Móvel nas Nuvens (*Mobile Cloud Computing - MCC*).

Algumas limitações relacionadas à arquitetura da *Cloud Computing* tais como latência elevada, e perda de pacotes levaram a aproximação de alguns servidores para as bordas da rede como tentativa de amenizar os danos causados a aplicações de tempo real. Satyanarayanan *et al.* [36] apresentam este conceito de *Cloudlet*, servidores próximos aos usuários acessados através de tecnologias sem fio em no máximo um salto na rede. O acesso aos recursos da *Cloudlet*, assim como na *Cloud*, são feitos através de máquinas virtuais.

A popularização de dispositivos que incorporam o acesso à Internet para melhorar sua experiência com o usuário, como eletrodomésticos, veículos e máquinas industriais, classificados como parte da Internet das Coisas (*Internet of Things* – IoT), trouxeram novas demandas relacionadas à Computação em Nuvem.

O baixo poder de processamento e armazenamento desses dispositivos levou a uma demanda por uma fonte de recursos mais abundante, como a Nuvem, para gerenciar um volume mais crítico de dados. A demanda por mobilidade e baixa latência por parte dos dispositivos IoT, no entanto, tornou a comunicação com servidores localizados muito longe desses dispositivos um ponto crítico relacionado ao desempenho dessas aplicações.

Com base nesse cenário, Bonomi *et al.* [7] propuseram o conceito de *Fog Computing*, “uma plataforma altamente virtualizada que provê processamento, armazenamento e serviços de rede entre os dispositivos e os tradicionais *Data Centers* em Nuvem, tipicamente mas não exclusivamente localizada na borda da rede”.

No contexto dos dispositivos de Internet-das-Coisas, os poucos recursos disponíveis nesses dispositivos motivaram a ideia de utilizar recursos externos, como a Nuvem, para executar aplicações mais robustas. No entanto, a conexão com servidores muito distantes do usuário pode comprometer o funcionamento de aplicações que necessitam de respostas rápidas do servidor, como jogos ou aplicações relacionadas a uma interação frequente com o usuário, como aplicações de realidade aumentada/virtual. Para atender à demanda desses aplicativos, servidores menores chamados de *Cloudlets* têm sido posicionados de forma distribuída pela cidade para oferecer um acesso mais próximo aos usuários. No contexto de redes veiculares, o acesso às *Cloudlets* é feito a partir de interfaces via rádio providas por *Roadside Units* - *RSUs*. A Figura 2.3 ilustra uma arquitetura hierárquica entre os dispositivos IoT móveis, Névoa e Nuvem.

Qualquer infraestrutura que ofereça recursos na borda da rede pode ser considerada um nó da Névoa. Roteadores com um elevado poder de processamento e armazenamento poderão se tornar servidores nessa rede, assim como *Access Points*, *Switches*, *Cloudlets* ou até mesmo os próprios dispositivos dos usuários [45]. Essa maior gama de fontes de recursos oferece mais opções próximas aos dispositivos IoT, diminuindo a latência na comunicação e amenizando o tráfego de dados no núcleo da rede.

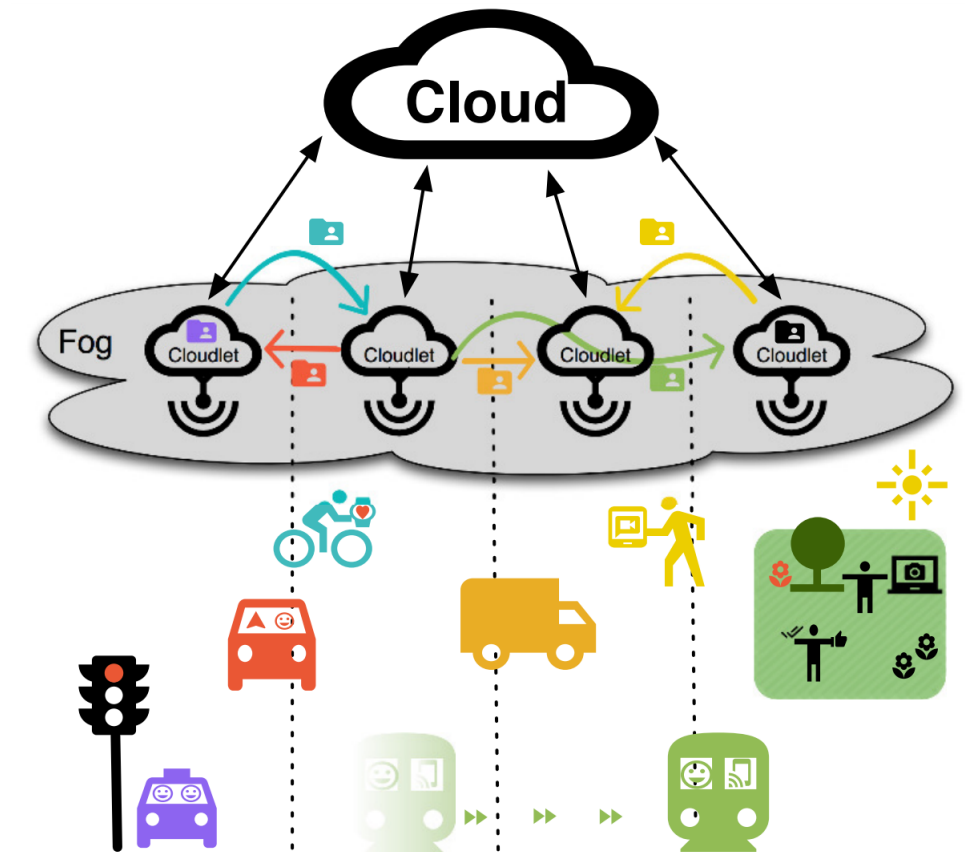


Figura 2.3: Arquitetura de um ambiente de Computação em Névoa para atender usuários móveis. Fonte: Bittencourt *et al.* (2015) [6].

2.3 Migração de Máquinas Virtuais

A migração de máquinas virtuais consiste na transferência da localização física da máquina virtual entre os servidores da rede, sendo eles presentes na Nuvem ou na Névoa, a fim para realizar um balanceamento de carga, manutenção física no servidor ou ajustar a localização da máquina de acordo com a posição do usuário.

Nesse sentido, o conceito de *Follow-me Cloud* (FMC) apresentado por Taleb e Ksentini (2013) [40] propõe que, dada uma rede com um conjunto de pequenos servidores geograficamente distribuídos em sua borda, uma vez que o usuário se movimenta dentro da rede, seu conteúdo presente nos servidores deve segui-lo, mantendo a qualidade e a disponibilidade do acesso a ele.

Estudos presentes na literatura [26] têm demonstrado o comportamento apresentado pela máquina virtual presente na Névoa com relação a diferentes cenários de migração. A Figura 2.4 ilustra uma comparação entre a latência obtida por um usuário móvel adotando duas políticas quanto a escolha da localização da máquina virtual durante seu trajeto. As políticas adotadas foram: (1) manter a máquina virtual na *Cloudlet* de origem escolhida no início do trajeto e manter o acesso à VM de forma indireta através de múltiplos saltos, ou (2) sempre realizar a migração da máquina virtual para a *Cloudlet* mais próxima de acordo com a localização do usuário. Nota-se o aumento acentuado nos níveis de latência apresentados pela abordagem de não migração da máquina virtual com o decorrer do

tempo. A partir de tais trabalhos pode-se perceber a necessidade de se realizar migrações para manter a qualidade da conexão entre o usuário e sua aplicação.

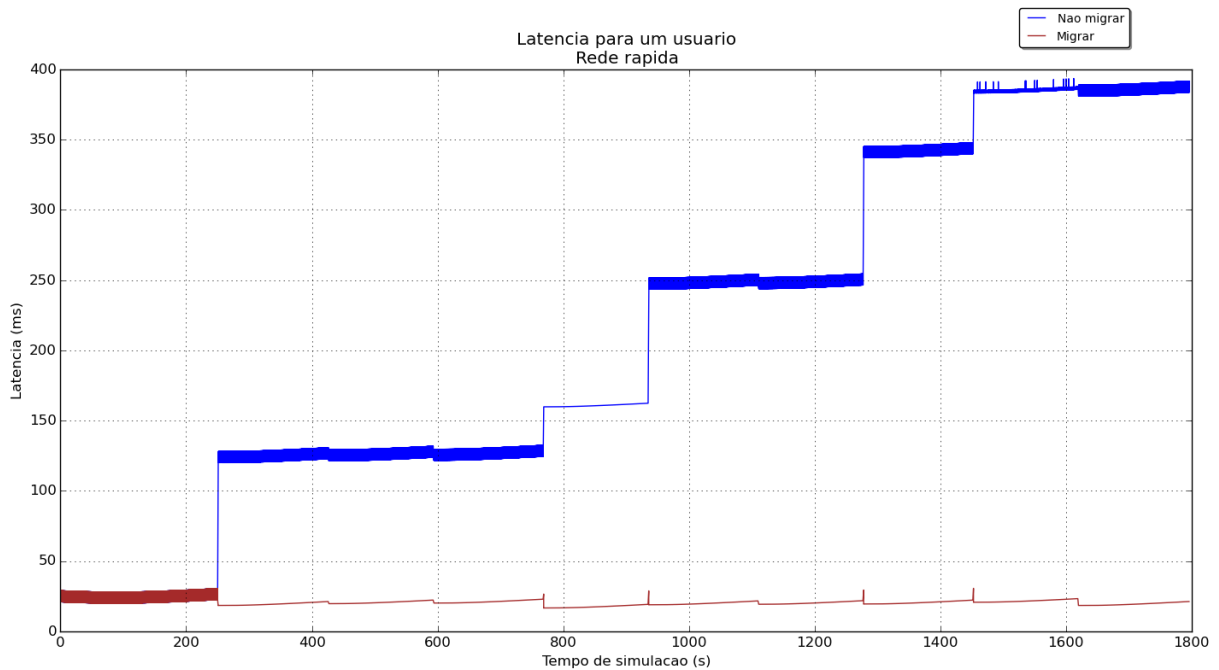


Figura 2.4: Latência obtida por um usuário móvel em cenários de migração e não migração de sua máquina virtual ao longo de seu trajeto. Fonte: Adaptado de Lopes (2017) [26]

Em uma arquitetura hierárquica de Computação em Névoa, como apresentada por Bittencourt *et al.* [6], as máquinas virtuais podem ser migradas entre as três camadas da arquitetura. O processo de migração pode ser classificado em quatro cenários de acordo com a origem e destino da máquina virtual [47]. Tendo como base as definições dadas segundo Yu *et al.*, de forma adaptada, descrevemos os cenários de migração como:

- *Inter-Cloudlet Migration*: Ocorre quando a máquina virtual é migrada de uma *Cloudlet* para outra, ilustrada na Figura 2.5 pelo caso 1. Esta migração ocorre apenas entre os nós presentes na Névoa. Quando o usuário passa a se locomover no mapa, a *Cloudlet* que melhor o atendia passa a se distanciar e comprometer a qualidade da conexão, uma nova *Cloudlet* é escolhida de acordo com a nova localização do usuário e o processo de migração é iniciado entre esses dois nós da Névoa;
- *Across Cloudlet-Cloud Migration*: Esta migração é definida quando a máquina virtual é levada da *Cloudlet* para a Nuvem, ilustrada pelo caso 2 na Figura 2.5. Preferencialmente a máquina virtual deve ser posicionada o mais próximo possível do usuário mas podem ocorrer situações em que esse estado não é possível. Quando o usuário se locomove para uma área sem cobertura de nós de Névoa pode ser necessário migrar seu conteúdo para a Nuvem a fim de manter a execução da aplicação, mesmo que com níveis de latência comprometidos;
- *Across Cloud-Cloudlet Migration*: Esta migração ocorre quando uma máquina virtual que está na Nuvem é transferida do núcleo da rede para uma *Cloudlet*. Esta

migração pode ocorrer quando o usuário passa a se localizar em uma área coberta por *Cloudlets*, passando a ter disponível um ambiente com melhores condições para execução de sua aplicação. É o processo inverso da migração definida como *Across Cloudlet-Cloud Migration*;

- *Across Cloud Migration*: É a transmissão do estado final da máquina virtual da Nuvem ou Névoa para o usuário, como ilustrado no caso 3 da Figura 2.5. Esse processo ocorre quando a utilização de recursos externos não é mais necessária para a execução da aplicação, seja pela diminuição nos recursos requisitados pela aplicação, no aumento dos recursos disponíveis localmente no dispositivo, ou devido ao fim da execução da tarefa requisitada pelo usuário.

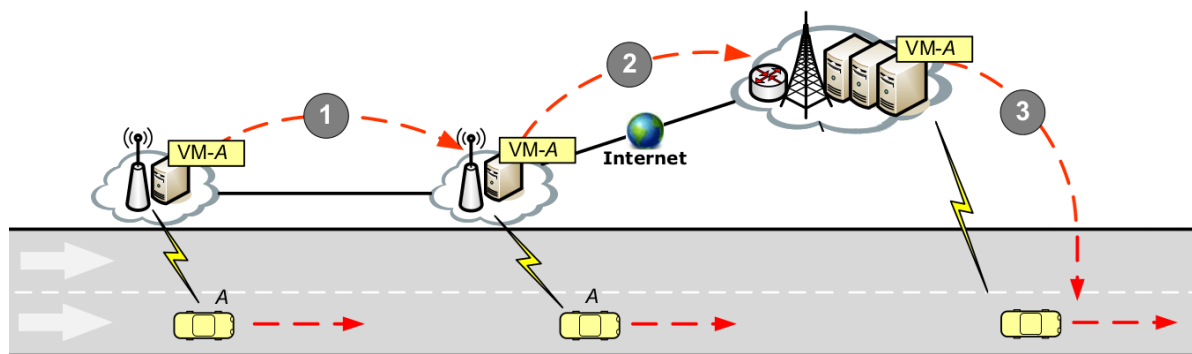


Figura 2.5: Cenários de migração de máquinas virtuais em um ambiente com usuários móveis. Fonte: Adaptado de Yu *et al.* (2013) [47].

Há a possibilidade do usuário ter acesso à sua VM que está na Nuvem ou Névoa de forma indireta. Um usuário ao entrar em uma área fora da cobertura da *Cloudlet* que está executando sua aplicação pode requisitar a migração ou manter a aplicação executando nesse nó da Névoa. Caso a aplicação se mantenha neste nó de origem, o usuário utilizará possivelmente a *Cloudlet* mais próxima como nó de ligação entre ele e sua aplicação. Para conectar a *Cloudlet* de origem e o usuário, poderão haver quantos nós intermediários (ou saltos na conexão) forem necessários. Estes saltos porém, podem aumentar a latência na conexão entre esses dispositivos.

Devido à flexibilidade na escolha da fonte de recursos apresentada pela Névoa, qualquer dispositivo na borda da rede pode prover recursos para os usuários da rede. Neste contexto, um conjunto de dispositivos pode compartilhar de forma distribuída seus recursos de processamento e armazenamento com os demais usuários da rede, formando a denominada Nuvem Móvel (*Mobile Cloud*). Neste cenário um conjunto de dispositivos oferece recursos para o usuário através uma conexão dispositivo-para-dispositivo (*device-to-device* - D2D). Esta abordagem também apresenta uma conexão com múltiplos saltos entre o usuário e sua aplicação, o que pode comprometer a qualidade da conexão.

Capítulo 3

Trabalhos Relacionados

A partir da alta mobilidade de alguns dispositivos IoT usuários da Névoa e a necessidade de atender a alguns requisitos desses usuários, como baixa latência e alta disponibilidade, novos desafios foram inseridos neste contexto. Dado este cenário, desenvolver soluções eficientes para o gerenciamento das aplicações desses usuários na Névoa é necessário para atender aos requisitos dos usuários. A partir de tal demanda, algumas abordagens foram propostas para a gerência dos recursos computacionais presentes na Névoa. Este Capítulo apresenta algumas das soluções disponíveis na literatura relacionadas ao tema de pesquisa deste trabalho.

A Seção 3.1 apresenta propostas de soluções para o problema de migração de máquinas virtuais no contexto de usuários móveis. Esta seção também apresenta alguns trabalhos que incorporaram dados acerca da localização futura do usuário aos seus mecanismos de gerenciamento de recursos computacionais. A Seção 3.2 descreve alguns trabalhos relacionados à otimização da distribuição de processos nos servidores disponíveis na Névoa.

A partir das características e dos objetivos dos trabalhos encontrados na literatura, descritos neste capítulo, pôde-se identificar pontos em aberto no estado da arte. Tais pontos foram explorados neste trabalho de mestrado a fim de gerar contribuições para a área de gerência de recursos em Névoa.

3.1 Migração de máquinas virtuais e predição de mobilidade

O problema de migração de conteúdo de interesse do usuário, sendo este conteúdo composto apenas pelos dados utilizados pela aplicação do usuário ou também pelo processamento relacionado a ela, é um dos desafios para um bom gerenciamento de recursos computacionais para usuários móveis.

No contexto de Computação em Névoa, a maioria das soluções propostas para este problema atuam apenas de forma reativa. Ou seja, estas abordagens apenas reagem ao deslocamento do usuário, iniciando o processo de migração apenas quando alguma métrica utilizada, como número de saltos ou latência, atinge um limiar especificado.

Alguns trabalhos no contexto de Nuvem Móvel (*Mobile Cloud Computing*) e Nuvem Veicular (*Vehicular Cloud Computing*) apresentaram bons resultados ao incorporar uma

predição de mobilidade do usuário a fim de migrar proativamente o conteúdo de interesse do usuário para regiões que farão parte de seu trajeto em um futuro próximo. No contexto de Computação em Névoa, as atuais abordagens ainda não utilizam essas informações sobre o trajeto do usuário para melhorar o processo de migração.

O estado da arte relativo ao processo de migração de máquinas virtuais em um ambiente de Névoa e trabalhos que utilizam informações do futuro trajeto do usuário para otimizar a transferência de dados mas que, até o momento, só foram aplicados em outros contextos, são descritos com mais detalhes em seguida e sumarizados na Tabela 3.1.

Tabela 3.1: Trabalhos relacionados à migração de máquinas virtuais em Névoa ou migração de conteúdo baseado no deslocamento futuro do usuário.

