Revista Brasileira de Computação Aplicada, Novembro, 2019

DOI: 10.5335/rbca.v11i3.9247
Vol. 11, Nº 3, pp. 72–87
Homepage: seer.upf.br/index.php/rbca/index

ISSN 2176-6649

ARTIGO ORIGINAL

Análise de desempenho de uma arquitetura de *Fog Computing* para *Internet of Things* via estudos de caso

Performance analysis of a fog computing architecture for the Internet of Things via case studies

Marco Aurélio Spohn[1], and Fernanda da Silva Bonetti[1]

[1]Universidade Federal da Fronteira Sul (UFFS)

*marco.spohn@uffs.edu.br; fernandasbonetti@gmail.com

Recebido: 29/03/2019. **Revisado:** 14/08/2019. **Aceito:** 11/09/2019.

Resumo

Com o avanço tecnológico e barateamento de recursos computacionais, tem-se equipado objetos convencionais com alguma forma de computação e comunicação que possibilitam, inclusive, a sua conexão à Internet. A esse cenário heterogêneo de conectividade à Internet que se denomina *Internet of Things* (IoT - Intenet das Coisas). Convencionalmente, recorre-se à computação em nuvem (*cloud computing*) para viabilizar a operacionalização das soluções em IoT. No entanto, fatores que influenciam na conectividade dos dispositivos com a infraestrutura na nuvem podem afetar a implantação e interação com os dispositivos. Para trazer o controle mais próximo destes, introduziu-se o paradigma de *fog computing* que, em síntese, traz as principais facilidades da nuvem para próximo de onde realmente estão os dispositivos. Este trabalho compara a solução baseada em nuvem e a solução em *fog* via estudos de caso desenvolvidos em uma plataforma de IoT. Os resultados indicam que há benefícios significativos da solução baseada em *fog*, sobretudo em termos de redução do atraso de comunicação com os dispositivos.

Palavras-Chave: Internet das coisas; computação em nuvem; computação em neblina

Abstract

Technological advancements and cheaper computational resources have allowed the integration of computing and communication resources into regular objects, even making it possible to get them connected to the Internet. Such heterogeneous scenario with things connected to the Internet is usually referred to as the *Internet of Things* (IoT). Conventionally, cloud computing is employed to enable the deployment of IoT solutions. However, factors impacting the connectivity of devices to the infrastructure in the cloud can affect device deployment and interaction. To bring the control closer to the objects themselves, the *fog computing* paradigm was introduced so that it could bring some of the main cloud capabilities closer to where the devices really are. This work compares the cloud and fog based solutions via case studies designed on an IoT platform. The results indicate that there are significant benefits when employing the fog approach, especially in terms of reducing the communication delay with the objects/devices.

Keywords: Internet of things; cloud computing; fog computing

1 Introdução

Internet of Things (IoT, ou Internet das Coisas) é um paradigma em ascenção cuja principal potencialidade está na conectividade de dispositivos/objetos usuais à Internet, garantindo sua interoperabilidade. Quando se fala em dispositivos, torna-se importante salientar que essa definição engloba todas as coisas (de onde vem o termo em inglês, *Things*) que podem ser unicamente identificadas e conectadas à rede, incluindo-se os dispositivos de usuários (*user devices*), sensores, atuadores e outros quaisquer dispositivos finais (Kang et al., 2017).

O avanço da IoT é observado pelo aumento do acesso a dispositivos móveis e pela perpetuação da computação ubíqua, através da implantação de cidades-inteligentes (*smart-cities*), casas-inteligentes (*smart-homes*), redes de sensores e demais tecnologias que se tornam cada vez mais onipresentes (Vaquero and Rodero-Merino, 2014).

No estado atual das tecnologias de comunicação, predominam as arquiteturas centralizadas em *data centers*, com o compartilhamento de recursos através de serviços baseados em nuvem (*cloud-based services*), concentrando o poder de processamento e armazenamento nos *data centers*. Esse modelo de arquitetura pode se tornar insustentável para o contexto das IoTs, não havendo precedentes em termos da quantidade de dispositivos conectados, impondo alta carga de requisições frente aos desafios inerentes à latência de rede, largura de banda, confiabilidade e segurança (Laurent et al., 2018).

Para tratar eficientemente os desafios apresentados, pode-se adotar uma estrutura descentralizada, não-monolítica e mais próxima aos dispositivos finais. Nessa linha, surgiu o paradigma da *fog computing* ("Computação em Névoa"), constituindo-se uma extensão do modelo de *cloud* (nuvem). Nesse modelo, provê-se os dispositivos/objetos com maior capacidade de processamento, armazenamento e comunicação, a fim de diminuir o volume e a frequência de troca de informações oriúndas desses dispositivos. Desta forma, equipa-se melhor a borda da rede (com a adoção de *gateways*), possibilitando uma gerência mais acurada em termos de atuação sobre os objetos e de comunicação da borda com a nuvem (Laurent et al., 2018). Ou seja, no modelo baseado estritamente na nuvem, a inteligência dos dispositivos está centrada em um único ponto e resultando, sobretudo, em altas latências e mesmo indisponibilidade de atuação e controle sobre os objetos.

Este trabalho traz como principal contribuição científica uma análise comparativa entre a solução baseada em nuvem e a solução em *fog* via estudos de caso desenvolvidos em uma plataforma de IoT. Destaca-se que são infinitas as possibilidades de configurações e cenários de aplicações de IoT; no entanto, os resultados obtidos indicam que há benefícios significativos da solução baseada em *fog*, sobretudo em termos de redução do atraso de comunicação com os dispositivos finais.

O restante desse artigo está organizado da seguinte forma: a Seção 2 apresenta conceitos básicos de IoT, enquanto as Seções 3 e 4 apresentam os fundamentos de computação em nuvem e *fog computing*, respectivamente; a Seção 5 trata sobre plataformas de desenvolvimento em IoT focando, principalmente, na plataforma Kaa utilizada nos estudos de caso; a Seção 6 apresenta detalhadamente os estudos de caso e os principais resultados obtidos; a Seção 7 apresenta alguns trabalhos relacionados e, por final, a Seção 8 apresenta as principais conclusões deste trabalho.

2 *Internet of Things*

Há muitas especulações sobre como será a *Internet* no futuro. Baseando-se na visão atual da estrutura e o crescimento do número de dispositivos conectados, a *Internet* do futuro abrigará bilhões de dispositivos pessoais (como *Smartphones* e demais *Gadgets*), incluindo-se demais objetos do cotidiano que serão integrados e interconectados. Esses objetos ganharão características de processamento, comunicação em rede, percepção e atuação, análise e armazenamento. Como principal resultado, a computação estará presente o tempo todo e em todos os ambientes (Raj and Raman, 2017a).

Essa capacidade de coleção e processamento de informações, tornará a rede um centro de conhecimento sobre ambientes e pessoas. Serão sistemas inteligentes que se intercomunicarão, seja no meio doméstico ou industrial. E em torno dessa ideia permeia o paradigma de IoT e, como todo novo paradigma, existem características e requisitos diferentes que precisam ser levados em conta para a sua sustentação. Para satisfação desses requisitos, as tecnologias atuais precisam evoluir e se adaptar às necessidades do paradigma.

Em um ambiente com um grande número de dispositivos conectados, com características únicas, é importante que as aplicações se adaptem para lidar com essas diferenças. De um lado, as aplicações hospedadas remotamente em máquinas com grande poderio computacional e, do outro, artefatos corriqueiros do dia-a-dia; muitas vezes, escassos em recursos computacionais (Raj and Raman, 2017a).

Outra característica relevante a ser destacada sobre o contexto das IoTs é seu potencial para constante produção de um grande volume de dados. Cidades inteligentes, tecnologias de monitoramento pessoal, sistemas de rastreamento e a própria indústria são bons exemplos de aplicações que produzem uma quantidade massiva de informações que precisam ser processadas, armazenadas e analisadas. A quantidade de informações geradas pelos dispositivos nesse contexto pode atingir a ordem de *Terabytes* a *Petabytes* (Raj and Raman, 2017a). Sendo assim, algumas aplicações podem necessitar se adequar ao conceito de *Big Data*, processando e analisando um grande volume de dados em tempo real.

Os serviços fornecidos pela IoT podem ser categorizados da seguinte forma (Al-Fuqaha et al., 2015): Serviços de Identificação, responsáveis por identificar os dispositivos do mundo real trazidos para o mundo digital; Serviços de Agregação de Informação, que coletam e sintetizam os dados obtidos através dos sensores dos dispositivos; Serviços de Colaboração e Inteligência que agem sobre os Serviços de Agregação de Informação e utilizam os dados para tomar decisões e reagir de

acordo com determinados cenários; e Serviços Ubíquos, que visam fornecer Serviços de Colaboração e Inteligência sempre que forem necessários, em qualquer lugar e para qualquer um que precise deles. O objetivo de toda aplicação de IoT é atingir o nível de Serviços Ubíquos.

A semântica na IoT refere-se à habilidade de extrair conhecimento dos dados de forma inteligente. Esse elemento abrange a descoberta de conhecimento, incluindo o reconhecimento e a análise dos dados e também a associação de demandas aos recursos apropriados. Para isso, utilizam-se técnicas da Web Semântica, como o *Resource Description Framework* (RDF) e a *Ontology Web Language* (OWL).

