



UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS DE CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

FERNANDA DA SILVA BONETTI

ANÁLISE DE DESEMPENHO DE UMA ARQUITETURA DE *FOG COMPUTING*
PARA *INTERNET OF THINGS* ATRAVÉS DE ESTUDOS DE CASO

CHAPECÓ

2018

FERNANDA DA SILVA BONETTI

**ANÁLISE DE DESEMPENHO DE UMA ARQUITETURA DE FOG COMPUTING
PARA *INTERNET OF THINGS* ATRAVÉS DE ESTUDOS DE CASO**

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Marco Aurélio Spohn

CHAPECÓ

2018

Bonetti, Fernanda da Silva

Análise de Desempenho de uma arquitetura de *Fog Computing* para *Internet of Things* através de estudos de caso / Fernanda da Silva Bonetti. – 2018.

52 f.: il.

Orientador: Prof. Dr. Marco Aurélio Spohn.

Trabalho de conclusão de curso (graduação) – Universidade Federal da Fronteira Sul, curso de Ciência da Computação, Chapecó, SC, 2018.

1. fog computing. 2. cloud computing. 3. internet of things. 4. desempenho. 5. análise. I. Spohn, Prof. Dr. Marco Aurélio, orientador. II. Universidade Federal da Fronteira Sul. III. Título.

© 2018

Todos os direitos autorais reservados a Fernanda da Silva Bonetti. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

E-mail: fernandasbonetti@gmail.com.br

FERNANDA DA SILVA BONETTI

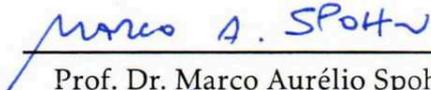
**ANÁLISE DE DESEMPENHO DE UMA ARQUITETURA DE FOG COMPUTING
PARA INTERNET OF THINGS ATRAVÉS DE ESTUDOS DE CASO**

Trabalho de conclusão de curso apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Marco Aurélio Spohn

Este trabalho de conclusão de curso foi defendido e aprovado pela banca avaliadora em: 2018-12-7.

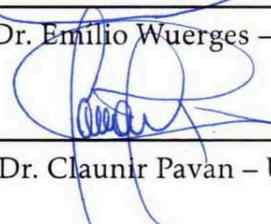
BANCA AVALIADORA



Prof. Dr. Marco Aurélio Spohn – UFFS



Prof. Dr. Emilio Wuerges – UFFS



Prof. Dr. Claunir Pavan – UFFS

AGRADECIMENTOS

Há um conjunto grande de pessoas que tiveram grande influência na finalização dessa jornada e merecem devido agradecimento. Primeiramente, agradeço a meus pais pelo privilégio que é possuir o incentivo a sempre buscar conhecimento. Agradeço também aos grandes professores que pude ter contato durante a graduação, especialmente ao Prof. Dr. Marco Aurélio Spohn pela orientação durante a pesquisa e no presente trabalho. E por fim, agradeço a meus colegas e grandes amigos da vida pelo apoio, pelas conversas edificantes e por tornar essa caminhada um pouco mais divertida.

*“Uma mente necessita de livros
da mesma forma que uma espada necessita de uma pedra de amolar,
se quisermos que se mantenha afiada.”*
(George R. R. Martin, A Guerra dos Tronos)

RESUMO

O presente trabalho possui como objetivo analisar em termos de desempenho a utilização de uma estrutura de *Fog Computing* para *Internet of Things* em contraposição ao cenário atual centralizado em *Clouds*. Para isso, são propostos estudos de caso com aplicações desenvolvidas com o suporte da plataforma de código aberto *Kaa IoT Platform*, avaliando cenários de *Fog Computing* e *Cloud Computing* através de métricas. Uma discussão é feita sobre o que os resultados colhidos demonstram sobre os cenários avaliados e o que a utilização desse novo paradigma representa para as aplicações de *Internet of Things*.

Palavras-chave: *fog computing. cloud computing. internet of things.*

ABSTRACT

The present work have as objective to analyze by means of performance the utilization of a Fog Computing structure for Internet of Things in contraposition to the current cloud-centered scenario. For this, case studies are proposed with applications developed with the support of the open-source platform Kaa IoT Platform, evaluating Fog Computing and Cloud Computing scenarios using metrics. A discussion is made about the collected results show about the evaluated scenarios and what the utilization of this new paradigm represents to Internet of Things applications.