Trabalho	Domínio	Utiliza dados de deslocamento dos usuários	Utiliza predição de mobilidade	Armazenamento \ Processamento
Yao <i>et al.</i> [43]	Nuvem Veicular (VCC) e Cloudlets	-	-	Armazenamento
Refaat <i>et al.</i> [32, 33]	Nuvem Veicular (VCC)	✓	-	Armazenamento
Gomes <i>et al.</i> [16, 14]	LTE	✓	✓	Armazenamento
Gomes <i>et al.</i> [15]	LTE	✓	-	Armazenamento
Zhang <i>et al.</i> [49]	Nuvem Móvel (MCC)	-	-	Processamento
Junior, Henrique e Lopes [23]	Nuvem Móvel (MCC)	-	-	Processamento
Wonkwang, Byoung-Yoon e Dong Ku [37]	Redes Móveis MANET	✓	✓	Armazenamento
Malandrino <i>et al.</i> [28]	Redes Veiculares (VANET)	✓	✓	Armazenamento
Kumar <i>et al.</i> [25]	Nuvem Veicular (VCC)	-	-	Armazenamento e Processamento
Mustafa <i>et al.</i> [29]	Nuvem Veicular (VCC)	✓	✓	Armazenamento e Processamento
Proposta	Cloudlets	✓	✓	Armazenamento e Processamento

Yao *et al.* (2015) [43] apresentam um estudo focado no processo de determinação do momento e destino da migração a fim de minimizar os custos de rede. Assim como nosso estudo, o trabalho citado tem como base a transferência de VMs entre *Vehicular Cloudlets* e realiza o cálculo de otimização de forma individual e a partir da visão do cliente, ou seja, o processo de decisão é feito a fim de minimizar os custos relacionados a cada cliente e não considerando ações que beneficiam um grupo de usuários. Apesar disto, o processo de migração é feito de forma reativa, ou seja, com base no deslocamento que o veículo já realizou, não sendo tomadas decisões com base em alguma previsão de deslocamento.

Apesar de considerar o deslocamento dos veículos, a abordagem apresentada por Refaat, Kantarci e Mouftah (2014) [32] prioriza a migração das máquinas virtuais entre os veículos, utilizando as *Roadside Units* apenas como intermediadoras na migração entre veículos distantes, não se beneficiando de recursos externos que as *Roadside Cloudlets* oferecem. Os veículos nesse contexto são utilizados como fonte de recursos para atender a

uma demanda de processamento da rede. O objetivo do trabalho é propor um *framework* mais eficiente para migração de VMs em situações onde uma topologia heterogênea se altera constantemente.

O estudo apresenta três abordagens para avaliar a escolha do destino da máquina virtual do usuário: *Uniform*, seleciona randomicamente um veículo dentre o conjunto de veículos da rede, não considerando nenhum critério de otimização ou verificação de que o destino está apto a receber a VM; *Least Workload*, seleciona o veículo com recursos suficientes para atender a demanda da VM mais próxima à sua localização inicial; *Mobility Aware* considera o deslocamento do veículo dentro do raio de cobertura da RSU, veículos que ficarão pouco tempo nesta cobertura não possuem prioridade no processo de escolha. Refaat, Kantarci e Mouftah [33] em 2016 complementam o estudo apresentando a quarta abordagem, *Mobility and Destination Workload Aware Migration*, que considera os recursos que o veículo pode oferecer à rede de acordo com seu tempo conectado a ela assim como na abordagem *Mobility Aware*, porém descontando o tempo necessário para a migração do conteúdo entre os veículos envolvidos. Considerando a mobilidade dos usuários, as duas últimas abordagens apresentaram os melhores resultados no trabalho, aumentando o número de VMs migradas com sucesso além de terem demandado menos recursos da rede.

Os trabalhos [16, 15, 14] avaliam a migração das máquinas virtuais considerando possíveis destinos dos veículos, porém o foco desses trabalhos está na seleção de dados para *caching*. Com isso o processo de migração visa otimizar a localização do conteúdo a partir de uma visão global dos dados, tomando decisões que beneficiem um grupo que estará acessando o conteúdo.

Gomes *et al.* (2016) [15] propõem um modelo baseado em *Mobile Follow-me Cloud* para migração de máquinas virtuais. Devido à mobilidade dos usuários, o estudo utiliza uma abordagem para a migração de máquinas virtuais de acordo com os conteúdos populares a certos grupos de usuários. Diferentemente da nossa proposta, o processo de decisão apresentado por Gomes *et al.* é feito baseado na otimização global da rede.

Gomes, Braun e Monteiro (2016) [14] apresentam uma estratégia para migração de máquinas virtuais baseada na previsão de deslocamento dos usuários. A abordagem considera probabilidades de destinos para cada usuário individualmente além de dados relacionados aos destinos da VM como espaço disponível, número de usuários conectados e o custo da migração. Apesar de cada usuário calcular os possíveis destinos para o conteúdo que ele está acessando, o servidor responsável pelo processo de migração decide pela movimentação dos dados apenas quando um grupo de usuários requisita a migração.

O trabalho [16] apresenta uma arquitetura baseada em *Mobile Cloud* com o objetivo de otimizar a utilização do cache por parte dos usuários. O estudo combina os resultados de [15, 14] para desenvolver sua arquitetura.

Apesar de muitos estudos focarem na otimização da migração de máquinas virtuais baseadas no armazenamento dos dados, alguns estudos como [49] têm como objetivo decidir pela migração ou não da máquina virtual a fim aproveitar os recursos oferecidos pela *Cloud* desde que o custo da migração não comprometa o processo.

Malandrino *et al.* (2014) [28] apresentam um modelo para a definição das fontes do conteúdo de interesse dos usuários baseado nos recursos disponíveis na sua localização.

Com base nos dados de mobilidade e conteúdo consumido pelos usuários, as RSUs selecionam as melhores fontes para o *download* de conteúdo de interesse dos usuários e quais partes do conteúdo estarão disponibilizadas nessa fonte. A partir de uma predição da mobilidade do usuário, a continuidade do conteúdo utilizado por ele pode ser preparado em fontes próximas que deverão ser utilizadas no futuro. O modelo assume RSUs e veículos como fontes de conteúdo, em casos em que essas fontes não estão disponíveis, a conexão é feita diretamente com servidores da *Cloud* no núcleo da rede através de conexões celulares.

Apesar de também utilizar mecanismos de predição para otimizar o gerenciamento dos recursos utilizados pelo usuário, considera apenas dados como fonte de interesse do usuário, o trabalho desenvolvido nesta dissertação também considera aplicações que demandam poder de processamento da névoa. Devido a esse fato, o trabalho citado não considera métricas importantes para a avaliação da execução de aplicações na borda da rede, como latência e tempo de indisponibilidade de acesso.

Outra diferença apresentada pelo trabalho citado se dá pelo uso da Nuvem como fonte dos dados, enquanto no trabalho desenvolvido nessa dissertação a aplicação do usuário é migrada apenas entre as *Cloudlets*, não considerando a Nuvem como fonte de dados. Há de se considerar também que o trabalho citado utiliza RSUs e outros veículos da rede como fonte externa de recursos para os usuários, enquanto o trabalho desenvolvido nesta dissertação utiliza dispositivos dedicados (*Cloudlets*) como local para execução das aplicações.

O trabalho proposto por Wonkwang, Byoung-Yoon e Dong Ku (2015) [37] apresenta uma abordagem para utilização de potenciais caches móveis, como veículos, para transportar dados solicitados por usuários de dispositivos móveis, aumentando a taxa de transferência da rede e diminuindo o uso de recursos da rede celular. Após a solicitação de um dado da *Cloud*, o sistema transfere os dados para bases fixas na via e em seguida para um veículo que possui uma rota que passa pela localidade do usuário, permitindo que o usuário receba seus dados através de uma fonte mais próxima. Apesar dos dados serem solicitados de acordo com os interesses individuais dos usuários, o caminho que esses dados irão percorrer e a prioridade em que serão atendidos são definidos com o objetivo de minimizar os custos globais da rede.

A fim de otimizar a distribuição de recursos das *Cloudlets*, Neeraj *et al.* (2015) [25] modelam este problema utilizando Teoria dos Jogos, representando os veículos como jogadores neste jogo, cada um disputando o acesso aos recursos da *Cloud*. A migração das máquinas virtuais neste trabalho tem como objetivo balancear a demanda de recursos sobre as *Cloudlets*. Além da diminuição da energia consumida pela infraestrutura, o trabalho também avaliou a abordagem considerando as mesmas métricas a serem utilizadas nessa proposta, apresentando bons resultados com relação à diminuição de latência e no tempo total de execução das aplicações. Esse trabalho porém, não utilizou quaisquer inferências acerca da localização futura do usuário para tentar otimizar a escolha dos destinos das máquinas virtuais.

Junior, Henrique e Lopes (2016) [23] apresentam uma avaliação de desempenho na execução de aplicativos de *smartphones* em *Mobile Cloud Computing*. O trabalho compara o tempo de execução de aplicativos relacionados ao processamento de imagem utilizando três fontes de recursos: *hardware* local do dispositivo, *Cloudlets* e *Cloud*. O estudo mostra

que mesmo com pequenas demandas de processamento, a execução na *Cloud* e principalmente em *Cloudlets* se mostram mais eficientes que a execução local.

O trabalho apresentado em [29] propõe a utilização de uma predição de mobilidade do trajeto do usuário para otimizar o processo de migração de máquinas virtuais em *Vehicular Cloud*. Apesar de também explorar informações sobre o destino do usuário, a principal fonte de recursos para realizar o processamento requisitado pelo usuário são os próprios veículos que compõem a rede. No escopo do trabalho, as *Roadside Units* são utilizadas apenas como intermediárias no processo de migração das máquinas entre os veículos, não fazendo uso efetivo dos recursos disponíveis na Névoa. O trabalho sugere que a utilização de informações do futuro trajeto do veículo melhoram a qualidade de serviço oferecida ao usuário, apesar de utilizar outras métricas para realizar essa afirmação, como número de tarefas requisitadas pelo usuário executadas com sucesso, e não utilizando outros critérios de avaliação essenciais no contexto de Computação em Nuvem, como latência e disponibilidade de acesso.

Dado o estado atual da arte apresentado pelos trabalhos citados anteriormente, este trabalho de mestrado tem como objetivo incorporar métodos de predição de deslocamento dos veículos em vias que oferecem acesso a *Cloudlets* aos procedimentos relacionados à *Follow-me Cloud* aplicados nessas redes. A migração das máquinas virtuais vinculadas aos usuários será realizada proativamente para *Cloudlets* que serão utilizadas futuramente pelo usuário de acordo com o seu trajeto, ao contrário das atuais soluções que apenas corrigem o posicionamento das máquinas virtuais quando sua localização começa a apresentar problemas de comunicação.

3.2 Localização de aplicações

Apesar de alguns trabalhos considerarem uma visão global da rede para definir o destino das aplicações dos usuários pelos nós da rede, ainda há poucos trabalhos que relacionam o processo de migração com mecanismos para otimizar a distribuição das aplicações pela rede. A utilização de uma visão global da rede pode contribuir para otimizar o gerenciamento de recursos da rede, seja para não realizar um balanceamento de carga por questões energéticas, ou para uma distribuição mais justa dos recursos entre os usuários. Alguns trabalhos tem como objetivo otimizar essa localização das aplicações (*Service Placement*) nas *Cloudlets* da Névoa.

Os autores apresentam em [38] uma abordagem para solucionar o problema de alocação de aplicações em ambientes de Computação em Névoa. Tendo como base a demanda de recursos de dispositivos da Internet das Coisas, o trabalho analisa o cenário e propõe um algoritmo baseado em Programação Linear Inteira para solucionar o problema. O simulador IFogSim foi utilizado para a validação dos experimentos. Apesar de considerar a mobilidade dos dispositivos e também ter como objetivo otimizar a distribuição de demanda pelos nós da Névoa, o trabalho não utiliza nenhuma informação relacionada à predição de mobilidade para otimizar sua proposta.

O trabalho proposto em [41] apresenta uma abordagem para otimizar a gerência de recursos da Névoa a fim de minimizar a latência entre o usuário e sua aplicação. A

heurística utilizada por esse trabalho para alcançar este objetivo foi minimizar o número de saltos entre o usuário e sua aplicação na Névoa. A partir de um algoritmo proposto baseado em Programação Linear Inteira, o trabalho procura definir a melhor distribuição das aplicações entre os nós da Névoa de acordo com a localização dos usuários. Assim como o projeto proposto neste documento, o trabalho proposto por Velasquez *et al.* também visa melhorar o desempenho da aplicação minimizando a latência de acesso oferecida ao usuário. Entretanto, o trabalho não utiliza nenhum dado acerca da posição futura do usuário nem realiza algum tipo de validação de sua proposta em um cenário de alta mobilidade dos usuários.

O trabalho apresentado por Klaimi, Senouci e Messous (2018) [24] propõe um modelo baseado em Teoria de Jogos para realizar a alocação das aplicações de usuários móveis na Névoa. Assim como neste trabalho de mestrado, o trabalho apresentado em [24] utiliza padrões de mobilidade de veículos para definir o deslocamento dos usuários na rede, bem como sua demanda por migrações. Esse trabalho também visa otimizar a localização das aplicações a fim de garantir a qualidade de serviço requerida pelos usuários. Assim como os demais trabalhos encontrados na literatura, essa solução proposta por Klaimi, Senouci e Messous não utiliza quaisquer informações acerca da localização futura dos usuários para otimizar seu mecanismo de migração e alocação de aplicações na Névoa.

Baseado no estado atual da arte, este trabalho de mestrado também visa avaliar a abordagem de migração proativa a partir de uma visão global do estado da rede. A partir dessa perspectiva será possível propor otimizações na distribuição das aplicações pela Névoa de forma mais justa entre os usuários.

Tabela 3.2: Trabalhos relacionados à alocação e distribuição de aplicações (*Service Placement*) em Névoa.

Trabalho	Domínio	Objetivo		Validação	Ferramental
		Minimizar	Maximizar		
Kumar <i>et al.</i> [25]	Nuvem Veicular (VCC)	Consumo de energia	-	SUMO	Teoria de Jogos
Skarlat <i>et al.</i> [38]	Névoa	-	Requisições prioritárias	IFogSim	Programação Linear Inteira
Velasquez <i>et al.</i> [41]	Névoa	Saltos e migrações	-	Não há	Programação Linear Inteira
klaimi <i>et at.</i> [24]	Nuvem Veicular (VCC) e Nuvem	Processamento, consumo de energia e processamento enviado à Nuvem	Requisições atendidas	MATLAB	Teoria de Jogos
Proposta	Cloudlets	Latência	Requisições atendidas	IFogSim	Programação Linear Inteira

Capítulo 4

Migração Proativa de Máquinas Virtuais

A utilização de recursos externos como a Névoa possibilitou que dispositivos IoT pudessem executar aplicações mais robustas, principalmente mas não exclusivamente, graças à baixa latência oferecida por essa infraestrutura. A mobilidade de alguns usuários, porém trouxe importantes desafios para o gerenciamento dos recursos computacionais da Névoa, visto que um Nó da Névoa pode não mais atender a qualidade de serviço demandada pelo usuário após ele se deslocar pelo mapa.

No cenário de cidades inteligentes, a mobilidade de alguns usuários, como passageiros de ônibus urbanos, pode apresentar um padrão de mobilidade altamente previsível. A disponibilidade de tais informações, até então, não havia sido utilizada pela literatura para otimizar o processo de migração de máquinas virtuais de usuários móveis em um ambiente de Computação em Névoa.

A partir do cenário descrito no Capítulo 3 acerca das atuais políticas de migração de máquinas virtuais em ambientes de Computação em Névoa, e dos benefícios que aplicações em outros contextos obtiveram ao incorporar informações sobre o trajeto futuro de seus usuários às suas políticas de gerenciamento de recursos, este trabalho visa contribuir com um melhor gerenciamento de recursos computacionais da Névoa baseado nestas duas constatações.

Este trabalho propõe a incorporação de predição de mobilidade do usuário ao processo de alocação de recursos em Névoa, adicionando dados relativos à posição futura do usuário aos critérios utilizados para a escolha do local de destino de um pedido de migração de máquina virtual.

Neste trabalho, a política de migração proposta foi avaliada considerando dois cenários compatíveis com o contexto de uma cidade inteligente. Em ambos os cenários, são utilizados grupos de usuários que apresentam um padrão de mobilidade altamente previsível. A escolha de usuários de ônibus urbanos para atender a esta característica deu-se para garantir que a avaliação da hipótese não seria prejudicada pela acurácia dos algoritmos de predição.

O primeiro cenário avaliado, apresentado na Seção 4.3 avalia a política de migração em um cenário otimista. Neste cenário a predição de mobilidade possui uma acurácia de 100% e as *Cloudlets* que oferecem a melhor latência ao usuário possuem garantia de

disponibilidade de recursos. O objetivo desta avaliação é expor a hipótese ao melhor cenário possível. O acréscimo de restrições ao cenário poderia mitigar o desempenho da política proposta. Com o acréscimo gradual de restrições, poderá se identificar o impacto de cada uma ao gerenciamento de recursos da Névoa.

Baseado na avaliação do Cenário 1, a Seção 4.4 apresenta o Cenário 2, que estende o cenário anterior removendo a garantia de disponibilidade de recursos por parte da Infraestrutura da Névoa. Neste cenário, há uma concorrência por recursos por parte dos usuários. A partir disto, uma *Cloudlet* que disponibiliza a melhor qualidade de serviço a um usuário pode não possuir recursos disponíveis para atendê-lo.

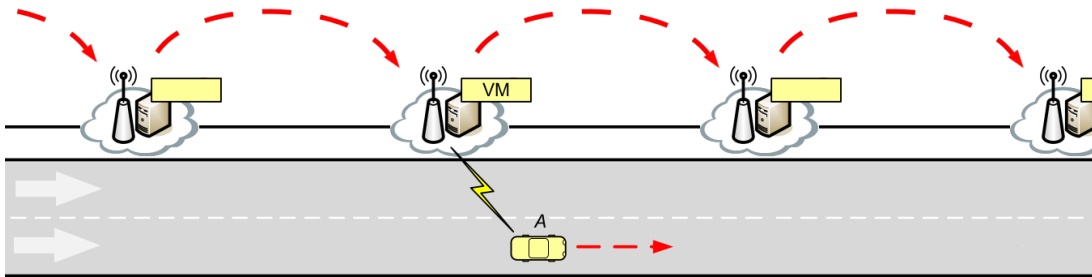
A partir do objetivo geral deste trabalho, este capítulo visa descrever com detalhes a abordagem proposta e os cenários utilizados para sua validação. A organização deste capítulo se apresenta da seguinte forma: A Seção 4.1 apresenta a política de migração proposta, descrevendo o processo de migração auxiliado pelo uso de predição de mobilidade do usuário. A Seção 4.2 descreve a arquitetura lógica proposta para atender aos requisitos do cenário avaliado, migração em Névoa a partir da mobilidade do usuário. A Seção 4.3 descreve a avaliação do modelo de migração proposto considerando um cenário propício para a migração da máquina virtual do usuário ao longo de seu trajeto. Neste cenário inicial foram considerados usuários que possuem uma mobilidade altamente predizível, além de uma garantia de disponibilidade de recursos por parte da infraestrutura da Névoa. Por fim, a Seção 4.4 apresenta a avaliação da política proposta a partir de um cenário com concorrência por recursos. Neste cenário, um grupo de usuários concorrem por recursos limitados de determinada região do mapa. Um algoritmo baseado em Programação Linear Inteira foi proposto para otimizar a alocação das máquinas virtuais nas *Cloudlets*.