Por ser um paradigma ainda recente, há uma falta de padrões específicos, além de problemas relacionados a segurança das informações, tratamento de dados em tempo real e, numa perspectiva mais ampla, como equalizar o mercado de soluções de IoT considerando que cada nicho de aplicações possui requisitos específicos. Há uma movimentação para criação de consórcios para definir padrões abertos e industriais, com a finalidade de elencar tecnologias e tornar viável a sua implantação (Raj and Raman, 2017a).

Apesar de não haver um padrão específico para IoT, existem diversas propostas de modelos arquiteturais para IoT. Por exemplo, a *Industrial Internet Reference Architecture* (IIRA) (Consortium, n.d.) é uma arquitetura aberta baseada em padrões para os sistemas de *Industrial Internet of Things* (IIoT), desenvolvida pelo Consórcio de Internet Industrial (*Industrial Internet Consortium* – IIC). O projeto *Internet of Things – Architecture* (IoT-A) Project (n.d.), desenvolvido pelo *Seventh Framework Programme* (FP7), propõe a criação de um modelo de referência arquitetural para a IoT, bem como a definição de um conjunto de elementos fundamentais para estabelecer as bases de uma IoT ubíqua. O projeto denominado *P2413 - Standard for an Architectural Framework for the Internet of Things* (Association, n.d., 2016) foi iniciado por orientação da equipe de IoT da *IEEE Standards Association* e visa suprir as necessidades do mercado, através do desenvolvimento do cenário tecnológico da IoT. O objetivo principal do P2413 é o fornecimento de um *framework* arquitetural extensível e integrado para a IoT. A intenção do grupo de trabalho envolvido no projeto é entregar um *framework* comum para todos os domínios da IoT, a fim de aumentar a transparência arquitetural e o suporte a *benchmarking*, segurança e proteção. Além disso, o grupo de trabalho tem intenção de promover a interação entre domínios e, para tal, pretende estabelecer as descrições de vários domínios de IoT, criar definições de abstrações de domínio e identificar semelhanças entre os diferentes domínios de IoT.

De acordo com Raj and Raman (2017a), há um conjunto de tecnologias necessárias para a solidificação das IoTs, incluindo-se os paradigmas de nuvem, *fog* e *edge computing*, bem como modelos de comunicação, desenvolvimento de *middlewares*, miniaturização de eletrônicos, virtualização e conteinerização para flexibilizar a utilização de recursos de *hardware*. A criação de plataformas como serviço (*Platform as a Service*, PaaS) abrigadas em estruturas de nuvem facilita e otimiza a construção de soluções para esse fim. Dessa forma as estruturas de nuvem estão se tornando o centro das aplicações de IoT.

3 *Cloud computing*

A evolução do poder computacional e o barateamento de recursos culminaram no surgimento de um novo modelo de computação denominado de *cloud computing* (computação em/na nuvem). Nesse novo modelo, recursos como processamento e armazenamento podem ser contratados/reservados conforme sua demanda. Isso reduz os custos com infraestrutura (*i.e.*, os recursos são cobrados conforme utilizados), terceirizando-se os riscos e a manutenção de infraestrutura para os fornecedores de recursos de nuvem, além de facilitar a escalabilidade desses recursos (Zhang et al., 2010).

Segundo Mell and Grance (2011), *cloud computing* é um modelo que habilita acesso a recursos computacionais (*e.g.*, rede, servidores, armazenamento, aplicações e serviços) de forma ubíqua, conveniente e sob-demanda via rede (*e.g.*, Internet). Os recursos computacionais devem ser aprovisionados e liberados celeramente com o mínimo de esforços de gerenciamento ou interação com o provedor do serviço.

Segundo Zhang et al. (2010), existem quatro tipos principais de estruturas de nuvem, conforme a necessidade do cenário escolhido:

- Nuvens públicas:
  - Estrutura em que os recursos são fornecidos para o público em geral; porém, o controle sobre quesitos de dados, rede e aspectos de segurança é reduzido.
- Nuvens privadas:
  - São estruturas desenvolvidas para uso único e restrito a uma única organização. Ao contrário das estruturas públicas, possibilitam maior controle sobre dados, desempenho e são em geral mais confiáveis.
- Nuvens híbridas:
  - É uma combinação dos dois modelos anteriores, complementando as características de cada uma. Oferece um maior controle sobre recursos do que as públicas; porém, maior flexibilidade de expansão que as privadas.
- Nuvens virtuais privadas (*Virtual Private Clouds, VPCs)*:
  - Basicamente uma plataforma de virtualização sobre nuvens públicas. Além da virtualização de servidores e aplicações, permite virtualizar comunicação interna na rede.

Com a utilização de nuvens ao invés da estruturação de *data centers* privados, os recursos passam a ser vistos e utilizados como serviços. Dessa premissa, surgem os conceitos de Infraestrutura como Serviço (*Infrastructure-as-a-Service*, IaaS) que provê a alocação de recursos como máquinas virtuais, Plataforma como

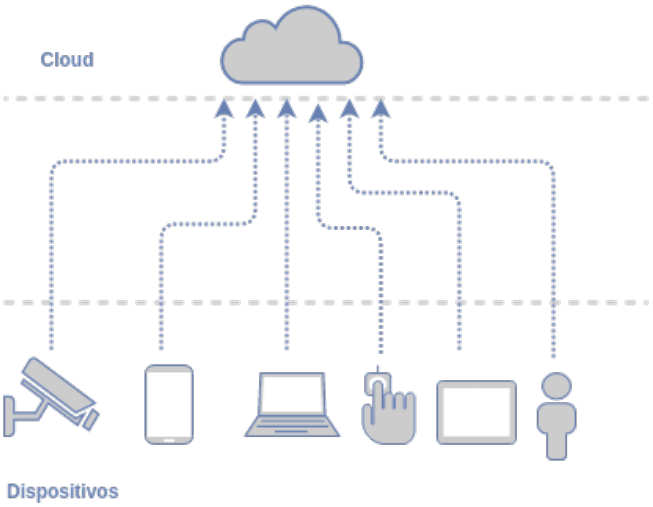

Figura 1: *Cloud* como núcleo das informações.

Serviço (*Platform-as-a-Service*, PaaS) oferecendo *frameworks* e funcionalidades, e *Software* como Serviço (*Software-as-a-Service*, SaaS) que provê aplicações específicas para serem utilizadas. Nesse contexto, praticamente qualquer recurso computacional pode ser disponibilizado como serviço, tornando as estruturas de nuvem o ponto central de processamento, armazenamento e análise (Fig. 1) (Zhang et al., 2010).

Para o cenário de IoTs, a implantação dos serviços na nuvem contribui para que o paradigma se torne cada vez mais presente. Para exemplificar, considerando-se aplicações de automação em ambientes domésticos, como as casas inteligentes (*Smart Homes*), sem a utilização de *cloud computing* seria necessário manter em cada casa uma estrutura de servidores que seria, além de cara, de difícil manutenção por usuários não especializados (Raj and Raman, 2017a).

Uma visão mais prática para esse cenário, seria manter a conectividade entre os dispositivos e a estrutura de nuvem, armazenando e processando dados de forma remota. Porém, a conectividade fora da rede local se torna um problema relevante, pois os dispositivos ficariam impossibilitados de acessar ou armazenar informações na nuvem em caso de desconexões. É nesse contexto que se tem estudado formas de trazer as capacidades da nuvem para mais próximo dos dispositivos.

4 *Fog Computing*

Lidando com estruturas centralizadas em nuvem, há alguns problemas presentes que precisam ser considerados para garantir a qualidade de serviço das soluções. As aplicações dependem fortemente dos provedores de serviços de nuvem em segurança e privacidade dos dados que estão sendo processados e/ou armazenados. Adicionalmente, pode haver restrições de migração de serviços e aplicações entre diferentes provedores (prática chamada de *vendor lock-in*) (Saharan and Kumar, 2015).

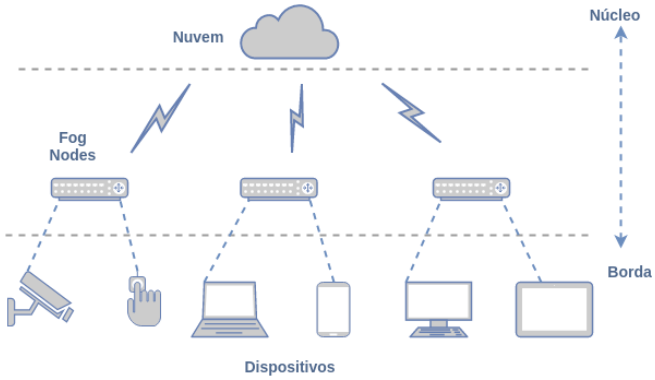

Figura 2: Aproximação aos dispositivos por nodos da *fog*.

Dentre os principais desafios encontrados quando se trata de IoT em um cenário de nuvem, pode-se citar Chiang and Zhang (2016):

a) Aplicações que requerem que a latência fim-a-fim seja ínfima;
b) Aplicações que geram uma grande quantidade de dados frequentemente, necessitando largura de banda diferenciada;
c) Dispositivos que possuem recursos computacionais extremamente limitados, cuja interação direta com a nuvem pode requerer algum processamento adicional como, por exemplo, operações criptográficas;
d) Serviços ininterruptos com conexões irregulares à nuvem podem apresentar perda de dados. Um exemplo citado é uma plataforma de petróleo, distante da costa, com apenas conexão via satélite (com frequente indisponibilidade) para atender uma aplicação de coleta de dados realizada com uma determinada frequência e subsequente envio à nuvem.