Keywords: *fog computing. cloud computing. internet of things.*

LISTA DE FIGURAS

Figura 1 – <i>Cloud</i> como núcleo das informações.	26
Figura 2 – Arquitetura da Plataforma Kaa. Fonte: CyberVision (2018)	29
Figura 3 – Aproximação aos dispositivos por nodos de <i>Fog</i> . Fonte: Adaptação de (Saharan; Kumar, 2015)	32
Figura 4 – Cenários da Aplicação: 1 e 100 dispositivos para as estruturas.	38
Figura 5 – Cenário do processamento das amostras.	39
Figura 6 – Representação gráfica da latência de cada envio.	41
Figura 7 – Exemplo da estrutura de comunicação com virtualização <i>Xen</i> . Fonte: Project (2011)	42
Figura 8 – Latências de 100 dispositivos ativos (sem remoção de <i>outliers</i>).	43
Figura 9 – Latência 100 dispositivos para <i>Cloud</i> (Após remoção).	44
Figura 10 – Latência 100 dispositivos para <i>Fog</i> (Após remoção).	44
Figura 11 – Latência dos Testes com processamento.	46
Figura 12 – Latência dos testes com processamento e 100 dispositivos.	47

LISTA DE TABELAS

Tabela 1 – Comparação entre nuvem e fog em diferentes parâmetros. Fonte: Saharan; Kumar (2015).	33
Tabela 2 – Especificações da Instância. Fonte: Amazon Web Services (2018) . .	35
Tabela 3 – Especificações do hospedeiro da plataforma na rede local dos dispositivos.	35
Tabela 4 – Resultados do teste com um único dispositivo.	40
Tabela 5 – Resultados do teste de 100 dispositivos (sem remoção de <i>outliers</i>). .	43
Tabela 6 – Resultados do teste de 100 dispositivos (com remoção de <i>outliers</i>). .	44
Tabela 7 – Resultados do teste com envio e processamento de amostras.	45
Tabela 8 – Resultados do teste com envio e processamento de amostras com 100 dispositivos.	46

SUMÁRIO

1	INTRODUÇÃO	19
1.1	Objetivos	20
1.1.1	Objetivo Geral	20
1.1.2	Objetivos Específicos	20
1.2	Justificativa	20
2	INTERNET OF THINGS	23
3	CLOUD COMPUTING	25
4	PLATAFORMAS DE INTERNET OF THINGS	27
4.1	<i>Kaa IoT Platform</i>	28
5	FOG COMPUTING	31
6	ESTUDOS DE CASO	35
6.1	Cenários de avaliação	35
6.1.1	Cenário Um: Uma Estrutura Centralizada	35
6.1.2	Cenário Dois: Uma Estrutura Descentralizada	35
6.2	Avaliação de Desempenho	35
6.2.1	Métricas	35
6.2.1.1	Latência	36
6.2.1.2	Taxa de Transferência	36
6.2.2	Metodologia	36
6.2.2.1	Envio e armazenamento de Amostras na Plataforma	37
6.2.2.2	Envio e processamento de amostras	38
6.2.3	Resultados e Análise	40
6.2.3.1	Envio de amostras para a plataforma	40
6.2.3.1.1	<i>Teste com um único dispositivo</i>	40
6.2.3.1.2	<i>Teste com 100 dispositivos</i>	43
6.2.3.2	Envio e processamento de amostras	45
6.2.3.2.1	<i>Teste com único dispositivo</i>	45
6.2.3.2.2	<i>Teste com 100 dispositivos</i>	45
7	CONCLUSÃO	49
	REFERÊNCIAS	51

1 INTRODUÇÃO

Internet of Things (IoT), é um paradigma em ascensão cuja capacidade é conectar dispositivos à rede e garantir sua interoperabilidade. Quando se fala em dispositivos, é 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, bem como, dispositivos de usuários (*user devices*), sensores, atuadores e outros quaisquer dispositivos finais. (Kang; Kim; Choo, 2017)

O rápido crescimento desse paradigma é dado 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; Roderomero, 2014)