4.1 Política de migração proposta

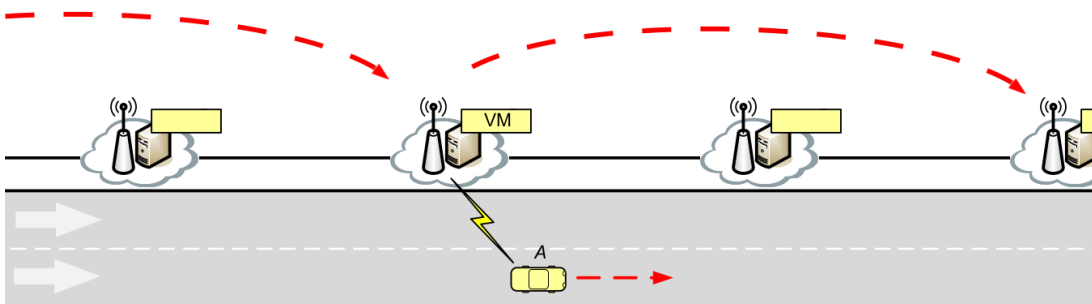
As atuais abordagens presentes na literatura, definidas neste trabalho como abordagens reativas, são caracterizadas por iniciarem seu processo de migração apenas quando a latência ou outra métrica utilizada atinge um limiar delimitado e por escolherem o destino da máquina virtual baseada na localização atual do usuário. Estas políticas de migração de VMs podem escolher o destino das máquinas virtuais com base na menor distância física ou latência imediata apresentada entre o usuário e a *Cloudlet*. Apesar desse tipo de política geralmente oferecer a menor latência possível para o usuário em um primeiro momento, pouco tempo após a migração o usuário passará a se distanciar novamente do seu conteúdo, antecipando a necessidade de se realizar uma nova migração.

Diferentemente das soluções atuais, apresentadas no Capítulo 3, este trabalho propõe realizar a migração para um local à frente no caminho do usuário. Dessa forma, em um primeiro momento, o usuário passará a se aproximar do seu conteúdo e, só após algum tempo, ele passará a se distanciar novamente até que necessite repetir o processo de migração. A Figura 4.1 ilustra o comportamento dos processos de migração utilizando as duas políticas avaliadas, reativa e proativa. A Figura 4.1(a) exemplifica o processo de migração reativa, selecionando como destino das máquinas virtuais sempre a *Cloudlet*

melhor localizada (geralmente a mais próxima do usuário) no momento em uma migração é necessária. No processo de migração proativa, ilustrado na Figura 4.1(b), o destino da máquina virtual se encontra, preferencialmente, em uma *Cloudlet* que está próxima ao trajeto futuro do usuário, não sendo necessariamente a mais próxima no momento em que o processo de migração se inicia.



(a) Abordagem reativa. A *Cloudlet* mais próxima do usuário é escolhida para o destino da máquina virtual



(b) Abordagem proativa. *Cloudlet* a frente do usuário é escolhida para o destino da máquina virtual

Figura 4.1: Ilustração da sequência de migrações em um cenário de migrações reativas e proativas. Fonte: Adaptado de Yu *et. al* (2013) [46].

A partir do momento em que o sistema detecta a necessidade de se realizar uma nova migração para corrigir a localização da VM com relação à atual posição do usuário, a política proposta irá selecionar as *Cloudlets* que estão disponíveis na área de interesse do usuário nos próximos minutos. A partir do conjunto de *Cloudlets* selecionadas, o destino da máquina virtual será escolhido de acordo com a menor latência apresentada para aquela área, respeitando os recursos disponíveis pelas *Cloudlets*. A migração da máquina virtual sempre ocorrerá a partir do momento em que *Cloudlet* atual deixar de apresentar a menor latência para o usuário comparada com o conjunto de *Cloudlets* selecionadas. Esta verificação é realizada periodicamente a cada 1 segundo pelo simulador.

A Figura 4.2 ilustra um exemplo de migração de máquina virtual ao longo do caminho de um usuário utilizando a política proposta. Ao identificar a necessidade de se realizar uma migração, inicia-se o processo para calcular a localização futura do usuário (indicada

pela linha tracejada) com base no período de predição que está sendo utilizado. A partir da localização futura, uma área de interesse do usuário é criada e todas as *Cloudlets* presentes nesta área se tornam candidatas a receber a máquina virtual do usuário.

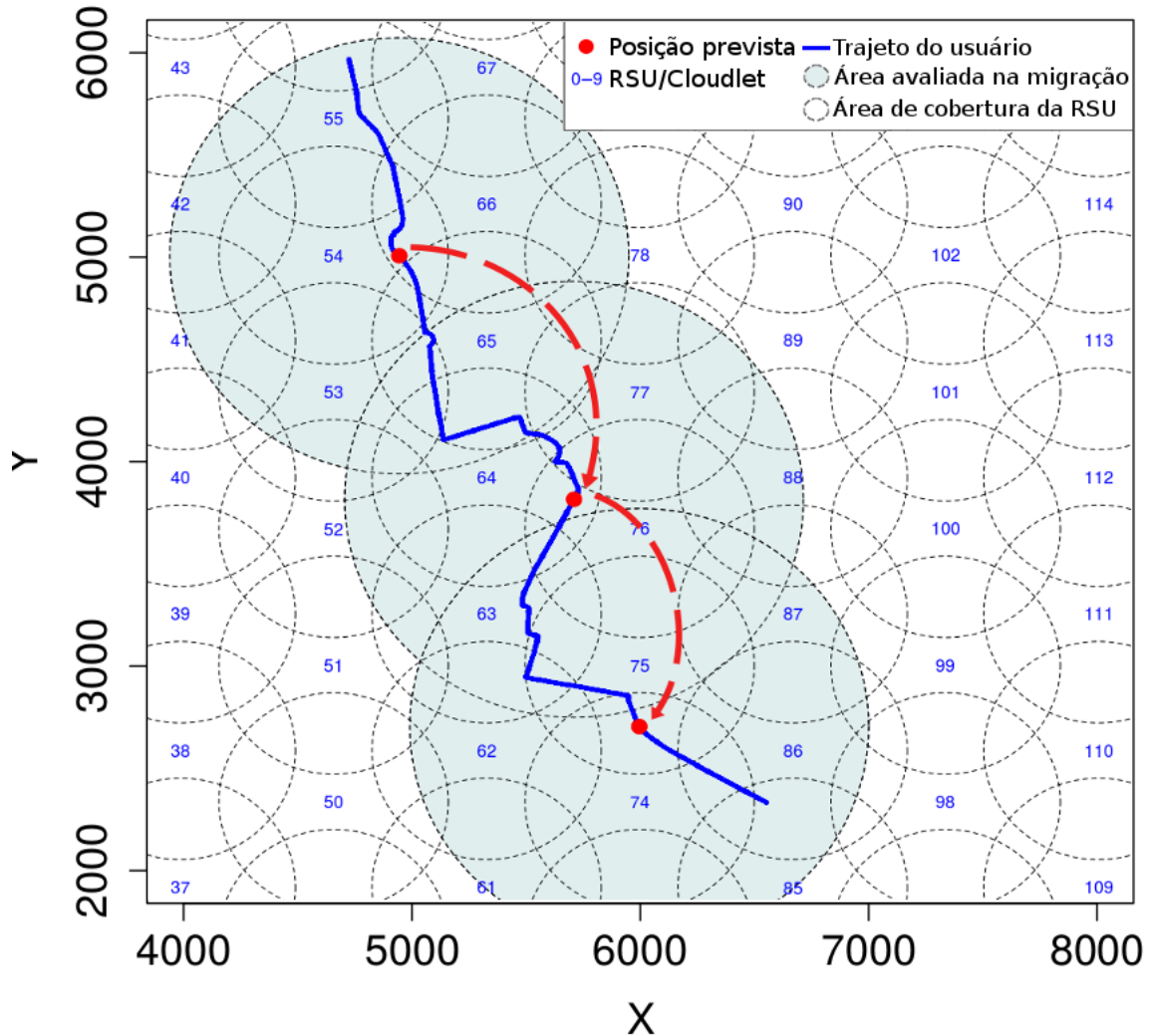


Figura 4.2: Ilustração do processo de migração proativa entre os nós da Névoa.

O destino do conteúdo do usuário deve ser escolhido, preferencialmente, tão à frente no caminho do usuário quanto possível, desde que a distância entre o usuário e a *Cloudlet* não comprometa os níveis de latência exigidos pelas aplicações. A minimização no número de migrações está relacionada à maximização do período de tempo em que uma *Cloudlet* consegue atender as necessidades do usuário.

O tempo em que o usuário manter-se-á conectado à *Cloudlet* pode ser otimizado escolhendo o servidor melhor localizado de acordo com seu trajeto nos próximos minutos. Quanto maior o conhecimento sobre o trajeto futuro do usuário, melhores são os dados disponíveis para escolher o destino para a aplicação. O intervalo de tempo futuro em que se tem conhecimento da localização do usuário é referenciado neste trabalho como *tempo de predição*.

A partir da hipótese proposta e da demanda de gerenciamento imposta por esse cenário, uma arquitetura lógica foi proposta a fim de oferecer os componentes necessários

para gerenciar os recursos da Névoa considerando a mobilidade dos usuários dessa infraestrutura. A arquitetura proposta apresenta os módulos necessários para a utilização de predição de mobilidade como parte dos critérios para escolha do destino das máquinas virtuais.

4.2 Arquitetura

A partir dos elementos presentes na arquitetura física de um ambiente de Computação em Névoa, como dispositivos IoT e *Cloudlets*, uma arquitetura lógica composta por um conjunto de módulos e funções se faz necessária para gerenciar esse ambiente. A arquitetura utilizada neste contexto precisa prover suporte para a mobilidade dos usuários, bem como possuir mecanismos inteligentes o suficiente para gerenciar a migração dos dados e processamento das aplicações, dada essa dinamicidade na mobilidade dos usuários.

A fim de prover suporte para a avaliação da proposta deste trabalho, a arquitetura lógica para a Névoa também deve possuir mecanismos para monitorar e prever a localização do usuário conectado à infraestrutura física da Névoa. Tais informações devem ser utilizadas pelos módulos de gerência de recursos da Névoa.

A arquitetura lógica proposta para este trabalho, ilustrada na Figura 4.3, visa oferecer uma gerência de forma inteligente do processo de migração dos aplicativos de dispositivos IoT para fontes externas de recursos, como as *Cloudlets*, e entre esses servidores de acordo com o deslocamento dos usuários.

Apesar da arquitetura desenvolvida apresentar algumas propostas de contribuições, se comparada às demais arquiteturas relacionadas presentes na literatura, o foco deste trabalho não está na avaliação desta arquitetura. Mais detalhes acerca da arquitetura, principalmente relacionados às propostas de contribuições apresentadas por ela como, por exemplo, o suporte a mobilidade e predição do trajeto dos usuários a fim de otimizar o gerenciamento dos recursos da névoa, podem ser consultados em Velasquez *et. at.* (2007) [42].

Esta arquitetura tem como principal objetivo especificar o escopo e os requisitos mínimos para a implementação deste trabalho. A partir dos módulos propostos, a arquitetura é utilizada como base para o desenvolvimento desta dissertação.

Neste cenário, dentre os módulos propostos, o módulo denominado *Cloudlet Controller*, presente na *Cloudlet*, realiza o gerenciamento dos recursos da Névoa de acordo com a oferta e demanda de recursos presentes nos diversos cenários.

Os módulos presentes no dispositivo IoT são responsáveis pelo processo de migração de aplicativos para as *Cloudlets*. Os aplicativos executados no dispositivo IoT são monitorados e, de acordo com a demanda por mais recursos, poderão ser migrados para uma *Cloudlet*. O dispositivo IoT identifica as *Cloudlets* próximas e estabelece a conexão com uma delas. A partir da demanda por recursos por parte do aplicativo e da resposta das *Cloudlets* informando os recursos disponíveis, o dispositivo IoT deverá selecionar quais tarefas deverão ser migradas para a *Cloudlet*.

Ao realizar o *offloading* de um aplicativo para uma *Cloudlet*, o dispositivo IoT permite que esse servidor colete informações a respeito do seu deslocamento a fim de contribuir

com decisões relacionadas à migração do aplicativo. Devido à mobilidade dos usuários, a aplicação deverá, preferencialmente, ser executada nas *Cloudlets* que ofereçam as melhores condições para o usuário no decorrer do seu trajeto, podendo ser necessária a migração de um aplicativo de uma *Cloudlet* para outra que oferecerá uma melhor condição de execução em um futuro próximo.

A arquitetura da *Cloudlet* possui componentes para o monitoramento do usuário e de *Cloudlets* próximas, contabilizando informações relativas à mobilidade do usuário e à qualidade da conexão com ele, dados referentes à conexão com outras *Cloudlets* e aos recursos disponíveis por elas. A partir dos dados do usuário, a *Cloudlet* poderá prever a sua trajetória e, combinado com dados referentes às demais *Cloudlets*, poderá requisitar a migração da aplicação para um local mais adequado.

A partir destes pedidos de migração, o módulo *Cloudlet Controller* presente na arquitetura da Névoa será o responsável por definir qual o próximo destino da aplicação que oferecerá a melhor condição de execução em um futuro próximo. Este gerenciamento do processo de migração deve ser feito de forma a otimizar o uso dos recursos da Névoa, aumentando o número de clientes atendidos e garantindo a qualidade de serviço exigida por eles.

Os módulos presentes no dispositivo IoT e nas *Cloudlets*, ilustrados na Figura 4.3, são descritos individualmente abaixo.

Arquitetura do dispositivo IoT: A arquitetura localizada no dispositivo IoT é responsável pelo gerenciamento e definição do local de execução dos aplicativos. O processo de *offloading* do aplicativo é avaliado de acordo com sua demanda e das fontes de recursos disponíveis. Os componentes necessários para a execução desse processo estão listados abaixo.

- *Communication Manager*: Módulo responsável por identificar a presença das *Cloudlets* próximas ao usuário e estabelecer a comunicação entre o dispositivo e esses nós da Névoa;
- *Resource Monitor*: Responsável por coletar dados referentes aos recursos disponíveis na *Cloudlet* e à qualidade da conexão entre o usuário e esses servidores;
- *Offloading Manager*: Baseado nos recursos disponibilizados pelas *Cloudlets* e nos dados coletados pelo *Benchmark Monitor*, esse módulo decide pela realização, ou não, do *offloading* da aplicação para a *Cloudlet*; é o responsável por definir quais módulos do aplicativo serão migrados para a *Cloudlet*, qual *Cloudlet* irá executar esses módulos e quando esse *offloading* irá ocorrer;
 - *Benchmark Monitor*: Monitora os módulos selecionados pelo *Code Partition*, coletando e mensurando o desempenho da aplicação que está sendo executada;
- *Code Partition*: Define por meio de regras estáticas ou dinâmicas quais módulos do aplicativo poderão ser executados remotamente;
- *Virtual Machine*: Virtualização dos recursos físicos disponíveis no dispositivo IoT, permitindo a execução dos aplicativos em um ambiente de fácil gerenciamento dos recursos e de fácil migração no local de execução.

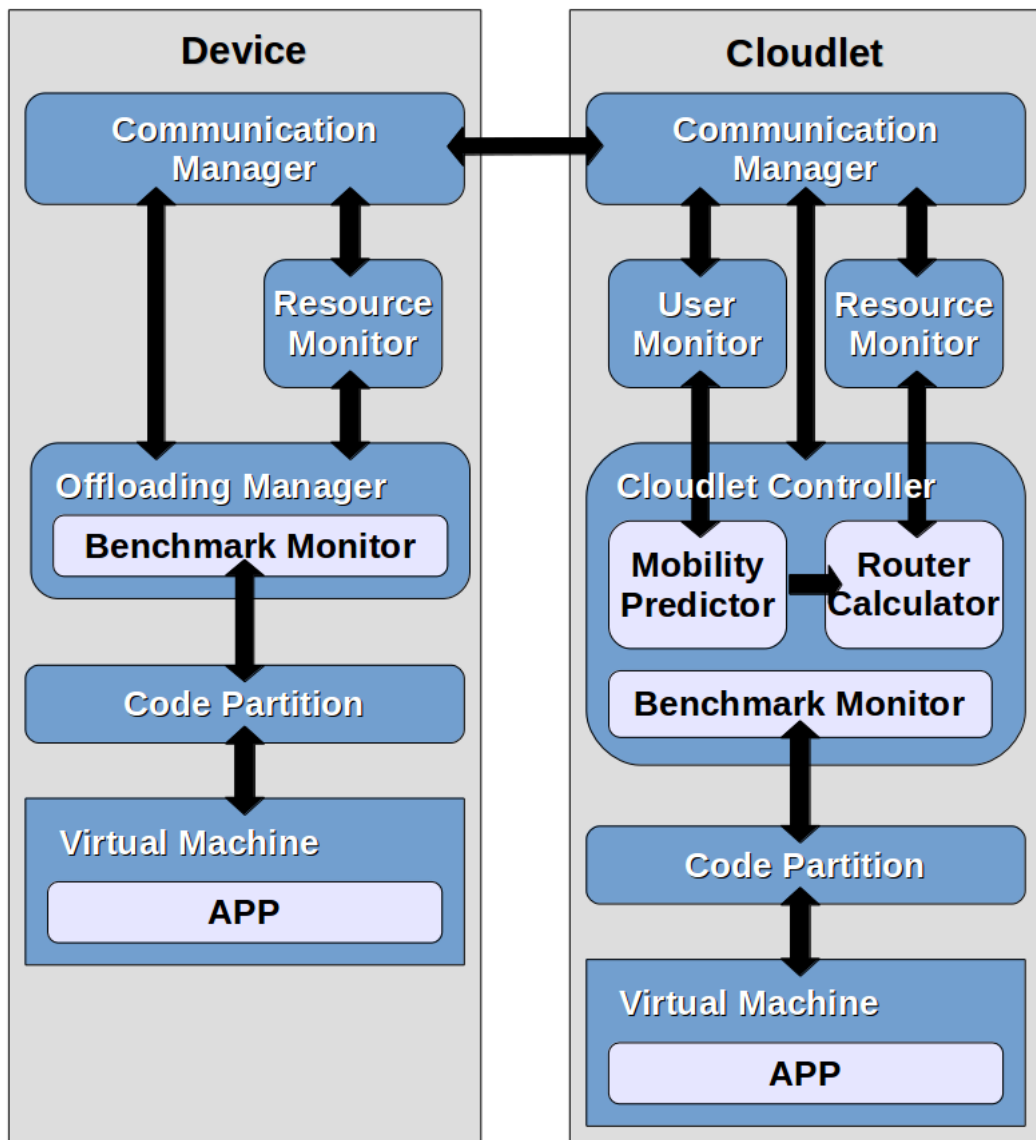


Figura 4.3: Arquitetura lógica de um ambiente de Computação em Névoa com suporte à mobilidade dos dispositivos IoT.