Considerando-se esses desafios, a Cisco Systems™ introduziu o paradigma de *fog computing* (Cisco Systems, 2015), correspondente a capacidade de aproximar as características de uma estrutura de nuvem aos dispositivos que geram e atuam sobre os dados. A presença de uma estrutura que realize a análise dos dados mais próxima da origem da coleta, diminui latência e o tráfego na rede, permitindo também maior controle sobre as informações que saem da rede local. A princípio, qualquer dispositivo que ofereça recursos de processamento, armazenamento e comunicação em rede é candidato a ser um nodo da *fog*.

A *fog* permite rapidez de acesso aos dados para ambos os sentidos, tanto para a nuvem quanto aos dispositivos conectados; ou seja, ela não funciona como um substituto da nuvem e sim como uma extensão dessa estrutura. As vantagens proporcionadas pela *fog* são relacionadas a sua implementação na borda da rede e a proximidade às fontes dos dados. Adicionalmente, amplia-se o potencial de escalabilidade na quantidade de dispositivos e volume de dados, já que se descentraliza o processamento e análise dos mesmos (Saharan and Kumar, 2015).

Parâmetros	***Cloud Computing***	***Fog Computing***
Localização dos nodos-servidor	Juntamente à Internet	Borda da rede local
Distância cliente-servidor	Múltiplos saltos	Único salto
Latência	Alta	Baixa
Delay Jitter	Alto	Muito Baixo
Segurança	Menos segura, indefinida	Mais segura, pode ser definida
Consciência de localização	Não	Sim
Vulnerabilidade	Probabilidade alta	Probabilidade muito baixa
Distribuição Geográfica	Centralizada	Densa e distribuída
Nº de nodos-servidor	Poucos	Muitos
Interacões em tempo real	Suportadas	Suportadas
Tipo de Conexão *Last Mile*	*Leased Line*	*Wireless*
Mobilidade	Suporte Limitado	Suportada

Tabela 1: Comparação entre nuvem e *fog* em diferentes aspectos. Fonte: (Saharan and Kumar, 2015).

A *fog computing* pode atuar como uma ponte entre os dispositivos inteligentes e os serviços de computação e armazenamento em nuvem (Al-Fuqaha et al., 2015). Através da *fog computing*, é possível estender os serviços da computação em nuvem até os dispositivos na borda da rede. Em razão da sua proximidade com os usuários finais, em comparação com os *data centers* da nuvem, a *fog computing* tem potencial para oferecer serviços com melhor desempenho. Geralmente, há uma diferença significativa de escala entre a computação em nuvem e a *fog computing*, de forma que a computação em nuvem tem uma capacidade de computação, armazenamento e comunicação massiva em comparação com a *fog computing*.

Considerando-se que as estruturas de *fog* e nuvem são complementares, algumas soluções podem ser fornecidas pela nuvem enquanto outras conseguem melhor desempenho fazendo uso da *fog* (Chiang and Zhang, 2016). Uma breve comparação entre características dessas duas estruturas é apresentada na Tabela 1.

A utilização da *fog computing* pode ser uma ótima escolha para os desenvolvedores de soluções para IoT, por conta das seguintes características (Raj and Raman, 2017b):

- Localização, pois os recursos da *fog computing* estão localizados entre os dispositivos inteligentes e os *data centers* da nuvem, proporcionando um melhor desempenho;
- Distribuição, pois a *fog computing* baseia-se em *"micro" centers* com capacidades de armazenamento, processamento e comunicação limitadas em comparação com a nuvem e, por isso, é possível implantar muitos *"micro" centers* próximos aos usuários finais, por um custo reduzido em relação ao custo de implantação de *data centers* da nuvem;
- Escalabilidade, pois sua estrutura permite que sistemas de IoT sejam mais escaláveis, de maneira que a medida em que o número de usuários finais aumenta, aumenta também o número de *"micro" centers* implantados, o que é inviável realizar com os *data centers* da nuvem por conta do custo elevado;
- Densidade de dispositivos, pois a *fog computing* auxília no fornecimento de serviços resistentes e replicados;
- Suporte à mobilidade, pois seus recursos agem como uma "nuvem móvel", dado que está próximo aos usuários finais;
- Tempo real, pois tem potencial para fornecer um melhor desempenho para os serviços interativos de tempo real;
- Padronização, pois a *fog computing* pode interoperar com diversos fornecedores de computação em nuvem;
- Análise agilizada, pois os recursos da *fog computing* podem realizar a agregação de dados para enviar dados parcialmente processados, ao invés de dados brutos, para *data centers* na nuvem, para que recebam processamento complementar.

Sendo assim, a *fog computing* tem potencial para aumentar o desempenho geral das aplicações de IoT, pois tenta desempenhar parte dos serviços de alto nível oferecidos pela nuvem com recursos locais.

5 Plataformas de *Internet of Things*

Como já foi exposto anteriormente, a popularização da computação em nuvem possibilitou o acesso e troca de informações entre aplicações de IoT, impulsionando a criação de diferentes tipos de serviços. Um dos modelos de serviço que ganhou grande enfoque foi o PaaS, fornecendo estrutura para facilitar o desenvolvimento e implantação de aplicações centralizadas em nuvem, sem a necessidade de adaptar uma plataforma especí-

fica para cada fim (Saharan and Kumar, 2015).

Plataformas voltadas à criação de soluções em IoT precisam garantir alta adaptação às necessidades de cada aplicação, disponibilidade e escalabilidade [1] dos recursos. A fim de lidar com um número massivo de fluxo de dados, devem ser capazes de efetuar escalabilidade horizontal (*i.e.*, inclusão de mais instâncias) e escalabilidade vertical (*i.e.*, ampliação dos recursos alocados por instância) (Auger et al., 2017).

As plataformas também devem ter a capacidade de lidar e se adaptar a uma grande variedade de dispositivos, o que se torna possível graças a utilização de *middlewares*: interfaces de comunicação e gerência de diferentes componentes e serviços que possibilitam aos dispositivos o uso de um sistema comum (Salami and Yari, 2018). Alguns componentes necessários para que uma plataforma consiga conciliar dispositivos e serviços são (Salami and Yari, 2018):

a) Camada de conectividade: componente mais básico, garante a formatação de diferentes protocolos e formatos de dados em uma linguagem comum e uniforme aos dispositivos da plataforma;
b) Gerência dos dispositivos: garante que os dispositivos conectados estejam funcionando corretamente;
c) Armazenamento: é um ponto crítico da plataforma, pois garante que os dados recebidos pela camada de conectividade sejam armazenados para posterior processamento e visualização;
d) Camada de processamento: permite ações baseadas em eventos para realizar processamentos específicos com os dados recebidos pela plataforma;
e) Visualização dos dados: funcionalidade necessária para algumas aplicações que requerem que os dados sejam representados de forma gráfica para uma melhor análise.

Uma outra forma de caracterizar as plataformas de IoT é pela sua arquitetura, sendo elas centralizadas (os dados são coletados, armazenados e processados em uma instância central) ou distribuídas (todas as entidades envolvidas são capazes de coletar, processar e armazenar dados, sendo possível também a integração de instâncias centralizadas, como serviços baseados em nuvem) (Roman et al., 2013).

As seguintes categorias de plataformas de IoT foram identificadas por Zdravković et al. (2016):

- Plataformas de domínio específico, que facilitam cenários específicos de determinado domínio.
- Plataformas de tecnologia específica, que levam em conta apenas um conjunto específico de dispositivos. Geralmente, são plataformas fechadas, baseadas em dispositivos com tecnologia proprietária.
- Provedores de conectividade máquina a máquina (*Machine-to-Machine* – M2M), cujo recurso principal é a conectividade como um serviço e que têm como objetivo primário a aquisição e a análise de dados.
- *Middlewares* genéricos de larga escala, que fornecem uma gama completa de serviços de conectividade, mas também facilitam o desenvolvimento de aplicações, com base em dados coletados pelos dispositivos e transformados por ferramentas analíticas.
- Plataformas de serviços de apoio. Essas plataformas não oferecem conectividade M2M, então não são plataformas de IoT; entretanto, oferecem funcionalidades que podem ser úteis para cenários de IoT.

O desenvolvimento de plataformas de IoT é impulsionado pela necessidade de se facilitar a conectividade M2M, que vêm crescendo a uma taxa sem precedentes. Com a ascensão das plataformas de IoT, a interoperabilidade entre plataformas e a reutilização estão emergindo. Há situações em que plataformas de IoT de domínio específico são implantadas através da utilização de provedores de conectividade M2M. Da mesma forma, os primeiros casos de colaboração entre plataformas aparecem com soluções de interoperabilidade.

5.1 Kaa IoT *Platform*

A plataforma Kaa CyberVision (2018), desenvolvida pela CyberVision Inc., é um *middleware* de código-livre, disponibilizado segundo a licensa Apache 2.0, que permite o desenvolvimento de soluções fim-a-fim em IoT. Algumas das funcionalidades que a plataforma apresenta em sua versão comunitária (0.10.0) incluem a coleta de dados, configuração, troca de mensagens e registro dos dispositivos. Essas são as principais características que motivaram a escolha da plataforma Kaa, destacando-se que há um conjunto extenso de plataformas, sejam estas proprietárias ou livres de licenças comerciais.

Sua arquitetura é centralizada e dividida em três componentes principais (Fig. 3): *Kaa Server*, compreendendo os serviços de *back-end* da plataforma, gerência de aplicações, usuários e dispositivos; *Kaa Extensions*, que são módulos independentes que provêem algumas funcionalidades às aplicações como, por exemplo, serviço de notificações e criação de *logs*; e o *Kaa Endpoint SDK*, que é uma biblioteca para desenvolvimento de aplicações-cliente (CyberVision, 2018).