Porém, o estado atual das tecnologias de comunicação é, em sua maioria, uma arquitetura centralizada em *datacenters*, com o compartilhamento de recursos através de serviços baseados em nuvem (*Cloud-based Services*), que concentra todo o poder de processamento e capacidade de armazenamento das informações que chegam a essa estrutura. Essa arquitetura torna-se insustentável para o contexto de *Internet of Things*, pois conta com um grande número de dispositivos, com a alta carga de requisições geradas e enfrentando desafios como a latência de rede, largura de banda, confiabilidade e segurança. (Laurent et al., 2018)

Para driblar esses problemas faz-se necessária a adoção de uma estrutura descentralizada, não-monolítica e mais próxima aos dispositivos finais. Essa necessidade de uma nova estrutura é o que movimentou a pesquisa e a criação de um novo paradigma: a *Fog Computing*.

A *Fog Computing*, ou "Computação em Névoa", é uma extensão da estrutura de *Cloud* (Nuvem) que provê aos dispositivos que originam as informações, capacidade de processamento, armazenamento e comunicação, a fim de diminuir a frequência de troca de informações oriundas desses dispositivos. Uma forma de garantir isso, é implantar essa estrutura nos dispositivos de borda da rede, possibilitando a saída apenas de informações relevantes, realizando uma análise prévia desses dados. (Laurent et al., 2018)

A motivação de implantar esse tipo de estrutura na borda da rede é diminuir

os problemas encontrados com a arquitetura convencional de *Internet of Things*: toda a inteligência dos dispositivos está centrada em um único ponto, que requer acesso constante e acarreta alta latência.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Analisar em termos de performance o uso de uma estrutura de *Fog* utilizando uma plataforma *Open-Source* de *Internet of Things*.

1.1.2 Objetivos Específicos

- Definir uma estrutura de *fog* que melhor se adequa ao cenário proposto
- Realizar uma abordagem comparativa entre a atual arquitetura de IoT e com a utilização de *fogs*.
- Elencar as vantagens, limitações e a aplicabilidade da utilização de cada arquitetura, baseado em performance.

1.2 JUSTIFICATIVA

A aplicação do método científico é o que embasa o caráter investigativo-empírico da Ciência da Computação. A constante necessidade de avaliar e viabilizar métodos motiva a produção de conhecimento. O processo de experimentação e análise é fundamental para a garantia da inovação de tecnologias, métodos e paradigmas.

Explorar um contexto emergente como o proposto pela *Internet of Things* requer o estudo dos problemas presentes no cenário atual e instiga o desafio de procurar soluções que acompanhem o crescimento rápido e que favoreçam a sua perpetuação. Dentre as soluções já propostas nesse meio, a *Fog Computing* traz o conceito de reestruturar a arquitetura atual a fim de trazer melhorias de desempenho para as aplicações de *Internet of Things*.

O presente trabalho tem por finalidade analisar a aplicabilidade do uso de uma estrutura de *Fog Computing*, evidenciando sua performance em contraposição a cená-

rios centralizados em Nuvem. A realização desse estudo permite um melhor aprofundamento do conhecimento de métricas, arquiteturas e cenários dentro de uma área da Computação que se situa em pleno desenvolvimento.

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*), mas além disso sensores, atuadores e demais objetos do cotidiano que serão integrados e interconectados. Esses objetos ganharão características de processamento, comunicação em rede, percepção e atuação, análise e armazenamento. Graças a isso, computação estará presente o tempo todo e em todos os ambientes. (Raj; Raman, 2017)

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 *Internet of Things*.

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 de grande número de dispositivos, 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 de recursos computacionais, que precisam se conectar com essas estruturas. (Raj; Raman, 2017)

Outra característica importante a ser destacada sobre o contexto de *Internet of Things* é a 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. Segundo Raj; Raman (2017), a quantidade de informações geradas por dispositivos nesse contexto atinge *Terabytes* e *Petabytes*. Sendo assim, as aplicações têm de se adequar ao conceito de *Big Data*, processando e analisando dados em tempo real.