Arquitetura da Cloudlet: A arquitetura proposta para a *Cloudlet* pretende utilizar dados de deslocamento do usuário para prever seu trajeto. Baseado nesses dados, a *Cloudlet* prevê quais outras *Cloudlets* estarão no caminho do usuário e, se for pertinente, inicia o processo de migração do aplicativo, ou parte dele, para uma nova *Cloudlet*. Os módulos presentes nessa arquitetura estão listados abaixo.

- *Communication Manager*: Responsável por estabelecer a conexão com os dispositivos IoT e as conexões I2I com as demais *Cloudlets*;
- *Resource Monitor*: Coleta dados referentes à conexão com outras *Cloudlets* e os recursos disponíveis por esses servidores;
- *User Monitor*: Coleta dados referentes à conexão com os usuários e seus dados de localização ao longo do tempo;

- *Cloudlet Controller*: Baseado na qualidade de serviço entregue ao usuário ao executar a aplicação nesta *Cloudlet*, ao deixar de atender os requisitos definidos pela aplicação, o módulo decide realizar o pedido de migração da aplicação para outra *Cloudlet* melhor localizada.
 - *Mobility Predictor*: Com base nos dados coletados pelo *User Monitor* a respeito da localização do dispositivo IoT, o módulo apresenta um conjunto de prováveis trajetos a serem realizados pelo usuário;
 - *Route Calculator*: A partir dos dados da conexão com as demais *Cloudlets* coletados pelo *Resource Monitor* e nos trajetos apresentados pelo *Mobility Predictor*, o módulo identifica quais *Cloudlets* poderão ser usadas pelo usuário;
 - *Benchmark Monitor*: Monitora os módulos selecionados pelo *Code Partition*, coletando e mensurando o desempenho da aplicação que está sendo executada;
- *Code Partition*: Define por meio de regras estáticas ou dinâmicas os módulos do aplicativo que poderão ser executados em outras *Cloudlets*;
- *Virtual Machine*: Ambiente virtualizado para a execução do aplicativo. Gerencia os recursos físicos do servidor aplicando regras para a alocação de recursos da *Cloudlet* como núcleos de processamento, memória RAM e armazenamento.

Em um fluxo normal de execução, um aplicativo executado na máquina virtual, como por exemplo um jogo, terá seus métodos monitorados pelo *Code Partition* e pelo *Benchmark Monitor*. O *Code Partition* se comunicará com o aplicativo com o objetivo de identificar os métodos que poderão ser migrados para a Névoa sem comprometer a execução do aplicativo. Em um jogo, por exemplo, a renderização de objetos 2D e 3D do cenário da fase atual e das próximas fases referentes a posição atual do jogador poderá ser processada na Névoa. O *Benchmark Monitor* monitora em tempo real a demanda de recursos de cada método selecionado pelo *Code Partition*. Na implementação desta arquitetura no ambiente de simulação no MyIFogSim, os módulos do aplicativo a serem executados remotamente foram definidos de forma estática no módulo *Code Partition*.

Caso o dispositivo IoT esteja em um local coberto por uma *Cloudlet*, o módulo *Communication Manager* irá estabelecer a conexão com a *Cloudlet* e o módulo *Cloudlet Monitor* irá coletar em tempo real dados relacionados a essa conexão como latência e taxa de transferência, além de, principalmente, os dados referentes aos recursos que estão sendo consumidos na *Cloudlet*.

Os dados referentes à demanda de recursos do aplicativo e aos recursos disponíveis localmente e remotamente serão avaliados pelo módulo *Offloading Manager* no processo de decisão pelo *offloading* da aplicação. Se os recursos disponíveis no dispositivo IoT suprirem a demanda do aplicativo, esse será executado localmente. Caso o aplicativo demande de uma grande fonte de recursos e a *Cloudlet* a qual o dispositivo IoT estabeleceu uma comunicação oferece recursos suficientes, o módulo *Offloading Manager* poderá concluir que o *offloading* da aplicação será o processo mais benéfico ao usuário e ao sistema.

Para a avaliação inicial desta arquitetura no simulador MyIFogSim, foi definida uma demanda para o aplicativo de forma que o processo de *offloading* seja sempre necessário.

Desta forma os módulos *Resource Monitor*, *Benchmark Monitor* e *Offloading Manager* foram implementados de forma a realizar uma avaliação pré-estabelecida sobre o comportamento da aplicação quanto à necessidade da realização do processo de *Offloading*. Posteriormente a implementação deste cenário dinâmico poderá ser realizada de forma a complementar a avaliação dessa arquitetura.

Após o processo de *offloading* ser realizado, o aplicativo é executado na *Cloudlet* até que o deslocamento do usuário force a migração desse aplicativo para uma *Cloudlet* mais próxima. A decisão de migrar um aplicativo para outra *Cloudlet* fica sob responsabilidade da *Cloudlet* que está executando o aplicativo e está conectada ao dispositivo IoT.

Ao estabelecer a conexão com um dispositivo IoT, a *Cloudlet* passa a coletar em tempo real, com a ajuda do módulo *User Monitor*, dados relacionados ao dispositivo IoT, como localização, velocidade, direção, qualidade da conexão, entre outros. Cada *Cloudlet* também mantém dados de outras *Cloudlets* próximas ao local em que ela foi instalada. O módulo *Cloudlet Monitor* registra os dados referentes à qualidade da conexão e os recursos ociosos nessas *Cloudlets*.

Esses dados auxiliam o módulo *Cloudlet Controller* a definir o melhor local para a execução do aplicativo devido ao deslocamento do usuário. O submódulo *Mobility Predictor* presente no *Cloudlet Controller* é responsável por processar os dados coletados pelo *User Monitor*. Baseado nesses dados, o submódulo executa algoritmos de previsão de mobilidade para obter um conjunto de prováveis trajetos a serem realizados pelo usuário.

Outro submódulo presente no *Cloudlet Controller* é o *Route Calculator*. Esse submódulo recebe o provável trajeto do usuário, os recursos disponíveis em cada uma das *Cloudlets* vizinhas e a demanda do aplicativo. A partir desses dados o submódulo identifica quais *Cloudlets* poderão ser usadas pelo usuário durante seu trajeto.

A partir da lista de *Cloudlets* candidatas a serem utilizadas pelo usuário e dos dados relacionados à demanda do aplicativo coletados pelo *Benchmark Monitor*, o *Cloudlet Controller* decide se há a necessidade de iniciar um processo de migração para a próxima *Cloudlet* para executar parte, ou todos, os módulos do aplicativo, atendendo ao objetivo de apresentar uma melhor execução da aplicação ao usuário, ou seja, diminuindo o tempo de processamento, latência na comunicação, etc.

Assim como a implementação realizada para o módulo *Code Partition* do dispositivo IoT no simulador MyIFogSim, esse módulo presente na *Cloudlet* também foi definido de forma estática. Ao se realizar a migração entre duas *Cloudlets*, todos os módulos presentes no aplicativo são migrados em conjunto. Isso significa que a aplicação do usuário é executada em apenas uma *Cloudlet*, não sendo realizado nenhum processamento distribuído entre os nós da Névoa.

O módulo *Cloudlet Controller* é o principal módulo presente na arquitetura proposta de acordo com o objetivo deste trabalho. Conectado com as principais fontes de dados relacionadas à demanda do aplicativo, recursos oferecidos pelas demais *Cloudlets* e a localização atual e futura do usuário, este módulo será o responsável pela definição do local de execução dos aplicativos, concentrando o foco do desenvolvimento e avaliação dos mecanismos de migração propostos.

4.3 Cenário 1 - Avaliação em cenário otimista

A partir da identificação dos potenciais benefícios obtidos pela utilização de inferências sobre localizações futuras dos usuários no gerenciamento de recursos computacionais em trabalhos correlatos, como descritos no Capítulo 3, este trabalho pôde propor uma hipótese baseada nestes trabalhos para o contexto de Computação em Névoa.

Ao incorporar informações referentes à predição de mobilidade, este trabalho visa propor e avaliar uma hipótese acerca destas informações. Esta hipótese basea-se nos seguintes questionamentos: O uso de informações acerca da posição futura do usuário pode contribuir positivamente no processo de escolha do destino da máquina virtual do usuário durante seu processo de migração? Ou ainda, qual o impacto do desempenho do processo de migração no que se diz respeito ao gerenciamento dos recursos da Névoa e à qualidade de serviço oferecida ao usuário?

Neste trabalho, o intervalo de tempo futuro em que se tem conhecimento da localização do usuário é referenciado como *tempo de predição*. A partir da elaboração da hipótese proposta neste trabalho, pode-se elaborar cenários a fim de avaliar o desempenho do processo de migração com base no tempo de predição utilizado.

A fim de avaliar se a abordagem proposta poderia obter resultados promissores para o processo de migração, optou-se por uma avaliação mais otimista quanto à acurácia da previsão de localização futura dos usuários. Esta escolha evita que a abordagem seja comprometida pela acurácia dos algoritmos de predição encontrados na literatura. Resultados ruins nesta fase de avaliação poderiam indicar uma alta probabilidade de não se obter benefícios ao incluir esta métrica em outras propostas de migração, visto que a utilização de algoritmos reais de predição fatalmente apresentam uma menor acurácia com relação à localização prevista para o usuário. Tal taxa poderia mitigar os possíveis ganhos obtidos pela abordagem.

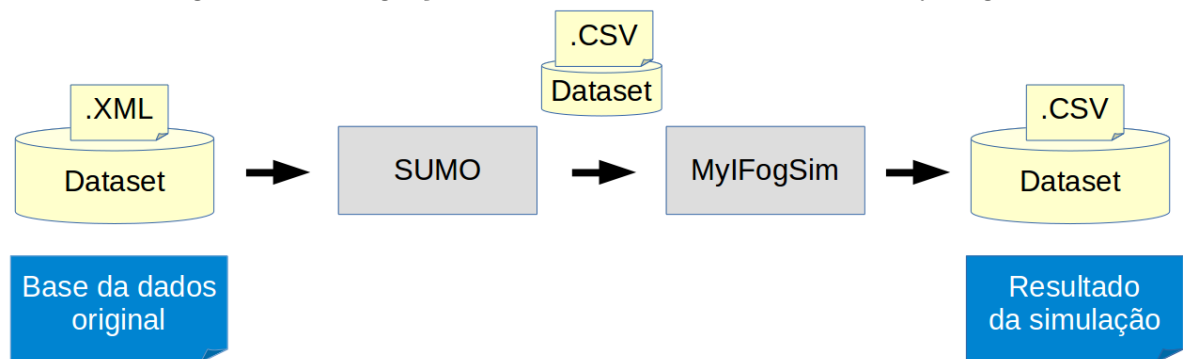
Com base nesse contexto, escolheu-se um cenário em que pode-se obter uma alta acurácia em relação à predição de localização dos usuários. Neste estudo foram consideradas rotas de ônibus urbanos como base para a mobilidade dos usuários. Devido ao seu trajeto fixo e um deslocamento relativamente previsível, as rotas dos ônibus apresentam um bom cenário para iniciar a avaliação desta abordagem.

4.3.1 Metodologia

Para a avaliação da abordagem proposta foi utilizado como base o simulador MyIFogSim [27], uma extensão do simulador IFogSim [20] que permite a inclusão de usuários móveis conectados à Névoa. A fim de avaliar o processo de migração a partir de um veículo, o modelo de mobilidade utilizado para esses usuários foi baseado no percurso de 500 ônibus da cidade de Luxemburgo [11]. Para a utilização dos percursos dos ônibus foi necessário realizar uma integração entre o simulador MyIFogSim e o simulador SUMO (Simulator for UrbanMObility)¹, responsável pela leitura e interpretação da base de dados. A Figura 4.4 ilustrada o fluxo dos dados para a simulação do cenário.

¹<http://sumo.dlr.de>

Figura 4.4: Integração entre os simuladores SUMO e MyIFogSim



A base de dados de localização de veículos fornecida pelos autores²[11] é utilizada como entrada pelo simulador SUMO. O simulador SUMO interpreta esta base de dados salva em formato .XML e filtra os veículos de interesse para este trabalho, mais especificamente, o simulador informa apenas o trajeto dos ônibus urbanos presentes na base de dados. O resultado da simulação é salvo em formato .csv.

Esta nova base de dados, que reflete o comportamento original dos veículos, foi gerada de forma otimizada para o contexto deste trabalho a fim de apresentar apenas os dados relevantes para o simulador MyIFogSim. Cada arquivo .csv representa a mobilidade de um veículo na simulação. Cada linha desse arquivo possui dados do veículo em um momento da simulação. Os dados utilizados foram posição vertical, posição horizontal, velocidade em metros por segundo, direção em radianos, além do tempo de simulação em que esses dados foram coletados.

Esta nova base de dados foi utilizada posteriormente como base para definir a mobilidade dos usuários no simulador MyIFogSim. O simulador faz a interpretação dessa base de dados e realiza algumas modificações para adaptar esses dados ao modelo de mobilidade utilizada por ele. Dentre essas alterações, há por exemplo, a conversão da velocidade do veículo de metros por segundo para quilometro por hora, e a conversão da direção, de radianos para os 8 principais pontos cardeais.

Após a simulação no MyIFogSim, uma nova base de dados é construída, esta apenas apresenta os resultados do comportamento do usuário para o gerenciamento dos recursos locais e na Névoa. Dentre esses resultados, há a latência apresentada pela aplicação ao longo do trajeto do usuário, migrações realizadas, pacotes requisitados e atendidos, dentre outros.

O simulador MyIFogSim permite uma avaliação do comportamento de aplicações relacionadas a dispositivos característicos da Internet das Coisas e a Computação na Névoa. Dentre os recursos oferecidos pelo simulador estão o gerenciamento do processo de transição (conhecido como *handoff* ou *handover*) entre as estações base, além de ferramentas para a realização e avaliação dos processos de *offloading* e migração de máquinas virtuais.

O simulador oferece alguns modelos para a realização do processo de migração das máquinas virtuais. O modelo utilizado neste trabalho é o processo de migração ao vivo (*live migration*), que consiste na transferência de partes da máquina virtual da máquina

²<https://github.com/lcodeca/LuSTScenario>

física de origem para o local de destino enquanto a aplicação ainda está em execução. Essa técnica diminui o tempo em que usuário fica impossibilitado de acessar sua aplicação durante o processo de migração.

Neste trabalho foi adotado o uso de máquinas virtuais tradicionais ao uso de *containers*. Esta escolha deu-se devido ao suporte a migração ao vivo, por parte do simulador, apenas para o uso em máquinas virtuais tradicionais. Apesar de haver na literatura algumas propostas para realizar a migração ao vivo de *containers* [19], por se tratar de uma proposta recente, esta técnica de migração ainda se encontra em fase de desenvolvimento. Se adotada, essa tecnologia promete um tempo de indisponibilidade menor ao usuário. Posteriormente a implementação desse cenário poderá ser realizada a fim de complementar a avaliação da abordagem de migração proposta.

Para o desenvolvimento deste trabalho, algumas modificações foram implementadas no processo de migração de máquinas virtuais realizado no simulador. A migração, no simulador, é realizada a partir de um conjunto de eventos relacionados ao processo de *handoff* (troca de antena na qual o usuário está conectado) e a escolha do conjunto de *Cloudlets* que pode oferecer uma melhor latência ao usuário. A sequência esperada para a realização dos eventos durante o processo de migração é ilustrada na Figura 4.5 e descrita com mais detalhes a seguir, baseada em Lopes *et al.* (2017) [27]:

- E1: Identificada a necessidade de se realizar o processo de migração;
- E2: Início do processo de migração;
- E3: Início do processo de *handoff*;
- E4: Fim do processo de *handoff*;
- E5: Fim do processo de migração;
- E6: Acesso à máquina virtual a partir da *Cloudlet* de destino.

As alterações executadas sobre o processo de migração se concentram exclusivamente no evento E1, responsável por determinar a realização do processo de migração e a escolha do destino da aplicação. Os demais eventos ocorrem como programado previamente no simulador.

O evento E1 define a realização do processo de migração baseado na disponibilidade de uma menor latência apresentada pelo conjunto de *Cloudlets* presentes na área de localização futura do usuário, comparada à atual *Cloudlet* em que usuário está conectado. A escolha deste conjunto de *Cloudlets* concentra a principal contribuição deste trabalho. Inicialmente, este conjunto era definido a partir da localização atual do usuário. A partir da hipótese deste trabalho, um ponto no trajeto futuro do usuário é utilizado como base para a escolha deste conjunto.

À medida em que o usuário se movimenta pelo mapa, ele passa a estar sob a cobertura de diferentes pontos de acesso, responsáveis por conectar o usuário à *Cloudlet* que possui sua máquina virtual. Idealmente, essa *Cloudlet* deve estar tão próxima quanto possível do ponto de acesso na qual o usuário está conectado a fim de obter uma menor latência.