O Kaa permite que a implantação da plataforma seja realizada no modo *Single Node* (*i.e.*, uma única instância) ou em forma de *cluster* (múltiplas instâncias). Utilizando pelo menos três instâncias interconectadas no modo *cluster*, torna-se possível suportar um modo de alta-disponibilidade onde, em caso de falha de um nó, os nós remanescentes assumem sua função (CyberVision, 2018).

O desenvolvimento de aplicações para IoT é facilitado pela versatilidade da biblioteca do Kaa, que suporta pelo menos 13 plataformas e dispositivos distintos e quatro linguagens de programação.

6 Estudos de caso

Com o intuito de avaliar cenários contemplando os dois modelos em questão (*i.e.*, baseado em nuvem e baseado em névoa), os estudos de caso abrangem uma configu-

[1] Capacidade de um sistema crescer e se adaptar quando exposto a uma carga maior de trabalho.

Modelo	Memória (GB)	Processador físico	Velocidade de clock (GHz)
t2.micro	1	Família Intel Xeon	Até 3.3

Tabela 2: Especificações da Instância. Fonte: *Amazon Web Services* (2019)

Memória (GB)	Processador físico	Velocidade de clock (GHz)
8	Intel® Core™ i5-3470	3.20

Tabela 3: Especificações do hospedeiro da plataforma na rede local dos dispositivos.

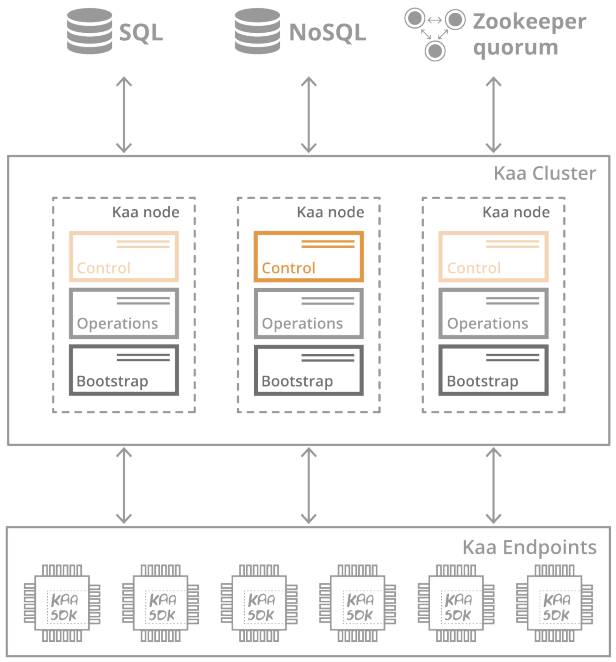

Figura 3: Arquitetura da Plataforma Kaa. Fonte: (CyberVision, 2018).

ração tipicamente centralizada e outra descentralizada. Para cada um dos cenários, reproduz-se as demandas correspondentes a um número variável de dispositivos.

O primeiro cenário foi concebido para analisar uma estrutura centralizada em nuvem. Para tanto, uma instância *Amazon Elastic Compute Cloud* (Amazon EC2) foi utilizada para hospedar uma instalação *single-node* da plataforma Kaa. As especificações técnicas da instância estão descritas na Tabela 2.

Com o intuito de reproduzir uma estrutura descentralizada de nuvem, utiliza-se um cenário contemplando uma instalação *single-node* da plataforma Kaa na rede local dos dispositivos finais, objetivando-se observar os impactos da relocação de boa parte do custo de processamento e comunicação próximo aos dispositivos (*i.e.*, na borda da infraestrutura). A Tabela 3 apresenta as principais especificações da máquina operando como *gateway* nesse cenário.

6.1 Métricas

Para analisar e quantificar propriedades dos protocolos envolvidos, estipula-se métricas de desempenho definidas de acordo com metodologias específicas de coleta, padronização de unidades de medida e outros critérios como o *framework* para métricas apresentado na RFC 2330 (V. Paxson et al., 1998). Dentre as práticas citadas neste documento, a metodologia de observância dessas métricas deve ser replicável, concreta e bem definida, sem nenhuma forma de viés entre tecnologias semelhantes e sem a indução de objetivos na medição de desempenho. Algumas dessas métricas de rede são parte integrante de Acordos de Nível de Serviço (ou, do inglês, *Service Level Agreement* - SLA) estabelecidos por fornecedores de serviços de comunicação.

A latência total (*round-trip delay*) é a medição do tempo total entre envio do primeiro *bit* de um pacote genérico P da origem até o destino até a recepção do último *bit* do pacote de resposta enviado pelo destinatário até a origem. A latência é um importante indicador de desempenho, principalmente para aplicações que precisam que o tempo de comunicação fim-a-fim seja ínfimo para funcionar de uma forma fluída. Latências altas podem ser um forte indício de congestionamento no caminho de comunicação, apesar das medições dependerem de muitas características do meio como, por exemplo, o caminho de envio de um pacote ser diferente do caminho de retorno. Adicionalmente, as medições obtidas em redes com serviços que oferecem certo nível de qualidade (*Quality of Service* - QoS) diferem de redes convencionais que ofertam apenas serviços de melhor esforço (*best effort*) (Almes et al., 1999).

Outra métrica importante é a taxa de transferência, que representa a quantidade total de dados transferidos durante um certo intervalo de tempo (*e.g.*, $\frac{bits}{s}$, ou bps). Taxas de transferência muito baixas são bons indicadores da existência de gargalos no(s) caminho(s) utilizado(s) para comunicação na rede. Em cenários de IoT, com uma alta carga de requisições, possuir uma taxa de transferência controlada é requisito fundamental para a fluidez das operações das aplicações.

6.2 Metodologia

A metodologia adotada está calcada no desenvolvimento de testes/experimentos que reproduzam o comportamento de aplicações reais de IoT. Esses testes visam a observância das métricas de desempenho de rede nos cenários definidos, para posterior análise.

Os testes compreendem uma aplicação relativa a um hipotético termostato, coletando amostras de temperatura e posteriormente enviando-as à plataforma/base central. A aplicação emula a leitura de temperatura de um sensor gerando uma amostra a cada segundo; porém, os envios são realizados em intervalos de 5

segundos. A duração total de cada teste é de cinco minutos. O objetivo deste teste é medir a latência média dos envios das amostras em ambos os cenários. As especificações gerais dos testes seguem nas próximas seções e os recursos de código para replicação dos testes estão disponíveis publicamente (da Silva Bonetti and Spohn, 2018).

6.2.1 *Experimentos*

Em soluções de IoT, bem como ambientes de sensores, monitoramento e coleta de dados, em caso de alteração de configurações e envio de ações para dispositivos através da rede, a latência da comunicação entre dispositivos e servidores pode ser um ponto crucial à implantação do sistema.

Para melhor visualização da importância da latência em alguns sistemas, pode-se, a fim de exemplificação, assumir um conjunto de sensores responsáveis por aferir a temperatura de elementos extremamente sensíveis às condições do ambiente (*e.g.*, calor, umidade, pressão, luminosidade). Nesse ambiente hipotético, as condições podem variar a uma taxa muito alta e pode haver interesse em armazenar essas informações com alta frequência, sem perda de dados, mantendo-se a acurácia na periodicidade dessas medições.

Para escolher um cenário ideal para a implementação dessa aplicação, pode-se realizar o armazenamento dos dados em uma estrutura fora da rede local dos sensores, bem como em uma estrutura mais próxima dos dispositivos. A quantidade de dispositivos interagindo com essa estrutura também pode repercutir no desempenho da aplicação; portanto, testes devem ser realizados variando-se a quantidade de dispositivos ativos.

Dado o cenário hipotético e utilizando a plataforma Kaa como base para realização desses testes, desenvolveu-se uma aplicação utilizando as bibliotecas fornecidas pela plataforma. Essa aplicação realiza a geração de amostras pseudo-aleatórias a cada segundo e as envia para serem armazenadas a cada cinco segundos. Para cada envio, calcula-se a latência correspondente, computada a partir de alterações no código da biblioteca Kaa SDK. A duração total do teste é de cinco minutos. O cenário compreende duas modalidades de teste (Fig. 4):

- Teste 1: latência média das amostras com apenas um dispositivo (*Endpoint*), enviando-as para as estruturas de *Fog* e *Cloud*.
- Teste 2: latência média das amostras com 100 dispositivos, enviando-as para as estruturas de *Fog* e *Cloud*.

O processamento das amostras geradas pelos dispositivos também pode impactar no desempenho do sistema como um todo. Para a observância do custo relativo a esse processamento, desenvolveu-se uma aplicação baseada nas bibliotecas de desenvolvimento da plataforma Kaa. Essa aplicação envia um conjunto de 10 amostras de temperatura geradas de forma pseudo-aleatória. Na estrutura de servidor, esses dados são recebidos e armazenados. Para a realização do cálculo da média da temperatura das amostras, consulta-se estas diretamente do banco de dados da plataforma (MongoDB), realiza-se a filtragem dos dados relevantes ao cálculo e a montagem da resposta através de um *script*. Após isso, a média das amostras é enviada até o dispositivo que solicitou esse processamento. A aferição do tempo total de envio, processamento e retorno dessas informações ao dispositivo origem é o objetivo desse teste.