Por ser um paradigma ainda muito imaturo, há uma falta de padrões específicos, uma grande preocupação com a segurança das informações, habilidade de processar dados em tempo real e como equalizar o mercado de soluções de *Internet of Things*,

pois cada fim e aplicação 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; Raman, 2017)

De acordo com Raj; Raman (2017), há um conjunto de tecnologias necessárias para a solidificação da *Internet of Things*, como novos paradigmas como *Cloud*, *Fog*, *Edge computing*, modelos de comunicação, desenvolvimento de *Middlewares*, miniaturização de eletrônicos, virtualização e containerização para flexibilizar recursos de *Hardware*. A criação de plataformas como serviço (*Platform as a Service*) abrigadas em estruturas de *Cloud* facilita e otimiza a construção de soluções para esse fim. Dessa forma as estruturas de *Cloud* estão se tornando o centro das aplicações de *Internet of Things*.

3 CLOUD COMPUTING

A evolução do poder computacional e o barateamento de recursos culminaram no surgimento de um novo modelo de computação: a *Cloud Computing*. Nesse novo modelo, recursos como processamento e armazenamento podem ser alugados conforme sua demanda. Isso reduz os custos com infraestrutura (os recursos são cobrados conforme são usados), os provedores de serviço terceirizam os riscos e a manutenção de infraestrutura para os fornecedores de serviços de *Cloud*, além de, possibilitar a escalabilidade desses recursos de forma fácil. (Zhang et al, 2010)

Segundo Zhang et al. (2010), existem quatro tipos principais de estruturas de *Cloud*, podendo ser escolhidas conforme a necessidade do cenário escolhido:

- *Clouds* Públicas:
 - Estrutura em que os recursos são fornecidos para o público em geral, porém, seu controle sobre quesitos de dados, rede e características de segurança é reduzido.
- *Clouds* Privadas:
 - São estruturas desenvolvidas para uso único e restrito a uma única organização. Ao contrário das estruturas públicas, fornecem maior controle sobre dados, performance e são em geral mais confiáveis.
- *Clouds* 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.
- *Virtual Private Clouds (VPCs)*:
 - Basicamente uma plataforma de virtualização sobre *Clouds* públicas. Além da virtualização de servidores e aplicações, permite virtualizar comunicação interna na rede.

Com a utilização de *Clouds* ao invés da estruturação de *datacenters* privados, os recursos passam a ser vistos e utilizados como serviços. Dessa premissa, surgem

os conceitos de Infraestrutura como Serviço (*IaaS*) que provê a alocação de estruturas como máquinas virtuais, Plataforma como Serviço (*PaaS*) oferecendo *frameworks* e funcionalidades e *Software* como Serviço (*SaaS*) que provê aplicações específicas para serem utilizadas. Nesse contexto, tudo pode ser disponibilizado como serviço, tornando as estruturas de *Cloud* o ponto central de processamento, armazenamento e análise (Assim como o ilustrado na Figura 1). (Zhang et al., 2010)

Para o cenário de *Internet of Things*, a implantação dos serviços em *Clouds* facilita que o paradigma se torne cada vez mais presente. Segundo Raj; Raman (2017), levando em conta 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 sem conhecimento técnico.

Uma visão mais prática para esse cenário, seria manter a conectividade entre os dispositivos e a estrutura de *Cloud*, armazenando e processando dados de forma remota. Porém, a conectividade fora da rede local se torna um problema. E se os dispositivos estiverem impossibilitados de acessar ou armazenar informações na *Cloud* por causa de problemas de conexão? É nesse contexto que têm-se estudado formas de trazer as capacidades da *Cloud* para mais perto de onde os dispositivos estão inseridos.