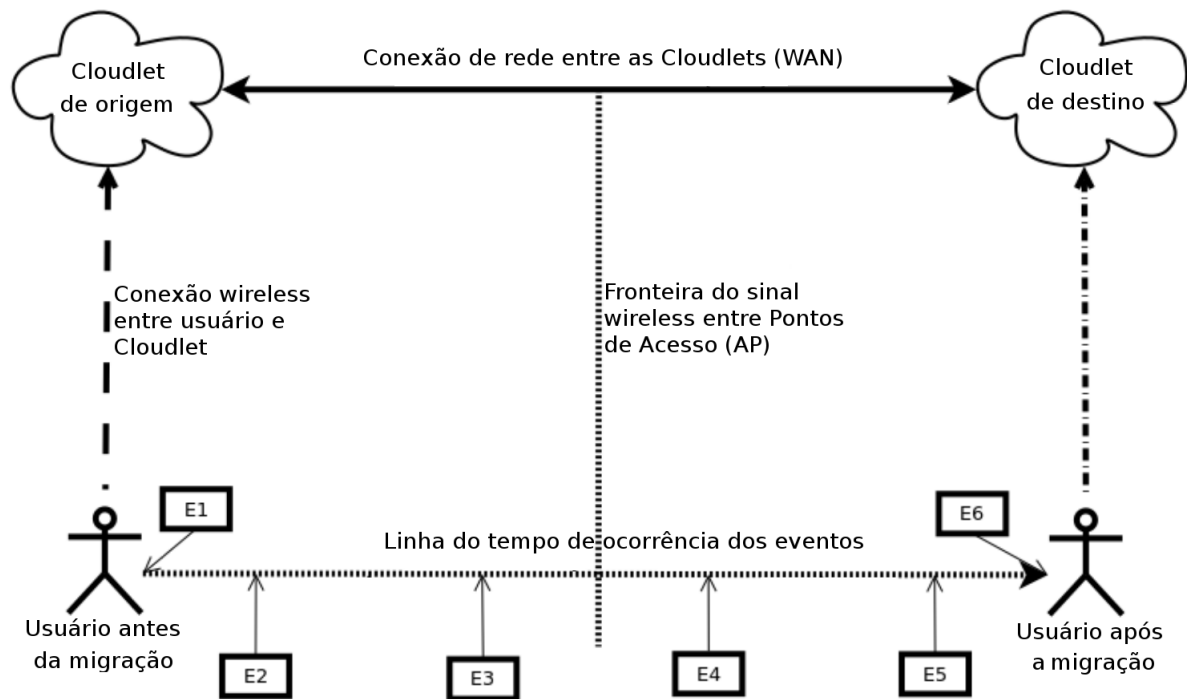


Figura 4.5: Sequência de passos do processo de migração. Adaptado de Lopes *et al.* (2017) [27]

A partir da posição, direção e velocidade do usuário com relação à área de cobertura do ponto de acesso em que ele está conectado, o processo de migração será iniciado caso haja alguma *Cloudlet* que ofereça uma melhor latência ao usuário. Como ilustrado na Figura 4.6, enquanto o usuário estiver presente na *zona de migração*, a viabilidade de realizar o processo de migração será avaliada e, em caso afirmativo, o processo de migração se iniciará no *ponto de migração*. O conjunto de até 10 *Cloudlets*, avaliadas nesse processo, se encontram em um raio de 45 graus a partir da direção em que o usuário está deslocando.

A cada troca de antena que o usuário realiza durante seu trajeto, esse processo também acaba apresentando um período de indisponibilidade de acesso à máquina virtual. A partir deste problema, idealmente, o processo de migração poderia sincronizar-se com o processo de *handoff* a fim de minimizar o tempo total de indisponibilidade ao usuário ao executar ambos os processos em paralelo. Na prática porém, essa sincronização nem sempre é possível.

Após a execução do processo E1, a ordem de execução dos demais eventos pode variar de acordo com a topologia da rede, com o deslocamento do usuário e com a disponibilidade de recursos oferecidas pela infraestrutura. Os eventos E3 e E4 (relacionados ao *handoff*), por exemplo, podem ocorrer antes do processo de migração (evento E2), ou ainda, após a migração ser finalizada (evento E5).

Para a avaliação do cenário proposto, as configurações da simulação referentes aos recursos disponíveis nas *Cloudlets* e a demanda requisitada pelos usuários foram definidos de forma coerente com trabalhos relacionados presentes na literatura [44, 27]. Cada *Cloudlet* responsável pelo processamento das aplicações foi configurada com um poder de processamento de 2,8 MIPS (Milhões de instruções por segundo), 25 Gigabytes de

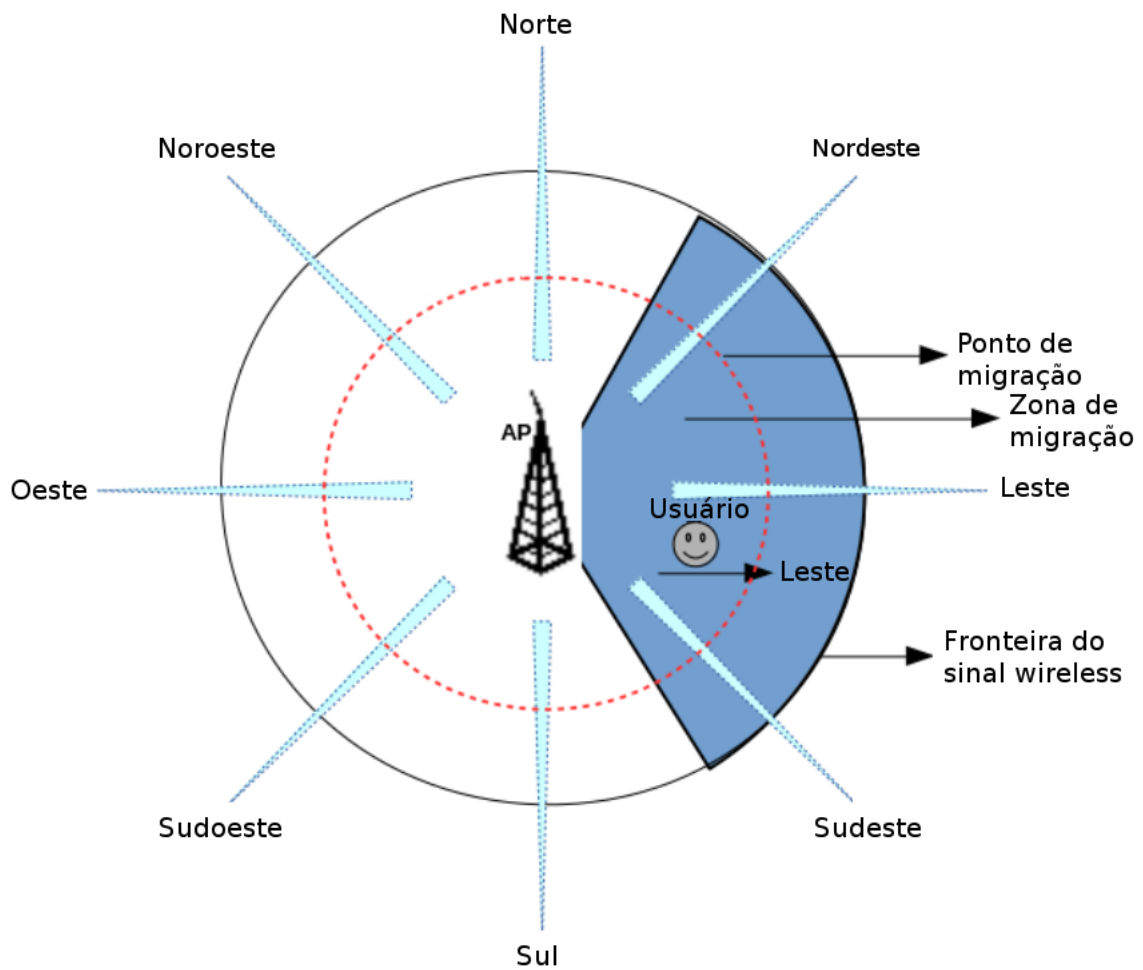


Figura 4.6: Processo de migração a partir da área de cobertura do ponto de acesso. Adaptado de Lopes *et al.* (2017) [27]

memória RAM e largura de banda de 1 Gigabit por segundo; cada *Cloudlet* foi conectada a um ponto de acesso através de um link de 100 Megabits por segundo com uma latência de 4 milissegundos. O raio de cobertura de cada ponto de acesso é de 500 metros.

A proposta foi avaliada analisando o comportamento das migrações a partir do conhecimento do trajeto futuro do veículo. Tais dados foram obtidos diretamente do trajeto recebido como entrada na simulação. Para cada um dos 500 trajetos de ônibus avaliados foram realizadas 6 simulações, cada uma tendo conhecimento do trajeto futuro do veículo dentro de um intervalo de tempo pré-definido, onde tais valores variaram de 0 a 300 segundos. O tempo de predição 0 não considera quaisquer informações sobre as futuras posições do veículo, sendo utilizada como base para comparação com a abordagem proposta. As métricas adotadas para a avaliação foram latência, número de migrações, tempo de utilização da máquina virtual sem que se necessite realizar uma migração, tempo de indisponibilidade de acesso à máquina durante a migração, volume de tráfego de dados gerado e tempo de migração.

Para avaliar se as características intrínsecas ao domínio dos usuários móveis avaliados neste trabalho, como velocidade e duração do trajeto, apresentavam algum impacto de desempenho no processo de migração, os veículos foram agrupados em 3 conjuntos de acordo

com sua velocidade média durante o trajeto. A Tabela 4.1 apresenta as características do conjunto de entrada utilizado no cenário avaliado.

Tabela 4.1: Características do conjunto de dados utilizado

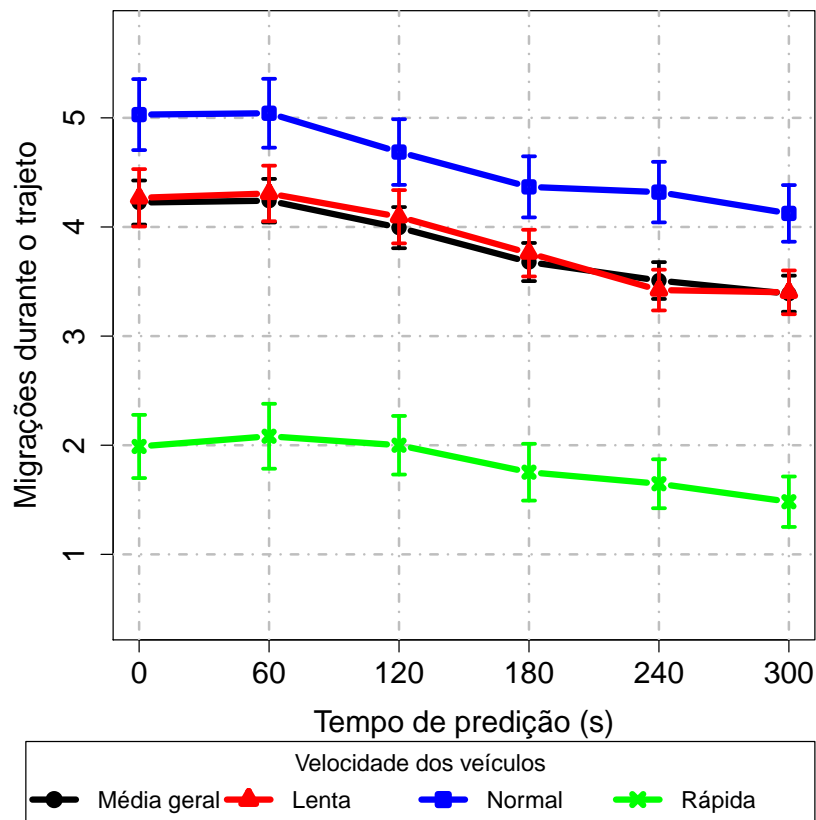
Conjunto	Duração Média do Percurso (min)	Quantidade
(0, 20) km/h	25,54	192
[20, 25) km/h	26,44	223
[25,) km/h	9,45	85
Média geral	23,31	500

Para cada combinação de métrica e conjunto avaliado, os resultados são apresentados na forma de média entre todos os indivíduos do conjunto e ilustrados dentro de um intervalo de confiança de 95%.

4.3.2 Resultados

A Figura 4.7 apresenta o número total de migrações realizadas durante todo o trajeto do veículo. Nota-se uma tendência na redução do número de migrações realizadas com o aumento no tempo de predição. Em média, os grupos chegaram a apresentar uma redução de até 20% no número total de migrações ao utilizar um conhecimento sobre o trajeto dos próximos 5 minutos do veículo.

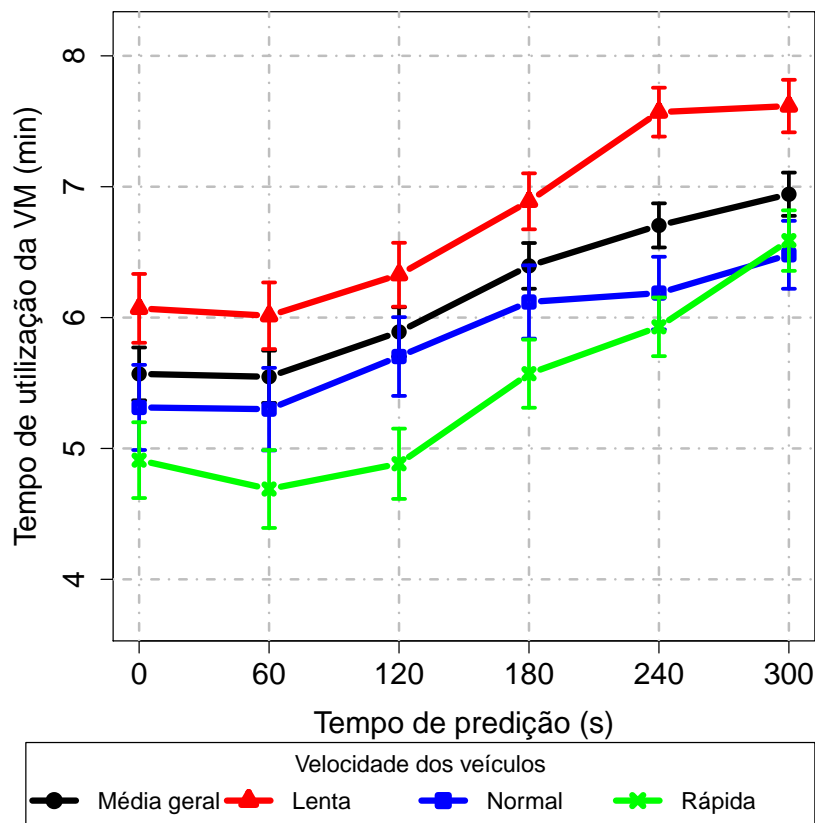
Figura 4.7: Número de migrações durante o trajeto



A disparidade no número de migrações relativa ao grupo de veículos rápidos em comparação aos demais pode ser explicada a partir das características do seu trajeto. O número absoluto de migrações está relacionado à distância que cada veículo percorreu. Assumindo intuitivamente a correlação entre a distância percorrida e o tempo despendido realizando tal trajeto, pode-se considerar que o gráfico apresenta uma escala coerente entre os grupos avaliados, tendo em vista as características apresentadas na Tabela 4.1.

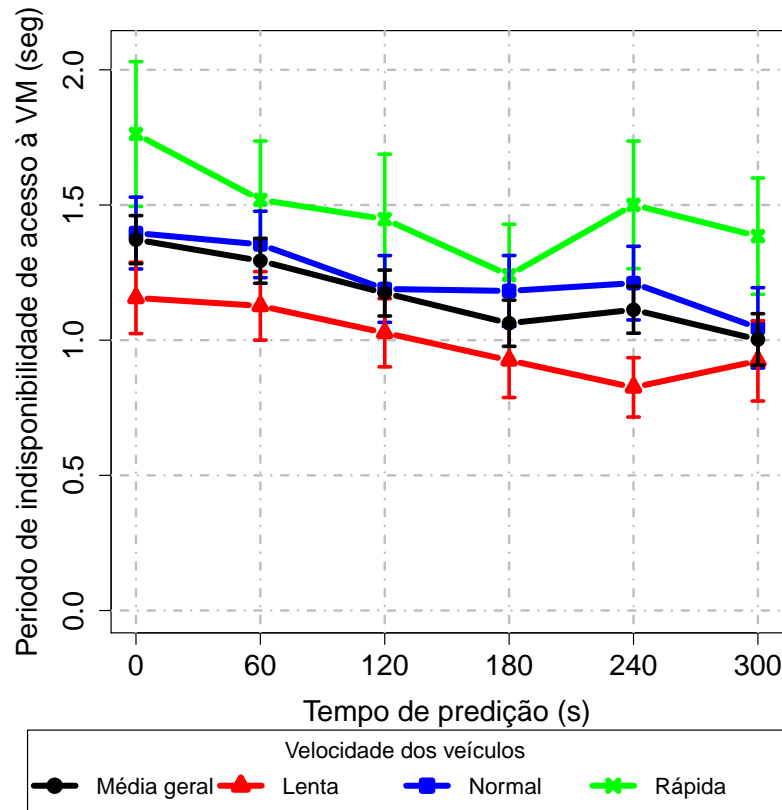
Apesar da redução no número absoluto de migrações ser um indicativo de que houve uma diminuição na necessidade de realizar uma migração, essas diferenças de trajeto apresentadas pelos grupos de veículos rápidos acabou diminuindo a percepção dos ganhos relativos obtidos em cada grupo. Para complementar a figura anterior, a Figura 4.8 apresenta o tempo médio em que as máquinas virtuais ficam hospedadas em cada *Cloudlet* antes que uma nova migração fosse necessária. O comportamento padrão apresentado pelos grupos foi de um crescimento no tempo de hospedagem da máquina à medida que se aumentava o conhecimento sobre o trajeto do veículo. Percebe-se que a velocidade dos veículos interfere no tempo de hospedagem da máquina virtual em cada máquina física. À medida em que a velocidade aumenta, menor o tempo necessário para que uma nova migração ocorra.

Figura 4.8: Intervalo de tempo entre migrações



Apesar da diminuição do número de migrações poder sugerir que os locais no qual as máquinas virtuais estão sendo posicionadas estão sendo mais adequadas ao usuário, uma métrica importante a se avaliar no contexto de aplicações relacionadas à Computação em Névoa é a latência. Um aumento na distância entre o usuário e sua aplicação pode

Figura 4.9: Tempo médio em que o veículo ficou sem acesso à máquina virtual a cada migração.