Em síntese, tem-se os seguintes testes (Fig. 5):

- Teste 1: latência total para processar 10 amostras com apenas um dispositivo (*Endpoint*) enviando-as para as estruturas de *Fog* e *Cloud*.
- Teste 2: latência total para processar 10 amostras com 100 dispositivos, nas estruturas de *Fog* e *Cloud*.

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	35,308	3,8850
Fog	8,754	0,9165

Tabela 4: Resultados do teste com um único dispositivo.

6.3 Resultados e análise

Após a execução dos experimentos propostos e a coleta dos resultados, é possível realizar uma comparação entre o uso das estruturas propostas baseadas na plataforma Kaa. Nas seções seguintes, a discussão de cada resultado é realizada separadamente.

6.3.1 *Envio de amostras para a plataforma*

6.3.1.1 Teste com um único dispositivo. Nesse teste as amostras são enviadas tanto para a implantação da plataforma Kaa na modalidade de nuvem como para a modalidade em *Fog*. Foi realizado o cálculo da latência do envio de cada grupo de amostras, como especificado anteriormente. A Tabela 4 apresenta uma comparação entre as duas modalidades avaliadas.

Para melhor compreensão dos resultados mostrados na Tabela 4 e no gráfico da Fig. 6, é preciso explicitar as condições que podem ter afetado os resultados. Como o ambiente de nuvem empregado localiza-se a uma quantia considerável de saltos em comparação com a estrutura de *Fog*, prevê-se que a latência entre os dispositivos e a estrutura de nuvem seja superior. O desvio padrão das amostras correspondentes à estrutura de nuvem é 4, 24 vezes superior ao desvio padrão das amostras enviadas à *Fog*. Esse desvio padrão elevado deve-se ao pico de latência registrado aos 95 segundos de teste, quando a latência das amostras atingiu 62, 635 milissegundos. Existem diversas condições que podem ter acarretado essa anomalia nas amostras, podendo ser um conjunto de fatores do meio, assim como da própria estrutura de nuvem.

Como descrito anteriormente, utiliza-se uma instância t2.micro da *Amazon Web Services* (AWS) para abrigar a estrutura do Kaa, localizada na região leste da América do Sul (*South America East* – São Paulo), estando cerca de 25 saltos da rede que abriga os dispositivos que originam as amostras. Salienta-se que essa instância é parte do nível gratuito de serviços, sendo mais limitada

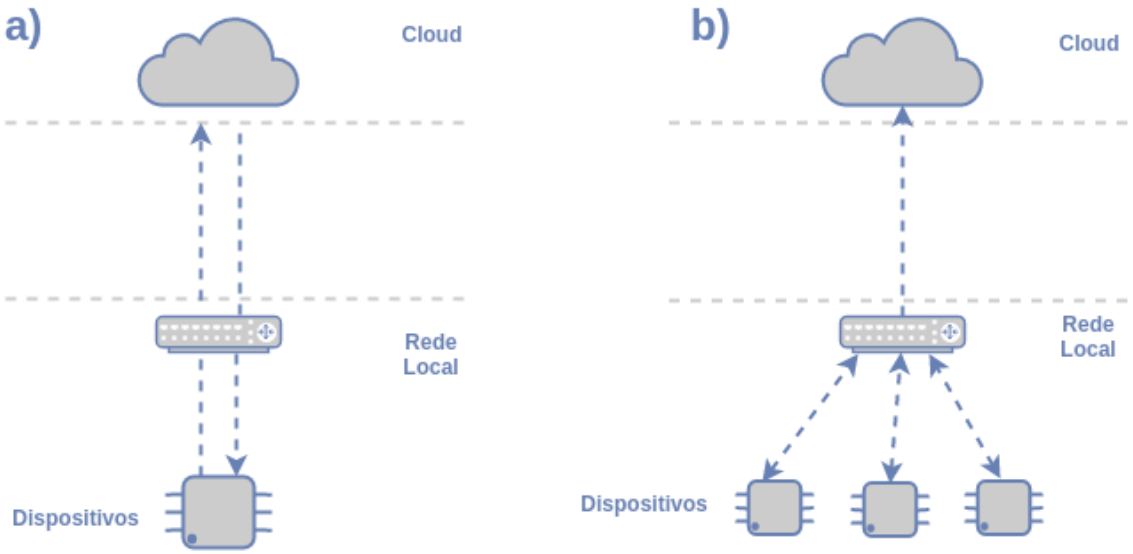

Figura 4: Cenários da Aplicação: 1 (a) e 100 (b) dispositivos para as estruturas.

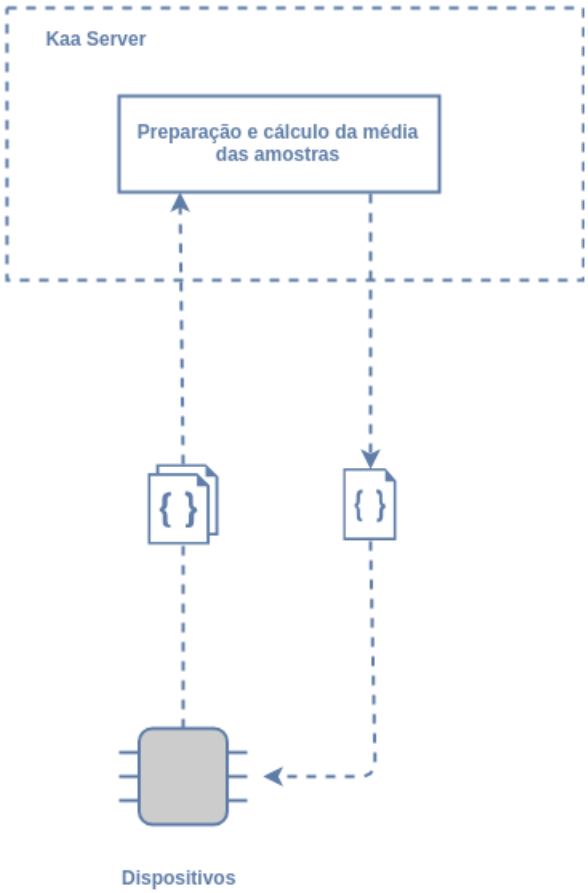

Figura 5: Cenário do processamento das amostras.

em comparação às outras instâncias comercialmente disponíveis na AWS.

Wang and Ng (2010) realizaram um estudo sobre o impacto de instâncias Amazon EC2 mais limitadas em testes de desempenho. Segundo os autores, as instâncias Amazon EC2 utilizam virtualização *Xen* para a gerência dos servidores, possibilitando a existência de várias máquinas virtuais alocadas em um único servidor físico, compartilhando-se recursos computacionais. Torna-se relevante destacar que, especificamente para o uso das interfaces de rede, essas máquinas virtualizadas (denominadas *Guest domains*) precisam se comunicar através de uma outra máquina virtual com maior privilégio (denominada *Driver Domain*) para acessar dispositivos físicos. Por exemplo, em operações de envio de pacotes, os mesmos são enviados das *Guest Domains* para a *Driver Domain* através de suas interfaces virtuais e, posteriormente, para a rede em si (Wang and Ng, 2010).

Segundo Wang and Ng (2010), instâncias menores (*small* e *micro*) estão sempre compartilhando processadores com outras instâncias e, através de políticas de controle computacional, as instâncias menores conseguem atingir apenas 50% do processador físico (mesmo na ausência de outras instâncias ativas no servidor). As principais conclusões dos autores concernentes aos problemas e possíveis causas são:

- Baixa taxa de transmissão (TCP):
  - Como não foram detectadas perdas ou retransmissões de pacotes por congestionamento da rede e nenhuma política de limitação de taxa, é provavel que o compartilhamento de recursos no escalonamento de instâncias influencie no comportamento da rede, visto que, durante o envio de pacotes a instância que está transmitindo pode perder o processador.

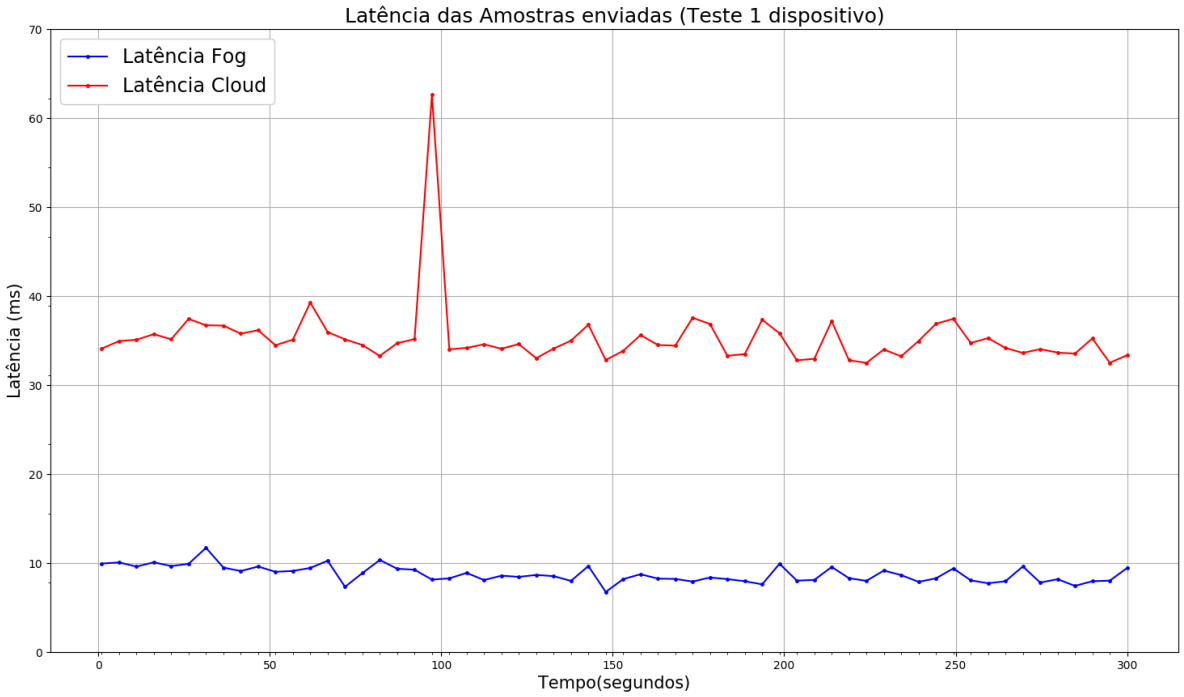

Figura 6: Representação gráfica da latência de cada envio.