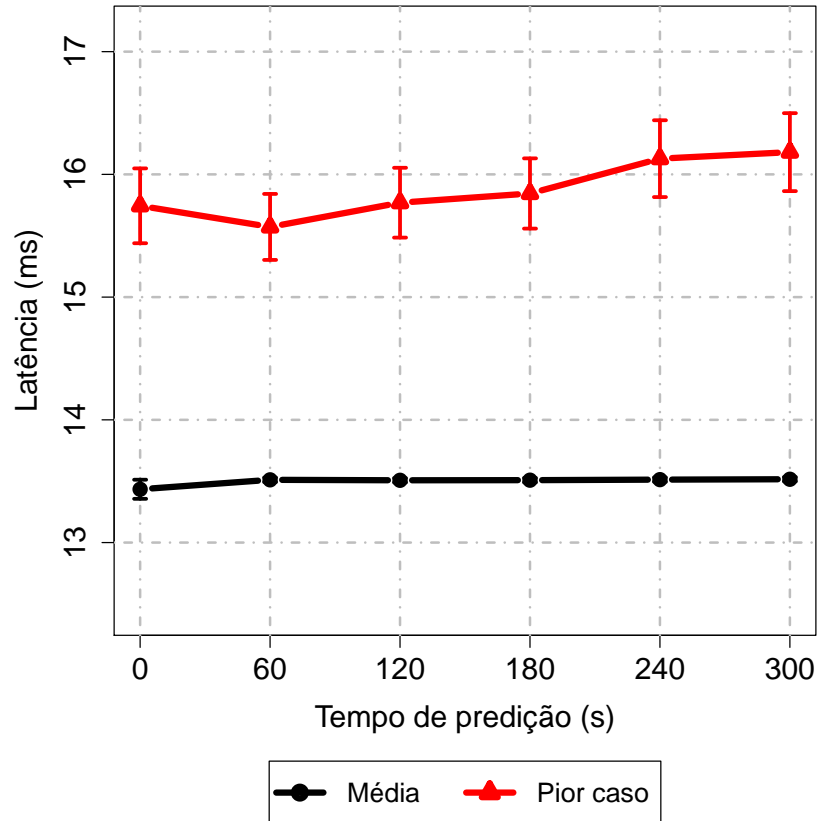


comprometer os níveis de latência exigidos em seu contexto. Considerando o intervalo de predição avaliado, as simulações realizadas não indicaram um aumento significativo da latência a partir da média geral e do pior caso encontrado. Mesmo com a redução no número de migrações, a latência não foi comprometida dentro do intervalo avaliado. Pode-se ver na Figura 4.10 que a latência média se mostrou estável em todos os casos e o pior caso não apresentou aumento estatisticamente significativo, com sobreposição dos intervalos de confiança dos pontos da curva.

As reduções no número de migrações e no tempo utilizado nesse processo contribuem para um aumento no tempo de disponibilidade da máquina virtual para o usuário, pois, durante parte do processo de migração, o usuário fica impossibilitado de acessar a VM. A Figura 4.11 apresenta o período de indisponibilidade de acesso à máquina virtual proporcional ao tempo total do trajeto do veículo. Percebe-se uma significativa melhora nos valores dessa métrica para todos os grupos de veículos avaliados com o aumento do tempo de predição. Para a média geral, percebe-se uma diminuição de até 40% no tempo de indisponibilidade total durante os trajetos dos veículos.

A partir da avaliação deste cenários pôde-se concluir que a utilização de informações acerca da posição futura do usuário permitiu uma melhor escolha para o destino da VM no processo de migração. A seleção de uma *Cloudlet* melhor localizada para o usuário, considerando sua posição em um futuro próximo, permitiu que esta *Cloudlet* oferecesse por mais tempo a qualidade de serviço requerida pelo usuário. Este aumento no tempo de uso da *Cloudlet* resultou em uma diminuição significativa no número de migrações necessárias

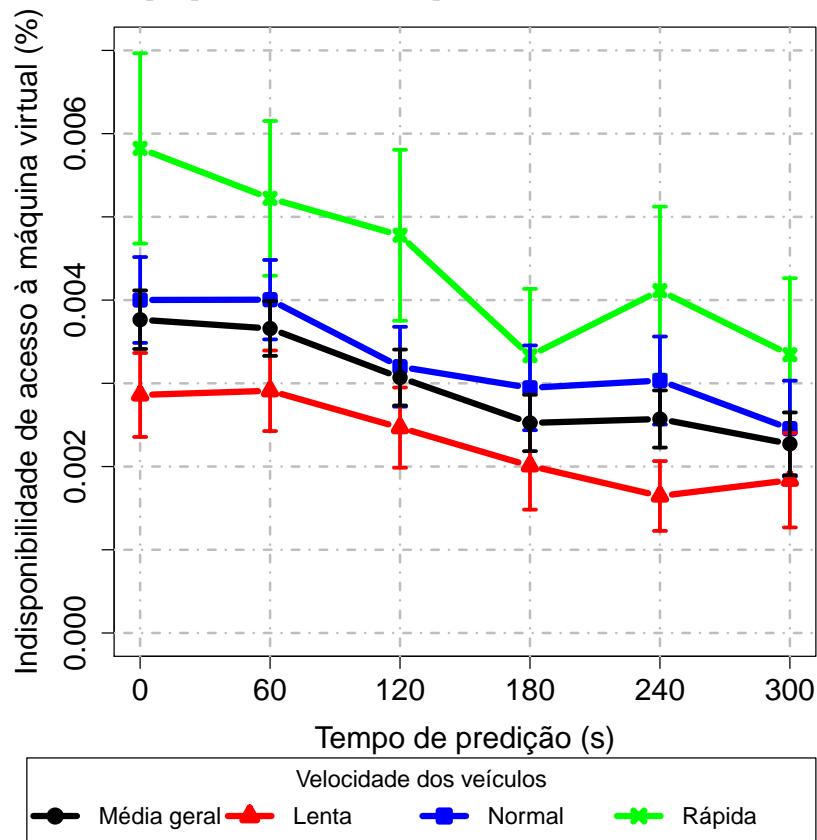
Figura 4.10: Latência



ao longo do trajeto do usuário. A diminuição no número de migrações também implicou em uma diminuição no período de indisponibilidade de acesso à máquina virtual por parte do usuários.

Os resultados deste experimento em um cenário propício para a migração permitiu avaliar a política proposta em um cenário de melhor caso para a qualidade da latência. Mesmo apresentando uma diminuição no número de migrações, a latência se mostrou estável para todos os cenários avaliados a partir da variação no tempo de predição utilizado. Os resultados deste trabalho foram apresentados no Simpósio Brasileiro de Redes de Computadores - SBRC 2018 [17]. Após a construção e avaliação deste cenário, um cenário com concorrência por recursos pôde ser avaliado.

Figura 4.11: Período proporcional de indisponibilidade de acesso à máquina virtual



4.4 Cenário 2 - Concorrência por recursos

A partir da avaliação inicial da política de migração proposta e dos resultados promissores apresentados por ela, outros cenários mais complexos puderam ser gerados a fim de complementar os testes realizados.

Considerando um ambiente altamente dinâmico como em uma cidade inteligente, a alta mobilidade dos usuários pode levar a uma grande variação na demanda dos usuários de acordo com a região e horário do dia. Considerando cenários onde, por exemplo, os usuários se deslocam das regiões mais periféricas da cidade em direção a zona industrial durante o início do horário comercial, e realizam este trajeto na direção contrária ao fim da tarde, uma crescente na demanda por recursos se fará presente em áreas específicas da cidade de acordo com o comportamento dos seus habitantes durante o dia.

Como ilustrado na Figura 4.12, uma crescente demanda pode ocorrer em uma mesma área de interesse para a migração das aplicações dos usuários. Neste contexto, uma concorrência por recursos poderá prejudicar o desempenho da aplicação. Nesse cenário, a melhor *Cloudlet* para um determinado usuário poderá não estar disponível, cabendo aos mecanismos de gerência de recursos da rede definir outra localização adequada para a aplicação.

A fim de avaliar a eficiência da proposta em cenários potencialmente mais comuns no contexto de uma cidade inteligente, o cenário construído para esta etapa se baseia na avaliação da migração proativa no contexto em que há concorrência pelos recursos dis-

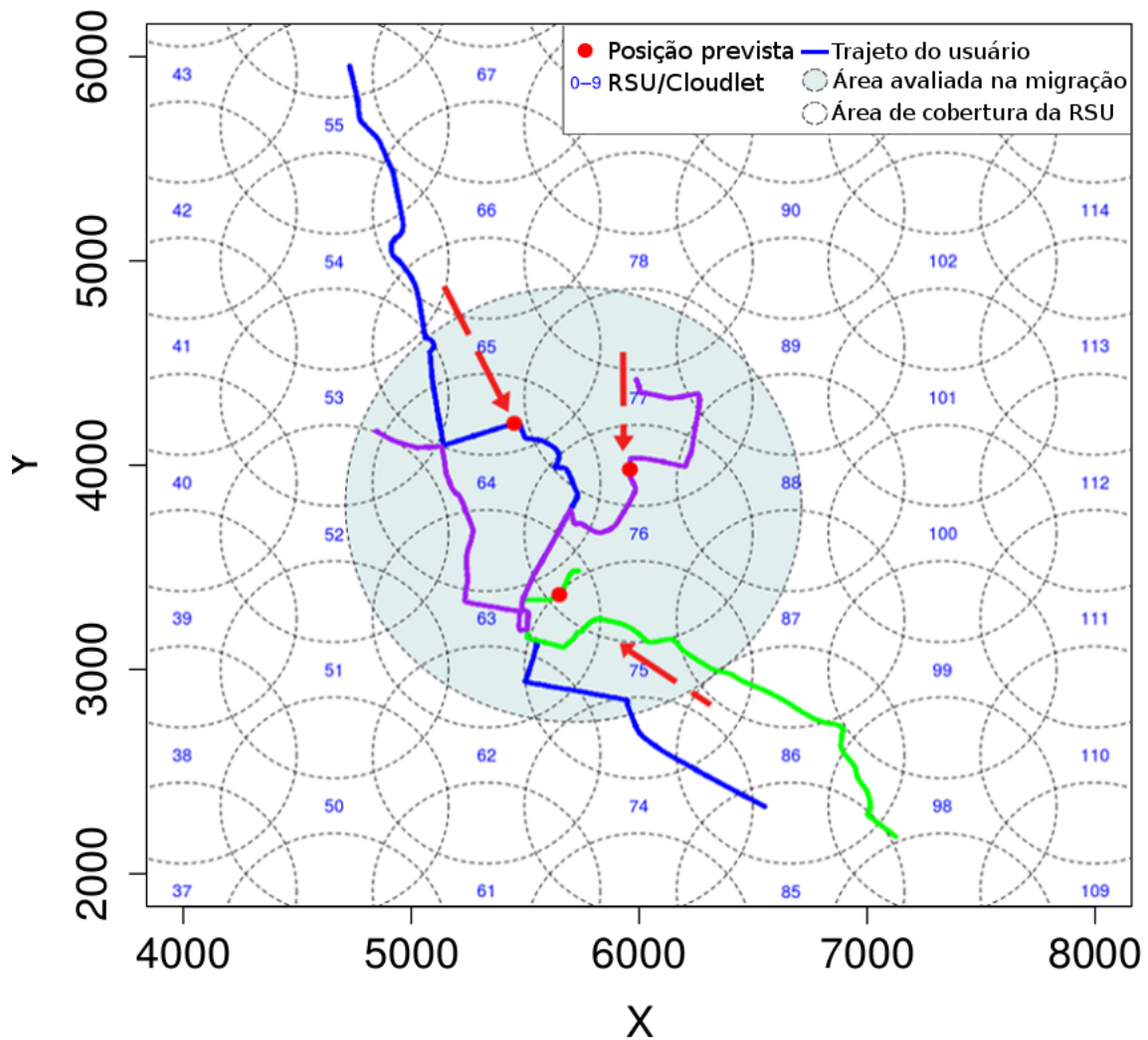


Figura 4.12: Cenário de migração de máquinas virtuais com concorrência por recursos.

poníveis na Névoa. Neste cenário, o destino ideal para máquina virtual de um usuário específico poderá não estar disponível devido à demanda de outros usuários. Neste contexto, deseja-se uma distribuição justa das máquinas virtuais entre as *Cloudlets* de forma a não comprometer o desempenho das aplicações dos usuários de maneira geral.

Os trabalhos encontrados na literatura focados na localização (*service placement*) ou migração de conteúdo de interesse de vários usuários simultaneamente, sendo dados para cache ou aplicações executadas em máquinas virtuais, no contexto de Computação na Névoa, geralmente são baseados em abordagens teóricas utilizando modelos matemáticos para solucionar o problema.

Assim como apresentado no cenário anterior, os trabalhos propostos para este problema não utilizam quaisquer informações sobre a localização futura dos usuários como parte dos critérios para escolha do destino do conteúdo do usuário. Com base neste contexto, este trabalho propõe um modelo baseado em Programação Linear Inteira para otimizar o processo de alocação das máquinas virtuais na Névoa. Diferentemente das demais soluções disponíveis na literatura, o modelo proposto considera dados de previsão de mobilidade dos usuários para selecionar o conjunto de *Cloudlets* candidatas a receberem

a máquina virtual do usuário. O modelo proposto visa maximizar o número de usuários atendidos pela infraestrutura e alocar as aplicações de tal forma a oferecer a menor latência possível a esses usuários.

Devido a restrições de desempenho, realizar a otimização da migração sobre uma visão global do mapa da cidade, ou seja, considerar todas as *Cloudlets* disponíveis como possíveis candidatas a receber a aplicação de um usuário, pode causar perdas relacionadas à escalabilidade e viabilidade da proposta. Com o aumento no número de usuários, e por consequência, um aumento na dinamicidade dos pedidos de migração, o número de possíveis combinações de soluções aumenta drasticamente. A partir desta característica do cenário, a resposta para o cálculo do melhor destino para a aplicação pode consumir um tempo indesejado, comprometendo a eficiência do processo.

Dado este cenário, este trabalho avaliou o comportamento do modelo linear aplicando-o sobre regiões, ou domínios pré-definidos. Todas as *Cloudlets* disponíveis no mapa foram agrupadas por regiões, e cada execução do algoritmo considera sob seu domínio apenas um desses grupos de até 10 *Cloudlets*, bem como todas as migrações que demonstram interesse como destino essa região. Neste trabalho, o grupo de *Cloudlets* é referenciado como *cluster* de *Cloudlets*. O modelo proposto é descrito com mais detalhes na seção a seguir.

4.4.1 Modelo de alocação proposto

O modelo de otimização baseado em Programação Linear Inteira proposto foi formulado a fim de otimizar a alocação das máquinas virtuais na Névoa de acordo com a mobilidade dos usuários. Considerando que cada usuário possui apenas uma máquina virtual, o modelo linear procura otimizar a distribuição das VMs através de suas duas funções objetivo: (1) maximizar o número de máquinas virtuais alocadas na Névoa e (2) minimizar a latência entre a máquina virtual e o usuário. No modelo proposto a primeira função objetivo possui prioridade sobre a segunda função.

Além das funções objetivo, o modelo linear proposto também é constituído por um conjunto de variáveis que visa caracterizar os recursos computacionais disponíveis na Névoa e os requisitados pelos usuários. A partir destas variáveis, um conjunto de restrições foram modeladas a fim de garantir uma solução factível com recursos disponíveis. A Tabela 4.2 apresenta as variáveis presentes no modelo.

O modelo linear proposto possui como entrada o conjunto N de *Cloudlets* presentes no *cluster* assim como o conjunto de aplicações A que demonstram interesse em realizar a migração para a região de domínio do *cluster*. Cada aplicação está associada a apenas um usuário e uma máquina virtual, assim como, no cenário avaliado, cada usuário possui apenas uma aplicação.

Além do conjunto de *Cloudlets*, o modelo possui como entrada um conjunto de variáveis que descrevem os recursos disponíveis em cada *Cloudlet* $n \in N$, sendo elas: processamento (Ω_n), memória RAM (ϕ_n), largura de banda (β_n) e armazenamento (γ_n). Analogamente, para cada aplicação $a \in A$, algumas variáveis de entrada representam os recursos computacionais demandados pela aplicação, como processamento (ω_a), memória RAM (φ_a), largura de banda (ρ_a) e armazenamento (α_a). O modelo também possui como entrada

Tabela 4.2: Variáveis do modelo linear proposto

Variáveis de entrada	
Variável	Descrição
N	Conjunto de Nós (<i>Cloudlets</i>) onde a aplicação pode ser executada
A	Conjunto de aplicações
Ω_n	CPU disponível no nó $n \in N$ (MHz)
ϕ_n	RAM disponível no nó $n \in N$ (MB)
β_n	Largura de banda disponível (Mbps) no nó $n \in N$
γ_n	Armazenamento disponível (MB) no nó $n \in N$
ω_a	CPU requisitada pela aplicação $a \in A$ (MHz)
φ_a	RAM requisitada pela aplicação $a \in A$ (MB)
ρ_a	Largura de banda (Mbps) requisitada pela aplicação $a \in A$
α_a	Armazenamento (MB) requisitado pela aplicação $a \in A$
$C_{a,n}$	Matriz de Custo. Matriz de Inteiros. Latência entre todos os pares a, n de nós $n \in N$ e o usuário relacionado a aplicação $a \in A$
Variáveis de decisão	
Variáveis	Descrição
$P_{a,n}$	Matriz de alocação. Matriz Binária. 1 quando a aplicação a está sendo executada no nó n

uma matriz de custo que indica a latência entre o usuário e cada uma das *Cloudlets* avaliadas.

A Equação 4.1 apresenta a primeira função objetivo do modelo, relativa à maximização do número de pedidos de migração atendidos. Ao maximizar a matriz de alocação $P_{a,n}$, o modelo visa alocar o número máximo de aplicações possíveis no *cluster* em questão. A segunda função objetivo, apresentada na Equação 4.2, minimiza a latência entre o usuário e a *Cloudlet* em que sua aplicação será migrada. Esta minimização é realizada a partir do produto entre a matriz de alocação e a matriz de custo.

$$\max \sum_{a \in A} \sum_{n \in N} P_{a,n} \quad (4.1)$$

$$\min \sum_{a \in A} \sum_{n \in N} P_{a,n} \times C_{a,n} \quad (4.2)$$

A segunda função objetivo é flexível o suficiente para adaptar o modelo a diferentes objetivos apenas alterando a métrica utilizada para preencher a tabela de custo. Além das funções objetivo, o conjunto de restrições também é descrito a fim caracterizar o modelo proposto de acordo com as especificações da infraestrutura.

$$\forall a \in A : \sum_{n \in N} P_{a,n} \leq 1 \quad (4.3)$$

$$\forall n \in N : \sum_{a \in A} P_{a,n} \times \omega_a \leq \Omega_n \quad (4.4)$$

A Equação 4.3 assegura que a solução gerada não aloca a aplicação em mais de uma *Cloudlet* simultaneamente. A Equação 4.4 garante a alocação de processamento de acordo com os recursos disponíveis e a demanda dos usuários. A soma da demanda por processamento por parte das aplicações alocadas no nó n não deve ultrapassar os recursos disponíveis nele.

$$\forall n \in N : \sum_{a \in A} P_{a,n} \times \varphi_a \leq \phi_n \quad (4.5)$$

$$\forall n \in N : \sum_{a \in A} P_{a,n} \times \rho_a \leq \beta_n \quad (4.6)$$

A Equação 4.5 impõe restrições quanto a alocação de memória RAM nas *Cloudlets*. Análoga ao uso de processamento na Equação 4.4, a soma da memória RAM alocada pelas aplicações não deve exceder os recursos disponíveis na *Cloudlet*. A Equação 4.6 limita a alocação de aplicações de acordo com as suas demandas por tráfego considerando a capacidade da conexão disponível no nó.

$$\forall n \in N : \sum_{a \in A} P_{a,n} \times \alpha_a \leq \gamma_n \quad (4.7)$$

A Equação 4.7 aplica restrições referentes ao armazenamento alocado pelas aplicações. A soma do armazenamento utilizado pelas aplicações alocadas não deve ultrapassar os recursos disponíveis na *Cloudlet*.

Este modelo foi implementado utilizando a ferramenta de modelagem IBM ILOG CPLEX 12.7.1 [12]. Simulações foram executadas para avaliar o seu desempenho e são apresentadas nas próximas seções.

4.4.2 Metodologia

A fim de avaliar o modelo linear proposto, um cenário de concorrência por recursos foi elaborado para avaliar o gerenciamento de recursos na Névoa. Assim como na primeira etapa de avaliação da política de migração proposta, a versão do simulador MyIFogSim integrada com o simulador de mobilidade SUMO foi utilizada para avaliar o cenário proposto.

O padrão de mobilidade dos usuários utilizado na simulação foi novamente extraído da base de dados Luxembourg SUMO Traffic (LuST). Para este cenário foram utilizados 2700 viagens de ônibus realizadas, em média, em 26 minutos a uma velocidade de 22.3 km por hora. Cada usuário instanciava apenas uma máquina virtual e se conectava à *Cloudlet* a partir de pontos de acesso uniformemente distribuídos no mapa utilizado. Cada ponto de acesso possui um raio de cobertura de aproximadamente 500 metros e é conectado a uma *Cloudlet*.

As configurações da simulação quanto aos recursos disponíveis nas *Cloudlets* e a demanda dos usuários foram definidas de forma consistente com os valores encontrados em trabalhos relacionados [43, 27]. Cada *Cloudlet* possui uma capacidade de processamento de 2800 milhões de instruções por segundo, 8 GB de memória RAM, 80 GB de armazenamento e uma conexão de 100 Mbps e latência de 4 milissegundos com seu ponto de

acesso. Uma conexão uniformemente distribuída entre 1 e 10 Gbps conecta cada par de *Cloudlets*. A máquina virtual utilizada nas simulações possui 200 MB.

A fim de comparar o modelo linear proposto, a estratégia utilizada como base de comparação se baseia em uma estratégia gulosa. Essas estratégias serão utilizadas durante o processo de migração para a seleção do destino das aplicações dentre as opções disponíveis no *cluster*. Dentre um conjunto de 10 *Cloudlets*, as abordagens irão selecionar os nós que irão servir aos usuários com as menores latências, considerando as restrições de recursos impostas anteriormente

As abordagens foram avaliadas em cenários com 30, 60 e 90 usuários simultâneos. Cada abordagem foi avaliada utilizando informações da localização futura do usuário variando de um intervalo de 0 a 5 minutos. A técnica de migração utilizada é novamente a migração ao vivo (*live migration*).

As métricas utilizadas para a avaliação do cenário foram a média e pior caso de latência apresentada pelos pacotes requisitados enquanto a VM não estava em processo de migração, número de migrações realizadas durante todo o trajeto do usuário, e o tempo de indisponibilidade de acesso à máquina virtual. Os resultados são apresentados a seguir considerando um intervalo de confiança de 95%. As abordagens utilizadas para a alocação das aplicações são apresentadas nos gráficos a seguir como PLI, para a abordagem proposta, e Padrão, para a abordagem gulosa. Os números 30, 60 e 90 após o nome da abordagem utilizada referem-se ao número de usuários utilizados durante a simulação. O eixo x representa o tempo de predição do trajeto do usuário.

4.4.3 Resultados

A partir das simulações dos cenários avaliados pode-se obter alguns resultados acerca do desempenho das abordagens avaliadas. O número de migrações, ilustrado na Figura 4.13, apresenta uma tendência em diminuir à medida em que se tem mais informações acerca do caminho futuro do usuário, ou seja, há um aumento no tempo de predição. Estas informações adicionais acerca do trajeto do veículo permitem que ambas as abordagens selecionem de uma maneira mais eficaz o conjunto de *Cloudlets* candidatas a receberem as aplicações dos usuários. A ciência acerca do trajeto do veículo pelos próximos 5 minutos permite uma diminuição de quase 31% no número de migrações efetuadas pela VM ao se utilizar a abordagem gulosa, e uma diminuição de quase 50% ao se utilizar o modelo linear proposto.

Mesmo utilizando otimizações no processo de migração, como a migração ao vivo, o usuário não terá acesso a sua aplicação na Névoa durante certo período de tempo durante parte do processo de migração. Reduzir o número de migrações durante o trajeto do usuário significa diminuir a soma desses períodos de indisponibilidade ao tempo total em que o usuário ficou impossibilitado de acessar sua aplicação. Os resultados obtidos pelos cenários simulados, considerando esta métrica, são apresentados na Figura 4.14.

A diminuição no número de migrações pode sugerir que a localização escolhida como destino da migração da máquina virtual está se apresentando como mais apropriada para o usuário. Apesar do número de migrações se mostrar relevante neste cenário, outra métrica também deve ser avaliada no contexto de aplicações relacionadas a Computação

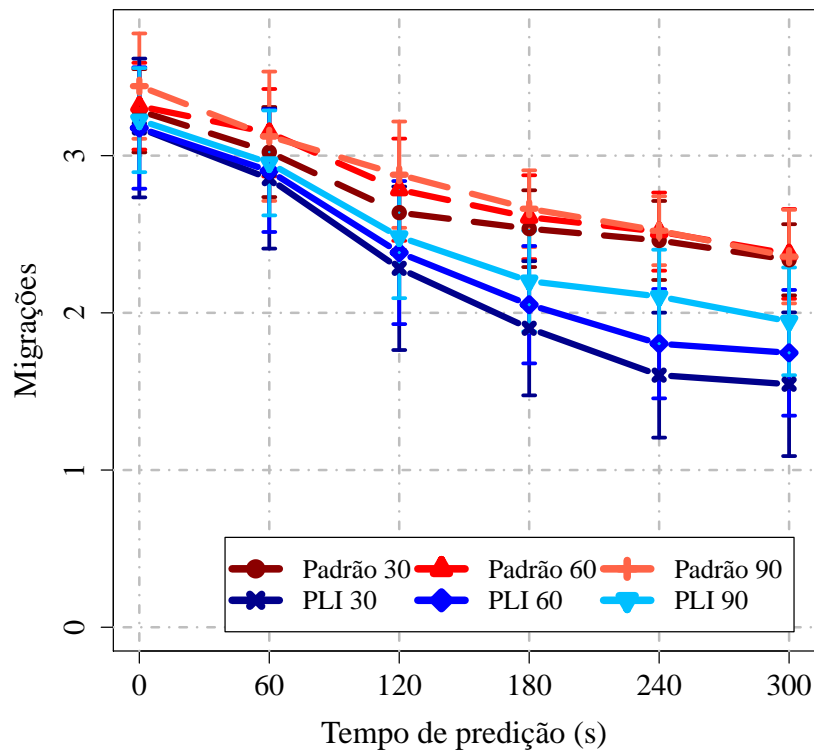


Figura 4.13: Número médio de migrações da VM durante o trajeto do usuário

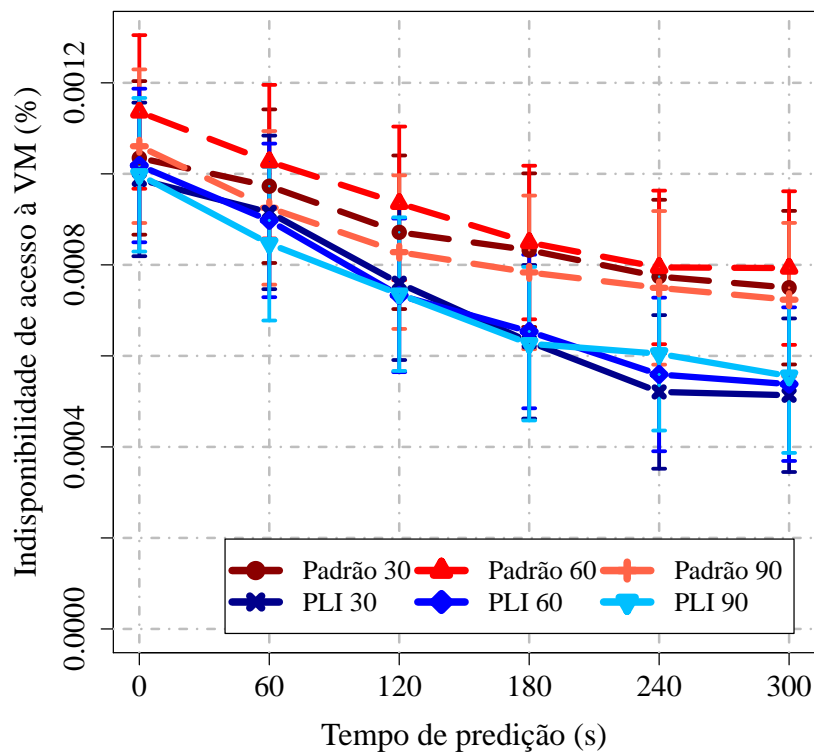


Figura 4.14: Período proporcional de indisponibilidade de acesso à VM

na Névoa, a latência. Um aumento na distância entre o usuário e sua máquina virtual

pode comprometer os níveis de latência necessários para uma boa execução da aplicação.

A Figura 4.15 apresenta a latência média observada nos cenários de simulação avaliados. A partir dos dados obtidos, pode-se observar que a arquitetura da Névoa providenciou uma latência de 15 ms para as aplicações dos usuários, o que pode garantir uma latência aceitável para muitas aplicações neste ambiente. Também pode-se notar que os resultados apontaram, para todo o intervalo de tempo de predição utilizado na avaliação dos cenários, níveis baixos e estáveis acerca da latência média apresentada pela aplicação.

A partir do desempenho apresentado pela abordagem gulosa utilizada como base para comparação com o algoritmo proposto, os resultados das simulações também indicam que o modelo linear proposto para otimizar a alocação das máquinas virtuais apresenta níveis de latência, em média, menores ou iguais aos obtidos pela abordagem gulosa utilizada. Este resultado é visível nos três níveis de densidade de usuários avaliados (30, 60 e 90 usuários).

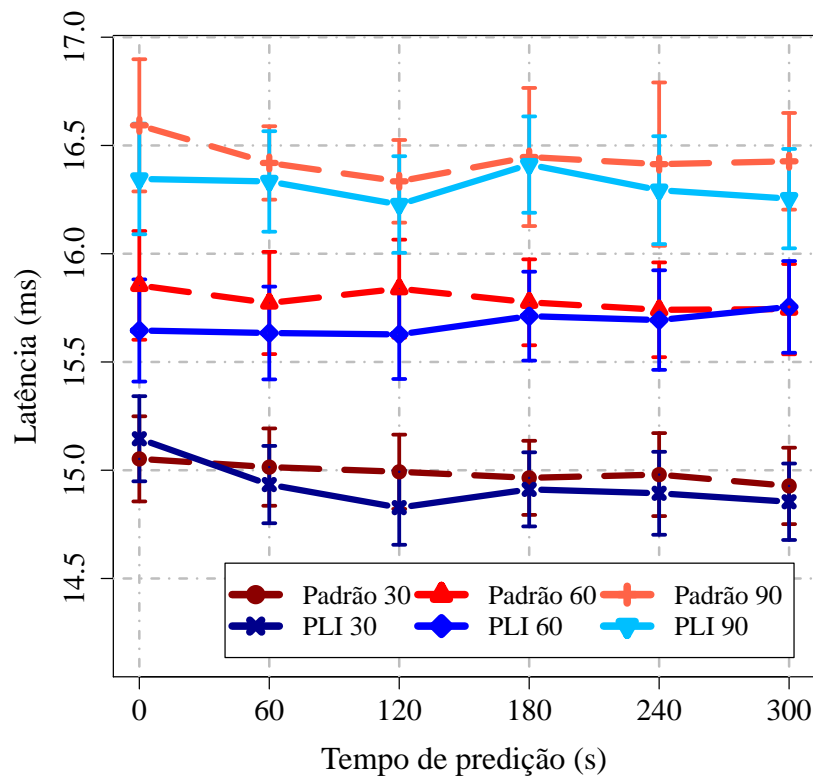


Figura 4.15: Latência média

A fim de complementar a avaliação acerca da latência média apresentada pelos algoritmos, a latência em pior caso também foi avaliada. Ao analisar o cenário de pior caso, a maior latência apresentada pela aplicação de um usuário ao longo do seu trajeto é coletada a fim de identificar se ele está obtendo uma latência insatisfatória, mesmo que por um curto período de tempo. Os resultados, apresentados na Figura 4.16, indicam que, em geral, não há um aumento significativo na latência obtida pela aplicação ao se utilizar a política de migração proativa. Esse comportamento pode ser observado nas duas abordagens avaliadas neste cenário, tanto no modelo linear proposto quanto na abordagem gulosa.

Assim como nos resultados relacionados à latência média, os resultados para a latência no pior caso também indicam que o modelo proposto baseado em Programação Linear Inteira apresenta os melhores resultados se comparado com a abordagem gulosa utilizada como base de comparação.

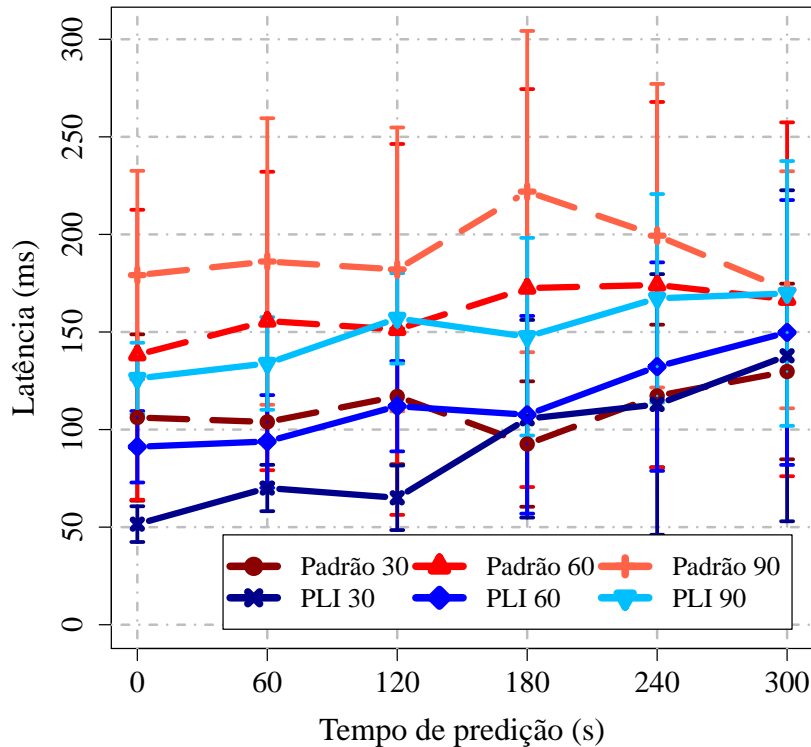


Figura 4.16: Latência em pior caso

Os valores para a latência em pior caso apresentados nos resultados das simulações indicam que a abordagem proativa, para o intervalo avaliado, não apresenta mudanças significativas na latência da aplicação se comparado com cenários em que nenhuma informação sobre o trajeto futuro do usuário é utilizada, mesmo a abordagem diminuindo em até 50% o número de migrações necessárias.

Como a prioridade modelada para o algoritmo proposto foi maximizar o número de requisições atendidas, o modelo proposto baseado em Programação Linear Inteira não necessariamente apresenta o resultado ótimo para a instância recebida no que se diz respeito a minimização da latência. Os resultados das simulações, no entanto, indicam que o desempenho se mostra satisfatório. Os resultados do cenário avaliado indicam que o modelo proposto apresenta melhores resultados tanto em termos de latência média e em pior caso, quanto em número de migrações realizadas.

O ambiente utilizado na Seção 4.3 avaliou o comportamento da política de migração proativa em um cenário otimista para o processo de migração. Nesse cenário, a *Cloudlet* que melhor atenderia o usuário como destino da VM possui garantia de disponibilidade. Em um cenário de escassez de recursos, essa disponibilidade da *Cloudlet* ótima não é garantida, o que poderia modificar a dinâmica de migrações na rede.

A fim de complementar os resultados presentes na Seção 4.3, um cenário com escassez

de recursos foi construído a fim de avaliar a política de migração proposta. Os resultados apresentados neste capítulo indicam que a escassez de recursos não alterou as tendências demonstradas na Seção 4.3. Assim como cenário anterior, o número de migrações diminuiu à medida em que o tempo de predição era incrementado. Os níveis de latência também não apresentaram um aumento significativo. Os resultados referentes a esse cenário foram apresentados no IEEE Symposium of Computers and Communication - ISCC 2018[18].

Capítulo 5

Conclusão

Gerenciamento e alocação de recursos da Névoa tem se mostrado um dos grandes desafios a serem endereçados a fim de oferecer uma qualidade de serviço aos usuários móveis conectados a essa infraestrutura. A migração de máquinas virtuais encontra-se nesse contexto. As atuais propostas pouco utilizam alguma inferência sobre a localização futura dos usuários durante esse processo de migração. Tendo em vista este cenário, este trabalho contribui com o estado da arte ao apresentar uma proposta de otimização do processo de migração de máquinas virtuais em um ambiente de Computação em Névoa baseada na predição de mobilidade de usuários móveis. Este trabalho denomina a política de migração proposta como migração proativa de máquinas virtuais.

Esta dissertação apresentou uma proposta de arquitetura hierárquica para Computação em Névoa com suporte para o descobrimento e uso de informações relativas ao trajeto futuro dos usuários da infraestrutura. A partir de tais informações, pode-se otimizar a escolha da *Cloudlet* a ser utilizada como destino da máquina virtual do usuário durante o processo de migração.

Tendo como base a arquitetura proposta, este trabalho avaliou a hipótese proposta em cenários favoráveis à migração encontrados no contexto de cidades inteligentes. Usuários de transporte público como ônibus, metro e trem possuem um trajeto bem definido, contribuindo para uma alta acurácia na predição de sua mobilidade.