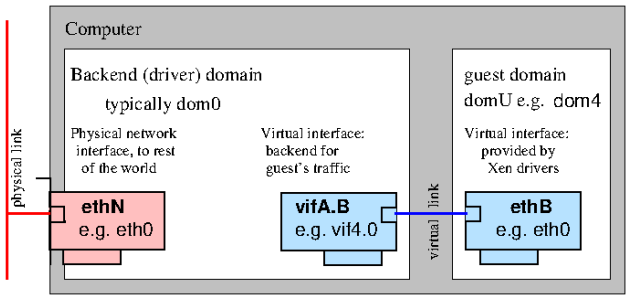

Figura 7: Exemplo da estrutura de comunicação com virtualização *Xen*. Fonte: (*Xen Networking*, 2011)

- Alta oscilação no atraso fim-a-fim:
  - Causadas pela formação de filas de pacotes (*buffering*) no *Driver Domain* das instâncias.

6.3.1.2 Teste com 100 dispositivos. Nesse teste, emprega-se a mesma premissa da avaliação anterior, mas com a utilização de 100 dispositivos transmitindo as amostras para as estruturas abrigando a plataforma Kaa. De forma semelhante à avaliação anterior, observou-se oscilações que influenciaram a medida de latência em momentos específicos. A seguir, apresenta-se os resultados completos bem como após a remoção dos *outliers* das amostras.

Com auxílio do gráfico da Fig. 8 é possível observar as amostras que possuem um afastamento drástico do resto da série. Com a remoção dos *outliers* do conjunto de amostras (de ambos os conjuntos), obtém-se uma visão mais justa dos resultados (Tabela 6).

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	35,412	18,170
Fog	10,090	9,209

Tabela 5: Resultados do teste de 100 dispositivos (sem remoção de *outliers*).

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	31,588	2,787
Fog	9,501	7,981

Tabela 6: Resultados do teste com 100 dispositivos (com remoção de *outliers*).

Os resultados com a remoção dos *outliers* permitem uma visão mais uniforme do comportamento das estruturas de *Fog* e nuvem; porém, deve-se levar em conta que com a utilização de uma estrutura fora da rede local, anomalias podem ocorrer e acarretar em latências maiores.

6.3.2 *Envio e processamento de amostras*

6.3.2.1 Teste com único dispositivo. Na Tabela 7 é possível visualizar os resultados dos testes realizados com os dois cenários. Contrariando os testes que envolviam somente a latência do envio de amostras, quando adicionada a capacidade de processamento das mesmas, os resultados de ambos os cenários fica menos discre-

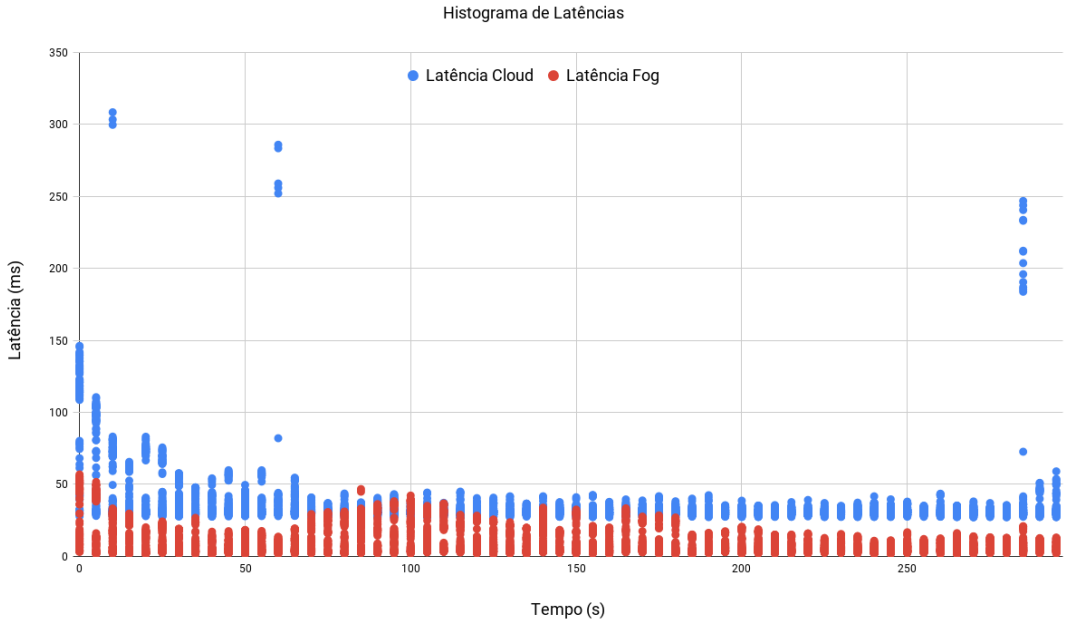

Figura 8: Latência com 100 dispositivos ativos (sem remoção de *outliers*).

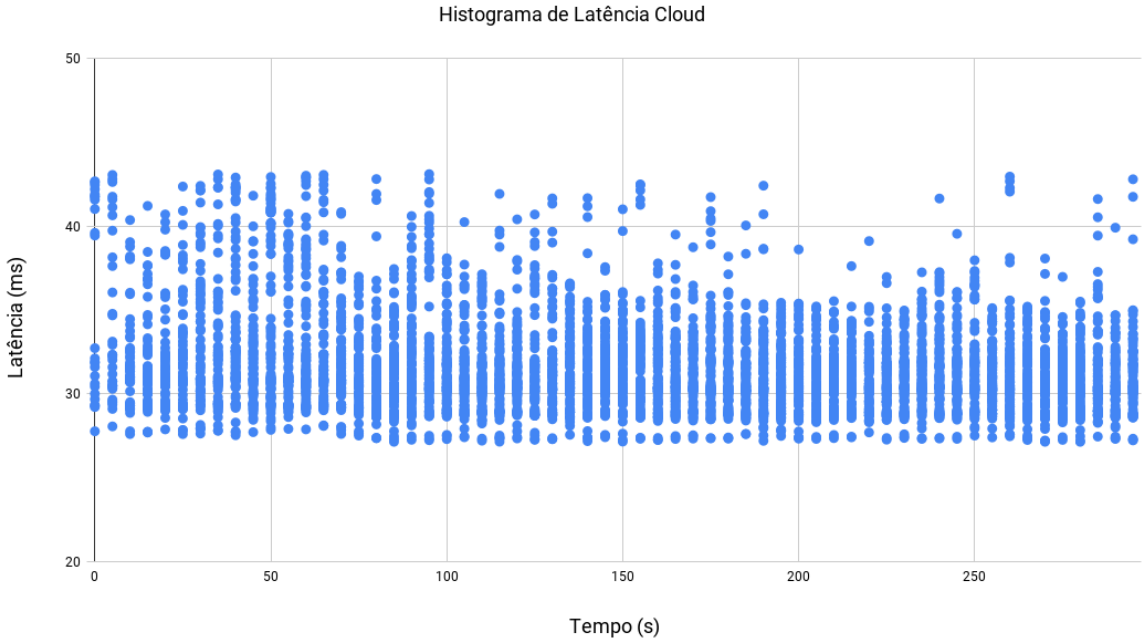

Figura 9: Latência 100 dispositivos para *Cloud* (após remoção de *outliers*).

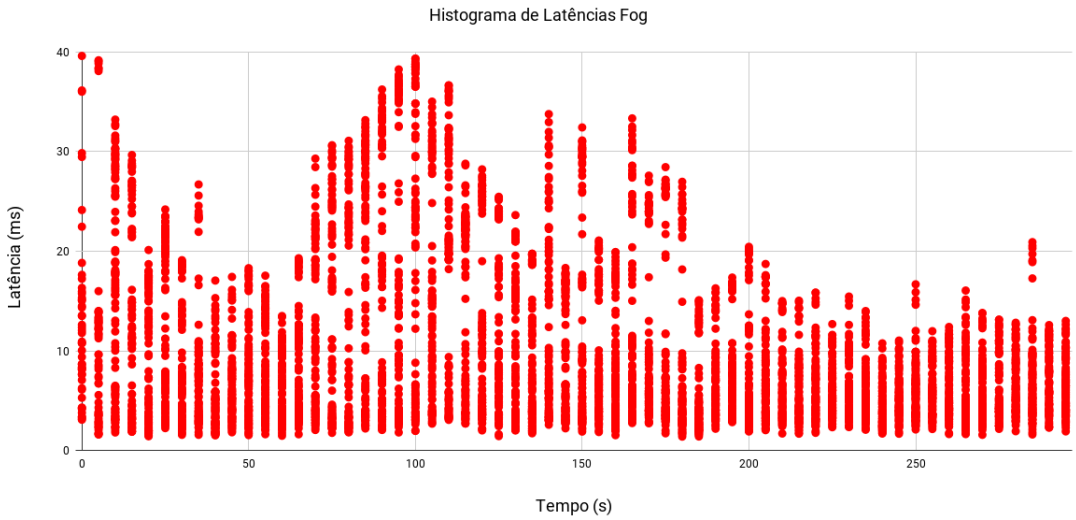

Figura 10: Latência 100 dispositivos para *Fog* (após remoção de *outliers*).

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	11.478,935	570,571
Fog	11.367,141	441,213

Tabela 7: Resultados do teste com envio e processamento de amostras.

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	14.766,909	1.232,360
Fog	12.660,712	529,560

Tabela 8: Resultados do teste com envio e processamento de amostras com 100 dispositivos.

pantes.

Em alguns testes (Fig. 11), a latência da estrutura de nuvem foi inferior a observada na *Fog*, mesmo com a nuvem tendo um *overhead* de rede muito maior do que a instância localizada na mesma rede local (*Fog*). A latência de processamento da estrutura de nuvem, em alguns momentos, é significativamente inferior ao seu próprio *overhead* de rede, resultando em uma latência total menor do que a latência observada na estrutura de *Fog*. Como já comentado anteriormente, uma estrutura virtualizada está em constante disputa pelos recursos da máquina que a abriga, havendo momentos em que a carga agregada de instâncias é reduzida, resultando em fatias de tempo de utilização dos recursos de forma mais frequente. No entanto, como já observado anteriormente, mesmo tendo que concorrer aos recursos de processamento, a instância virtualizada é executada sobre uma infraestrutura de *hardware* mais poderosa (por se tratar de uma estrutura para servidor) que a estrutura avaliada na modalidade de *Fog*.

6.3.2.2 Teste com 100 dispositivos. Os resultados dos testes de latência, contemplando processamento de amostras, podem ser observados na Tabela 8. Comparando com o teste anterior, que envolvia somente um dispositivo fonte, nesse novo cenário o desvio padrão das medições para a estrutura de nuvem praticamente dobra de valor.

Testes com maior número de dispositivos e, consequentemente, mais requisições, demonstram uma desvantagem da estrutura de nuvem proposta. A variação observada pode ser associada às condições da rede entre os dispositivos e a estrutura, bem como às próprias condições da instância utilizada para abrigar a plataforma. Para cenários onde o número de dispositivos pode crescer exponencialmente, a utilização de um ambiente centralizado sob influência de características de virtualização de instâncias, deve ser analisada levando-se em conta as variações de latência. Aplicações dependentes de um curto tempo de resposta podem ter seu desempenho prejudicado, ou mesmo impossibilitado, com a adoção da estrutura em nuvem.

Em ambientes reais de aplicações de IoT, espera-se que o número de dispositivos seja muito maior do que o apresentado em nossos cenários de teste. Por exemplo, em ambientes com diversidade de sensores e atuadores, como em cidades e casas inteligentes, esse número é facilmente superado e possui crescimento contínuo pela fácil integração de novos dispositivos às aplicações.

Ao se projetar aplicações de IoT com alta demanda, deve-se levar em conta todos os elementos que possam impactar negativamente o desempenho das aplicações. Quando hospedadas em ambientes centralizados, como os de *Cloud Computing*, torna-se maior o número de fatores que podem acarretar nessas interferências. Por não haver condições de garantias de serviço em redes de melhor esforço, como é o caso da Internet, tem-se como maior vilão as oscilações em níveis de congestionamento entre a rede dos dispositivos e a rede da provedora de serviços na nuvem.

Os recursos e potencialidades disponibilizadas para hospedar os serviços tem impacto direto no desempenho do sistema como um todo. Neste trabalho, optou-se pela utilização de uma estrutura mais limitada, tratando-se de uma instância gratuita fornecida

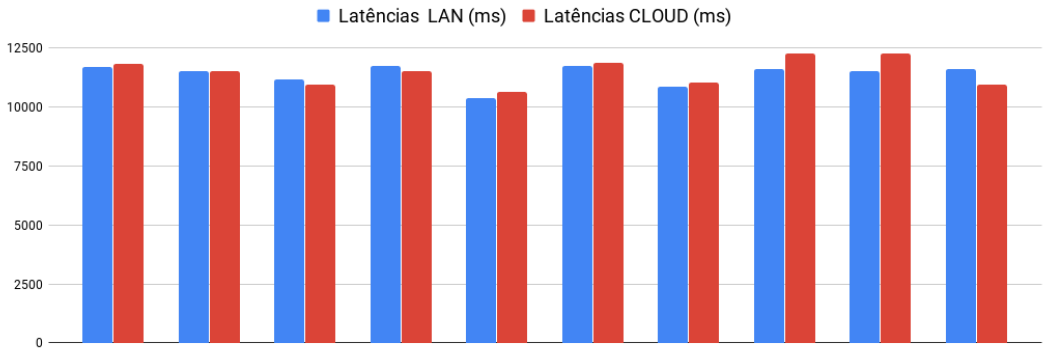

Figura 11: Latência dos testes com processamento (um dispositivo).

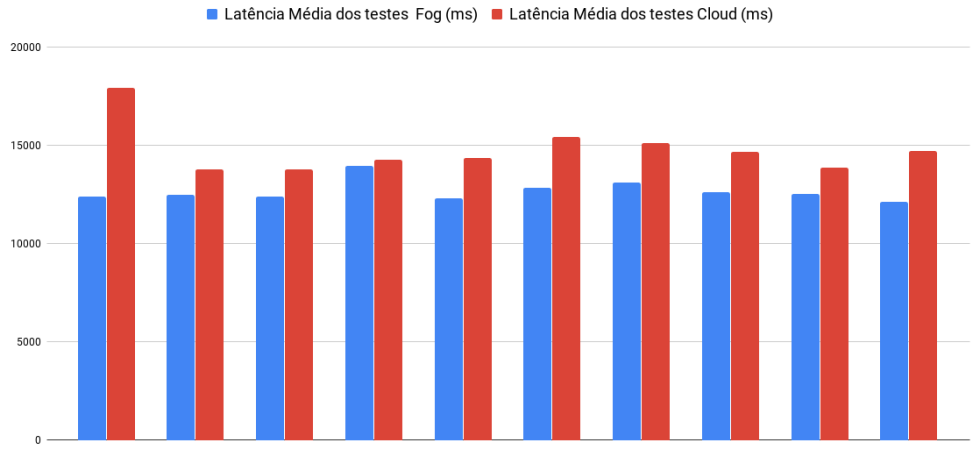

Figura 12: Latência dos testes com processamento e 100 dispositivos.

pela Amazon e com um nível de qualidade de serviço inferior às instâncias mais robustas e customizadas.

Com a utilização de uma estrutura localizada na rede local próxima aos dispositivos, incertezas relacionadas ao caminho da comunicação são minimizadas, dado o conhecimento das estruturas utilizadas e pela possibilidade de controlar como essa comunicação é realizada. A descentralização das capacidades da nuvem traz a diminuição do atraso de comunicação e também uma maior segurança para as informações que são enviadas dos dispositivos para as plataformas. A escolha por descentralizar a comunicação depende dos requisitos das soluções que serão desenvolvidas, sabendo-se que aplicações que não dependem de forma crucial do atraso de comunicação podem se beneficiar muito bem da estrutura em nuvem.

A presença de nodos de *Fog* abre um leque de cenários aplicáveis para IoT. Há a possibilidade de escolha do local de acesso e armazenamento das informações baseado na localização dos dispositivos, podendo-se diminuir a latência fim-a-fim armazenando e processando as informações na *Fog* e, periodicamente, replicando-as na estrutura de nuvem, permitindo o acesso às informações cruciais fora da infraestrutura local. Graças ao pré-processamento nos próprios dispositivos ou em *gateways* na rede local, tem-se a capacidade de estruturar os dados antes de serem armazenados na nuvem, filtrando-os e enviando somente dados que realmente são necessários. Desta forma, realiza-se uma economia de recursos de armazenamento e uma redução na frequência de envio de requisições para fora da rede, resultando em taxas de transmissão maiores.

7 Trabalhos relacionados

La et al. (2019) avaliam através de estudos de caso o emprego de mecanismos e soluções de inteligência como facilitadores da computação em névoa. O estudo não emprega uma plataforma específica de IoT, mas utiliza dispositivos OpenMote-CC2538 como nós sensores (executando o sistema Contiki-OS) e Raspberry-Pi 3 como *gateway*. Dentre as métricas avaliadas, tem-se atraso, consumo de energia, *Radio Duty Cycle* e taxa de entrega de pacotes. Os resultados indicam que a solução baseada em névoa propiciou uma redução no consumo de energia e na latência.

Giang et al. (n.d.) apresentam um estudo de caso em desenvolvimento de aplicações para computação em névoa baseadas na plataforma de código aberto *Distributed Node-RED* (DNR). Demonstra-se como aplicações podem ser decompostas e implantadas em uma infraestrutura geograficamente distribuída, bem como componentes existentes de *software* podem ser adaptados e reutilizados para participar de uma aplicação em névoa. A solução proposta não é implementada em uma plataforma de IoT específica, mas utiliza-se de simulações baseadas no simulador Omnet++ para demonstrar que o modelo proposto suporta a natureza dinâmica de aplicações em névoa. Outro resultado relevante diz respeito ao impacto do número de coordenadores – que poderiam ser os *gateways* na *fog* – no desempenho das aplicações: mais coordenadores não necessariamente garante melhor desempenho das aplicações, bem como o número ótimo de coordenadores não á afetado pelas características dinâmicas do sistema.