Com base nas simulações desenvolvidas durante este trabalho, pode-se concluir que incorporar informações sobre o trajeto futuro do usuário pode reduzir o número de migrações necessárias para atender os clientes ao longo de seu trajeto sem comprometer a latência oferecida. Em um cenário propício para o processo de migração (alta acurácia na predição de mobilidade do usuário e garantia de disponibilidade de recursos computacionais por parte da infraestrutura de Névoa), o conhecimento sobre a localização aproximada do usuário nos próximos 5 minutos é suficiente para ajustar o destino da migração a ponto de diminuir em 20% o número total de migrações durante todo o percurso do usuário. Com a redução no número de migrações, o tempo de indisponibilidade de acesso à máquina virtual causada durante cada migração, também pôde ser reduzido.

Após a avaliação da hipótese proposta e a constatação dos benefícios e dos resultados promissores apresentados pela migração proativa em um cenário propício, outros cenários mais complexos foram analisados. Esta dissertação também avaliou o impacto da migração proativa em um cenário de escassez de recursos na Névoa, e propôs um algoritmo baseado

em Programação Linear Inteira para otimizar a escolha dos destinos das máquinas virtuais. O algoritmo proposto não necessariamente apresenta o resultado ótimo dadas todas as combinações de respostas possíveis, mas apresenta melhores resultados se comparada com uma abordagem gulosa.

O algoritmo proposto apresentou escolhas que resultaram, em média, em um menor número de migrações necessárias durante o trajeto do usuário se comparado com os resultados apresentados por uma abordagem gulosa para a seleção do destino das máquinas virtuais. As simulações também demonstraram que os níveis de latência se mostram estáveis comparadas com o abordagem gulosa utilizada como padrão.

Em todos os cenários avaliados, tanto para o algoritmo proposto quanto para a abordagem gulosa, a incorporação de informações sobre o trajeto futuro do usuário melhoram a escolha dos destinos das máquinas virtuais, aumentando o período de tempo em que uma *Cloudlet* consegue atender a demanda do usuário antes que seja necessário iniciar um novo processo de migração.

Pontos em aberto: Considerando os diversos cenários e demandas relacionados a IoT no contexto de Cidades Inteligentes, alguns grupos de usuários, como usuários de ônibus urbanos, metrô e trem, apresentam um padrão de mobilidade altamente predizível. No entanto, nem todos os cenários apresentam esta característica. Como apresentado neste trabalho, incorporar informações relacionadas à predição de mobilidade desses usuários ao processo de migração de máquinas virtuais em um ambiente de Névoa apresenta vantagens quanto ao aumento da qualidade de serviço oferecida aos usuários.

A partir do atual cenário apresentado pelo estado da arte, alguns tópicos de pesquisa se mostram em aberto nesta nova abordagem para migração de dados e processamento em ambientes de Névoa. Tendo em vista os resultados promissores do processo proativo de migração de máquinas virtuais em cenários em que a mobilidade dos usuários possui uma rota pré-definida, outros cenários mais genéricos poderiam ser avaliados. Cenários em que é conhecido apenas o destino final dos usuários e não necessariamente sua rota já podem apresentar um nível de acurácia inferior aos avaliados pelo estado da arte. A viabilidade do modelo proativo neste novo cenário pode ser avaliada.

A continuidade da pesquisa no tema abordado durante esse trabalho de mestrado pode incorporar otimizações tanto do ponto de vista do usuário quanto do provedor dos recursos. A partir do ponto de vista do usuário dos recursos, há tópicos a serem abordados a fim de propor abordagens que otimizem a qualidade de serviço entregue aos usuários, diminuindo, por exemplo, os atuais níveis de latência e indisponibilidade de acesso à aplicação por parte das atuais abordagens encontradas na literatura. Baseado no ponto de vista do provedor dos recursos, pode-se apresentar modelos para a otimização do processo de migração das aplicações dos usuários móveis de tal forma que os recursos de armazenamento, processamento e rede sejam minimizados, beneficiando coletivamente o maior número de usuários.

Estes tópicos em aberto poderão ser alcançados através da investigação das seguintes questões de pesquisa:

1. *A migração proativa se mostra eficiente em cenários de menor acurácia na predição de mobilidade dos usuários?* Como apresentado durante este trabalho, a migração

proativa apresenta ganhos ao adicionar predição de mobilidade no auxílio da escolha do destino da aplicação do usuário no processo de migração. Os cenários avaliados no escopo deste trabalho, porém, consideram uma alta acurácia nessa predição. Pode-se realizar uma avaliação sobre a viabilidade dessa abordagem em cenários mais complexos, tanto relacionados à acurácia da predição quanto às demandas específicas por recursos por parte dos usuários. Baseando-se na abordagem proativa e na avaliação de sua viabilidade neste cenário, pode-se propor mecanismos para otimizar o gerenciamento de recursos da rede;

2. *A utilização de outras ferramentais, como Teoria de Jogos e Aprendizagem de Máquina, se mostra viável para otimizar o gerenciamento de recursos da Névoa?* Apesar dos autores apresentarem em [18] uma solução baseada em um modelo matemático de Programação Linear Inteira para o gerenciamento de recursos da Névoa em cenários onde há escassez de recursos para os usuários, outras abordagens ainda não foram exploradas pela literatura neste novo cenário. Devido ao grande volume de metadados gerados pelos diversos dispositivos, trabalhos recentes têm se baseado nessa coleta de dados acerca dos usuários e aplicações para otimizar os processos de decisão relacionados ao gerenciamento de recursos computacionais [3]. Nesse contexto, o uso de Aprendizagem de Máquina tem sido utilizado em variados cenários [8], tanto para otimizar o processo de migração [29] quanto para a definição do destino das aplicações [16]. Devido à necessidade de se distribuir de forma eficiente as aplicações pelas *Cloudlets* disponíveis, alguns trabalhos [24, 48] têm proposto o uso de Teoria de Jogos para essa função. Esta técnica possibilita estabelecer de forma mais justa a distribuição de recursos em disputa pelos usuários. Tanto os trabalhos relacionados a Aprendizagem de Máquina quanto Teoria de jogos, porém, não consideram em seus cenários o uso da Névoa e mobilidade dos usuários simultaneamente em sua avaliação. Uma área de pesquisa a ser realizada a partir do desenvolvimento dessa dissertação pode ser a avaliação e proposta de abordagens baseadas no uso dessas técnicas no contexto de usuários móveis na Computação na Névoa;
3. *É possível otimizar a transmissão dos dados no processo de migração da aplicação do usuário? Se sim, qual o impacto desta abordagem na qualidade de serviço oferecida ao usuário?* Apesar das definições de *quando* e *para onde* o processo de migração das aplicações dos usuários deva ser efetuado serem de suma importância para assegurar uma qualidade de serviço satisfatória para o usuário, um estudo acerca de *como* este processo de migração será realizado pode ser investigado neste cenário. Um estudo acerca da otimização da transmissão dos dados do usuário, tanto do ponto de vista do usuário quanto do provedor dos recursos pode acarretar em uma diminuição do consumo dos recursos de rede e no tempo de migração dos dados, o que é desejável por todos os envolvidos;
4. *Em quais condições o uso da Névoa ainda se mostra vantajoso se comparado com a Nuvem?* Apesar do uso da Névoa apresentar muitos benefícios comparados a Nuvem, como baixa latência, alguns cenários podem comprometer essas características.

Em um contexto onde todas as *Cloudlets* instaladas ao redor do usuário estão indisponíveis devido a alta demanda de outros usuários, a aplicação de novos usuários acaba sendo enviada para nós mais distantes. A partir de quais condições o cenário onde se prioriza o envio da aplicação do usuário para nós mais distantes se mostra mais vantajoso, tanto em termos de latência quanto do volume de dados trafegados na rede, se comparados com o uso da Nuvem? Um cenário de estudo derivado do desenvolvimento deste trabalho pode ser a avaliação do impacto da alta mobilidade de usuários móveis nesse contexto de definição do local de execução das aplicações, considerando dinamicamente as opções entre a Nuvem e a Névoa.

Referências Bibliográficas

- [1] David Perez Abreu, Karima Velasquez, Marilia Curado, and Edmundo Monteiro. A resilient internet of things architecture for smart cities. *Annals of Telecommunications*, 72(1-2):19–30, 2017.
- [2] Ejaz Ahmed, Abdullah Gani, Mehdi Sookhak, Siti Hafizah Ab Hamid, and Feng Xia. Application optimization in mobile cloud computing: Motivation, taxonomies, and open challenges. *Journal of Network and Computer Applications*, 52:52–68, 2015.
- [3] Sara Ayoubi, Noura Limam, Mohammad A Salahuddin, Nashid Shahriar, Raouf Boutaba, Felipe Estrada-Solano, and Oscar M Caicedo. Machine learning for cognitive network management. *IEEE Communications Magazine*, 56(1):158–165, 2018.
- [4] Daniel Macêdo Batista, Alfredo Goldman, Roberto Hirata, Fabio Kon, Fabio M Costa, and Markus Endler. Interscity: Addressing future internet research challenges for smart cities. In *Network of the Future (NOF), 2016 7th International Conference on the*, pages 1–6. IEEE, 2016.
- [5] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. Sumo – simulation of urban mobility: An overview. In SINTEF & University of Oslo Aida Omerovic, RTI International Research Triangle Park Diglio A. Simoni, and RTI International Research Triangle Park Georgiy Bobashev, editors, *SIMUL 2011*. ThinkMind, October 2011.
- [6] Luiz Fernando Bittencourt, Márcio Moraes Lopes, Ioan Petri, and Omer F Rana. Towards virtual machine migration in fog computing. In *P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on*, pages 1–8. IEEE, 2015.
- [7] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, pages 13–16. ACM, 2012.
- [8] Raouf Boutaba, Mohammad A Salahuddin, Noura Limam, Sara Ayoubi, Nashid Shahriar, Felipe Estrada-Solano, and Oscar M Caicedo. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. *Journal of Internet Services and Applications*, 9(1):16, 2018.

- [9] Charles C Byers and Patrick Wetterwald. Fog computing distributing data and intelligence for resiliency and scale necessary for iot: The internet of things (ubiquity symposium). *Ubiquity*, 2015(November):4, 2015.
- [10] Visual Networking Index Cisco. Global mobile data traffic forecast update, 2015–2020 white paper. *Document ID*, 958959758, 2016.
- [11] Lara Codeca, Raphaël Frank, and Thomas Engel. Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research. In *Vehicular Networking Conference (VNC), 2015 IEEE*, pages 1–8. IEEE, 2015.
- [12] IBM ILOG CPLEX. 12.7, user’s manual for cplex, 2016.
- [13] NIST Draft. Working definition of cloud computing v15.
- [14] Andre S Gomes, Torsten Braun, and Edmundo Monteiro. Enhanced caching strategies at the edge of lte mobile networks. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops*, pages 341–349. IEEE, 2016.
- [15] Andre S Gomes, Vitor Fonseca, Bruno Sousa, David Palma, Paulo Simoes, Edmundo Monteiro, and Luis Cordeiro. A mobile follow-me cloud content caching model. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium*, pages 763–766. IEEE, 2016.
- [16] Andre S Gomes, Bruno Sousa, David Palma, Vitor Fonseca, Zhongliang Zhao, Edmundo Monteiro, Torsten Braun, Paulo Simoes, and Luis Cordeiro. Edge caching with mobility prediction in virtualized lte mobile networks. *Future Generation Computer Systems*, 70:148–162, 2017.
- [17] Diogo M Gonçalves, Luiz Fernando Bittencourt, and Edmundo Roberto Mauro Madeira. Migração proativa de máquinas virtuais para aplicações móveis na computação em névoa. In *Simpósio Brasileiro de Redes de Computadores (SBRC)*, volume 36, 2018.
- [18] Diogo Machado Gonçalves, Karima Velasquez, Marilia Curado, Luiz Fernando Bittencourt, and Edmundo Roberto Mauro Madeira. Proactive virtual machine migration in fog environments. In *Computers and Communication (ISCC), 2018 IEEE Symposium on*. IEEE, 2018.
- [19] Keerthana Govindaraj and Alexander Artemenko. Container live migration for latency critical industrial applications on edge computing. In *2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA)*, volume 1, pages 83–90. IEEE, 2018.
- [20] Harshit Gupta, Amir Vahid Dastjerdi, Soumya K Ghosh, and Rajkumar Buyya. ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments. *Software: Practice and Experience*, 47(9):1275–1296, 2017.

- [21] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. Towards wearable cognitive assistance. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 68–81. ACM, 2014.
- [22] Yuxuan Jiang, Zhe Huang, and Danny HK Tsang. Challenges and solutions in fog computing orchestration. *IEEE Network*, 32(3):122–129, 2018.
- [23] Warley Júnior, Adriano Henrique, and Kelvin Lopes. Avaliação de desempenho da técnica de offloading computacional em nuvens móveis. *wperformance*, 2016.
- [24] Joelle Klaimi, Sidi-Mohammed Senouci, and Mohamed-Ayoub Messous. Theoretical game approach for mobile users resource management in a vehicular fog computing environment. In *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 452–457. IEEE, 2018.
- [25] Neeraj Kumar, Sherali Zeadally, Naveen Chilamkurti, and Alexey Vinel. Performance analysis of bayesian coalition game-based energy-aware virtual machine migration in vehicular mobile cloud. *IEEE Network*, 29(2):62–69, 2015.
- [26] Márcio Moraes Lopes et al. Arquitetura e mecanismos para migração de máquinas virtuais na computação em névoa. 2017.
- [27] Márcio Moraes Lopes, Wilson A. Higashino, Miriam A.M. Capretz, and Luiz Fernando Bittencourt. Myifogsim: A simulator for virtual machine migration in fog computing. In *Companion Proceedings of the 10th International Conference on Utility and Cloud Computing*, UCC '17 Companion, pages 47–52, New York, NY, USA, 2017. ACM.
- [28] Francesco Malandrino, Claudio Casetti, Carla-Fabiana Chiasserini, and Marco Fiore. Content download in vehicular networks in presence of noisy mobility prediction. *IEEE Transactions on Mobile Computing*, 13(5):1007–1021, 2014.
- [29] Ahmad M Mustafa, Omar M Abubakr, Omar Ahmadien, Ahmed Ahmedin, and Bassem Mokhtar. Mobility prediction for efficient resources management in vehicular cloud computing. In *Mobile Cloud Computing, Services, and Engineering, 5th IEEE International Conference on*, pages 53–59. IEEE, 2017.
- [30] Tahmid Nabi, Pranjal Mittal, Pooria Azimi, Danny Dig, and Eli Tilevich. Assessing the benefits of computational offloading in mobile-cloud applications. In *Proceedings of the 3rd International Workshop on Mobile Development Lifecycle*, pages 17–24. ACM, 2015.
- [31] Carlo Puliafito, Enzo Mingozzi, and Giuseppe Anastasi. Fog computing for the internet of mobile things: issues and challenges. In *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*, pages 1–6. IEEE, 2017.

- [32] Tarek K Refaat, Burak Kantarci, and Hussein T Mouftah. Dynamic virtual machine migration in a vehicular cloud. In *2014 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6. IEEE, 2014.
- [33] Tarek K Refaat, Burak Kantarci, and Hussein T Mouftah. Virtual machine migration and management for vehicular clouds. *Vehicular Communications*, 4:47–56, 2016.
- [34] Rodrigo Roman, Javier Lopez, and Masahiro Mambo. Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, 2016.
- [35] Eduardo Felipe Zambom Santana, Ana Paula Chaves, Marco Aurelio Gerosa, Fabio Kon, and Dejan S Milojicic. Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Computing Surveys (CSUR)*, 50(6):78, 2017.
- [36] Mahadev Satyanarayanan, Paramvir Bahl, Ramón Caceres, and Nigel Davies. The case for vm-based cloudlets in mobile computing. *IEEE pervasive Computing*, 8(4):14–23, 2009.
- [37] Wonkwang Shin, Byoung-Yoon Min, and Dong Ku Kim. Vehicaching: Embracing user request on vehicle route with proactive data transportation. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE, 2015.
- [38] Olena Skarlat, Matteo Nardelli, Stefan Schulte, and Schahram Dustdar. Towards qos-aware fog service placement. In *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, pages 89–96, May 2017.
- [39] Tarik Taleb, Sunny Dutta, Adlen Ksentini, Muddesar Iqbal, and Hannu Flinck. Mobile edge computing potential in making cities smarter. *IEEE Communications Magazine*, 55(3):38–43, 2017.
- [40] Tarik Taleb and Adlen Ksentini. Follow me cloud: interworking federated clouds and distributed mobile networks. *IEEE Network*, 27(5):12–19, 2013.
- [41] Karima Velasquez, David Perez Abreu, Marilia Curado, and Edmundo Monteiro. Service placement for latency reduction in the internet of things. *Annals of Telecommunications*, 72(1):105–115, Feb 2017.
- [42] Karima Velasquez, David Perez Abreu, Diogo Goncalves, Luiz Bittencourt, Marilia Curado, Edmundo Monteiro, and Edmundo Madeira. Service orchestration in fog environments. In *2017 IEEE 5th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 329–336. IEEE, 2017.
- [43] Hong Yao, Changmin Bai, Deze Zeng, Qingzhong Liang, and Yuanyuan Fan. Migrate or not? exploring virtual machine migration in roadside cloudlet-based vehicular cloud. *Concurrency and Computation: Practice and Experience*, 27(18):5780–5792, 2015.

- [44] Hong Yao, Changmin Bai, Deze Zeng, Qingzhong Liang, and Yuanyuan Fan. Migrate or not? exploring virtual machine migration in roadside cloudlet-based vehicular cloud. *Concurrency and Computation: Practice and Experience*, 27(18):5780–5792, 2015.
- [45] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*, pages 37–42. ACM, 2015.
- [46] Rong Yu, Yan Zhang, Stein Gjessing, Wenlong Xia, and Kun Yang. Toward cloud-based vehicular networks with efficient resource management. *IEEE Network*, 27(5):48–55, 2013.
- [47] Rong Yu, Yan Zhang, Huimin Wu, Periklis Chatzimisios, and Shengli Xie. Virtual machine live migration for pervasive services in cloud-assisted vehicular networks. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 540–545. IEEE, 2013.
- [48] Huaqing Zhang, Yanru Zhang, Yunan Gu, Dusit Niyato, and Zhu Han. A hierarchical game framework for resource management in fog computing. *IEEE Communications Magazine*, 55(8):52–57, 2017.
- [49] Yuan Zhang, Hao Liu, Lei Jiao, and Xiaoming Fu. To offload or not to offload: an efficient code partition algorithm for mobile cloud computing. In *Cloud Networking (CLOUDNET), 2012 IEEE 1st International Conference on*, pages 80–86. IEEE, 2012.