Premsankar et al. (2018) apresentam um estudo de caso para avaliação do impacto da computação na borda da rede (*i.e.*, *edge computing*) em aplicações de IoT. As plataformas de computação na borda da rede são classificadas em: servidores com muitos recursos implantados na borda da rede; nós de borda heterogêneos; e, federação na nuvem de servidores de borda. O estudo de caso avaliado envolve *mobile gaming*, por ser um cenário representativo de aplicações que envolvem sensores físicos além das entradas fornecidas pelos usuários. Os resultados apontam que, dada a natureza da aplicação (*i.e.*, baixa tolerância a atrasos e demanda computacional), a computação na borda acaba sendo, de fato, necessária para atender os requisitos de qualidade de experiência (*i.e.*, *Quality-of-Experience* (QoE)). Outra constatação relevante diz respeito ao fato que, mesmo acrescentando mais recursos computacionais na nuvem, não se compensa o atraso intrínseco à rede; ou seja, mesmo com reduzida capacidade computacional na borda da rede consegue-se atender os requisitos de QoE da aplicação em questão.

Puliafito et al. (2019) enfatizam o fato que a computação em névoa é uma tentativa de trazer os principais benefícios da computação em nuvem para próximo dos dispositivos e usuários. Nesse contexto, um conjunto de seis domínios de aplicações em IoT é posto em evidência: sistemas de transporte inteligente; assitência inteligente à saúde; segurança pública; rede elétrica inteligente; indústria 4.0; e, automação residencial e predial. Um conjunto de plataformas para IoT com suporte a computação em névoa é apresentado e analisado. Destaca-se o fato que, apesar de avanços significativos em termos de *software* e *hardware*, há uma carência de estudos de caso que evidenciem as potencialidades e limitações do paradigma da computação em névoa em aplicações de IoT.

8 Conclusão

Para a evolução do paradigma de IoT, deve-se tratar apropriadamente a escalabilidade em termos de dispositivos e demanda aos serviços ofertados, sobretudo em cenários com geração/tratamento de grande volume de dados em tempo real e com limites restritos de latência fim-a-fim. Como uma tentativa de sanar as limitações encontradas pelas aplicações quando hospedadas em cenários centralizados, como o de *cloud computing*, o paradigma de *fog computing* se apresenta como alternativa factível e promissora.

O presente trabalho possibilitou identificar vantagens e propriedades de um cenário de *fog computing* baseado na *Kaa IoT Platform*. Os resultados apresentam certa vantagem da estrutura descentralizada para aplicações que demandam menor tempo de resposta. Estruturas localizadas na nuvem estão sob total dependência do estado da rede externa (*i.e.*, características variáveis relativas ao caminho entre os dispositivos e os servidores na nuvem). Diminuir a frequência com

que as informações precisam sair da rede local dos dispositivos tornam as aplicações mais eficientes e menos dependentes das entidades externas localizadas na nuvem.

Como futuras oportunidades de pesquisa dentro desse tema, pode-se explorar diferentes configurações de *hardware* tanto na nuvem como na estrutura local da *fog*. A avaliação pode ser estendida a outras métricas relevantes como, por exemplo, confiabilidade, disponibilidade, privacidade e custo de implantação e manutenção das estruturas envolvidas.

Referências

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications, *IEEE Communications Surveys Tutorials* **17**(4): 2347–2376. https://doi.org/10.1109/COMST.2015.2444095.

Almes, G., Kalidindi, S. and Zekauskas, M. (1999). A Round-trip Delay Metric for IPPM, *RFC 2681*, The Internet Society . Disponível em https://tools.ietf.org/html/rfc2681.

Amazon Web Services, I. (2018). Tipos de instância do amazon ec2. Disponível em https://aws.amazon.com/pt/ec2/instance-types.

Association, I. S. (2016). Standard for an architectural framework for the internet of things (i ot). Disponível em http://grouper.ieee.org/groups/2413/Intro-to-IEEE-P2413.pdf.

Association, I. S. (n.d.). P2413 - standard for an architectural framework for the internet of t hings (iot). Disponível em https://standards.ieee.org/project/2413.html.

Auger, A., Exposito, E. and Lochin, E. (2017). Sensor observation streams within cloud-based iot platforms: Challenges and directions, *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, pp. 177–184. https://doi.org/10.1109/ICIN.2017.7899407.

Chiang, M. and Zhang, T. (2016). Fog and iot: An overview of research opportunities, *IEEE Internet of Things Journal* **3**(6): 854–864. https://doi.org/10.1109/JIOT.2016.2584538.

Cisco Systems, I. (2015). Fog computing and the internet of things: Extend the cloud to where the things are. Disponível em https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf.

Consortium, I. I. (n.d.). The industrial internet of things. volume g1: Reference architecture[online]. Disponível em https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pdf.

CyberVision, I. (2018). Kaa iot platform. Disponível em https://kaaproject.github.io/kaa/docs/v0.10.0/Welcome/.

da Silva Bonetti, F. and Spohn, M. A. (2018). Kaa application scripts. https://doi.org/10.5281/zenodo.1640766.

Giang, N. K., Lea, R. and Leung, V. C. (n.d.). Developing applications in large scale, dynamic fog computing: A case study, *Software: Practice and Experience* **0**(0). https://doi.org/10.1002/spe.2695.

Kang, B., Kim, D. and Choo, H. (2017). Internet of everything: A large-scale autonomic iot gateway, *IEEE Transactions on Multi-Scale Computing Systems* **3**(3): 206–214. https://doi.org/10.1109/TMSCS.2017.2705683.

La, Q. D., Ngo, M. V., Dinh, T. Q., Quek, T. Q. and Shin, H. (2019). Enabling intelligence in fog computing to achieve energy and latency reduction, *Digital Communications and Networks* **5**(1): 3 – 9. https://doi.org/10.1016/j.dcan.2018.10.008.

Laurent, J., Benoit, P., Dalmasso, L. and Gil, T. (2018). Computing in the fog with reconfigurable gateways, *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–4. https://doi.org/10.1109/ISCAS.2018.8351774.

Mell, P. and Grance, T. (2011). The nist definition of cloud computing, *Technical report*, National Institute of Standards and Technology – NIST. https://doi.org/10.6028/NIST.SP.800-145.

Premsankar, G., Di Francesco, M. and Taleb, T. (2018). Edge computing for the internet of things: A case study, *IEEE Internet of Things Journal* **5**(2): 1275–1284. https://doi.org/10.1109/JIOT.2018.2805263.

Project, F. E. (n.d.). Internet of things architecture (iot-a) [online]. Disponível em https://iotforum.org/wp-content/uploads/2014/09/D1.5-20130715-VERYFINAL.pdf.

Puliafito, C., Mingozzi, E., Longo, F., Puliafito, A. and Rana, O. (2019). Fog computing for the internet of things: A survey, *ACM Trans. Internet Technol.* **19**(2): 18:1–18:41. http://doi.acm.org/10.1145/3301443.

Raj, P. and Raman, A. C. (2017a). *The Internet of Things*, 1 edn, Auerbach Publications, New York.

Raj, P. and Raman, A. C. (2017b). *The Internet of things: enabling technologies, platforms, and use*, Taylor & Francis, CRC Press, Boca Raton, FL.

Roman, R., Zhou, J. and Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things, *Computer Networks* **57**(10): 2266 – 2279. https://doi.org/10.1016/j.comnet.2012.12.018.

Saharan, K. P. and Kumar, A. (2015). Fog in comparison to cloud: A survey, *International Journal of Computer Applications* **122**(3): 10–12. https://doi.org/10.5120/21679-4773.

Salami, A. and Yari, A. (2018). A framework for comparing quantitative and qualitative criteria of iot platforms, *2018 4th International Conference on Web Research (ICWR)*, pp. 34–39. https://doi.org/10.1109/ICWR.2018.8387234.

V. Paxson, G. A., Mahdavi, J. and Mathis, M. (1998). Framework for IP Performance Metrics, *RFC 2330*, The Internet Society . Disponível em https://tools.ietf.org/html/rfc2330.

Vaquero, L. M. and Rodero-Merino, L. (2014). Finding your way in the fog: Towards a comprehensive definition of fog computing, *SIGCOMM Comput. Commun. Rev.* **44**(5): 27–32. http://doi.acm.org/10.1145/2677046.2677052.

Wang, G. and Ng, T. S. E. (2010). The impact of virtualization on network performance of amazon ec2 data center, *2010 Proceedings IEEE INFOCOM*, pp. 1–9. https://doi.org/10.1109/INFCOM.2010.5461931.

Xen Networking (2011). Disponível em https://wiki.xenproject.org/wiki/Xen_Networking.

Zdravković, M., Trajanović, M., Sarraipa, J., Jardim-Gonçalves, R., Lezoche, M., Aubry, A. and Panetto, H. (2016). Survey of Internet-of-Things platforms, *6th International Conference on Information Society and Techology, ICIST 2016*, Vol. 1, Kopaonik, Serbia, pp. 216–220. Disponível em https://hal.archives-ouvertes.fr/hal-01298141/file/ICIST_2016_SoTA_IoTPlatforms_MZ.pdf.

Zhang, Q., Cheng, L. and Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges, *Journal of Internet Services and Applications* **1**(1): 7–18. https://doi.org/10.1007/s13174-010-0007-6.