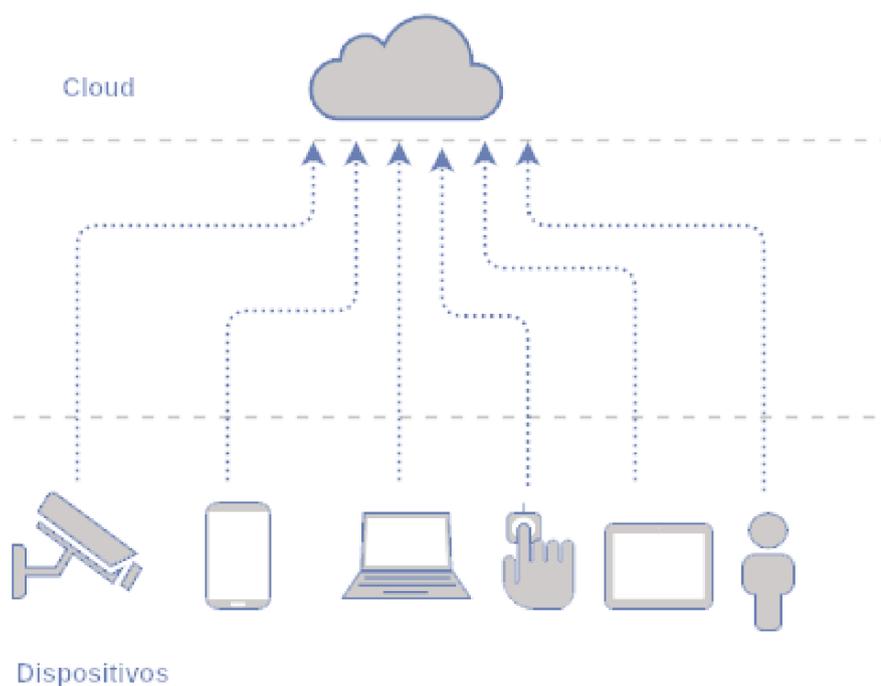


Figura 1 – *Cloud* como núcleo das informações.

4 PLATAFORMAS DE *INTERNET OF THINGS*

A popularização da computação em nuvem possibilitou o acesso e troca de informações entre aplicações de *Internet of Things*. Isso movimentou a criação de diferentes tipos de serviço, afim de suprir necessidades específicas desse contexto. Um dos modelos de serviço que ganhou grande enfoque em *Internet of Things* foi o *PaaS (Platform as a Service)*, fornecendo estrutura para facilitar o desenvolvimento e implantação de aplicações centralizadas em nuvem, sem a necessidade de customizar uma plataforma específica para cada fim. (Saharan; Kumar, 2015)

Plataformas voltadas à criação de soluções em *Internet of Things*, precisam garantir alta adaptação às necessidades de cada aplicação, disponibilidade e escalabilidade¹ dos recursos. A fim de lidar com um número massivo de fluxo de dados, devem ser capazes de efetuar escalabilidade horizontal (adição de mais instâncias) e escalabilidade vertical (aumentar os recursos alocados por instância). (Auger; Exposito; Lochin, 2017)

Essas plataformas também devem ter por natureza a capacidade de lidar e se adaptar a uma grande variedade de dispositivos. Essa adaptabilidade é possível graças a utilização de *Middlewares*, uma interface de comunicação e gerência de diferentes componentes e serviços que possibilita aos dispositivos o uso de um sistema comum. (Salami; Yari, 2018)

Alguns componentes necessários para que uma plataforma consiga conciliar dispositivos e serviços, segundo Salami; Yari (2018), são:

- 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 estão funcionando corretamente;
- c) Armazenamento: é um ponto crítico da plataforma, pois garante que os dados que foram recebidos pela camada de conectividade sejam armazenados para posterior processamento e visualização;

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

- d) Camada de processamento: permite ações baseadas em eventos para realizar processamentos específicos dos 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 *Internet of Things* é pela sua arquitetura, sendo elas centralizadas (os dados são coletados e 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; Zhou; Lopez, 2013)

4.1 KAA IOT PLATFORM

A plataforma Kaa, desenvolvida pela CyberVision Inc., é um *middleware* de código-livre, disponibilizado através da licença Apache 2.0, que permite o desenvolvimento de soluções fim-a-fim em *Internet of Things*. Algumas das funcionalidades que a plataforma possui em sua versão atual (0.10.0) são de coleta de dados, configuração, troca de mensagens e registro dos dispositivos. (CyberVision, 2018)

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

O Kaa permite que a implantação da plataforma seja realizada no modo *Single Node* (única instância) ou em forma de *cluster*. Utilizando pelo menos três instâncias interconectadas no modo *cluster*, é possível suportar um modo de alta-disponibilidade, onde se um nó falhar, os outros possam assumir. (CyberVision, 2018)

O desenvolvimento de aplicações para *Internet of Things* é facilitado pela versatilidade da biblioteca do Kaa, que suporta pelo menos 13 plataformas e dispositivos diferentes e quatro linguagens de programação.

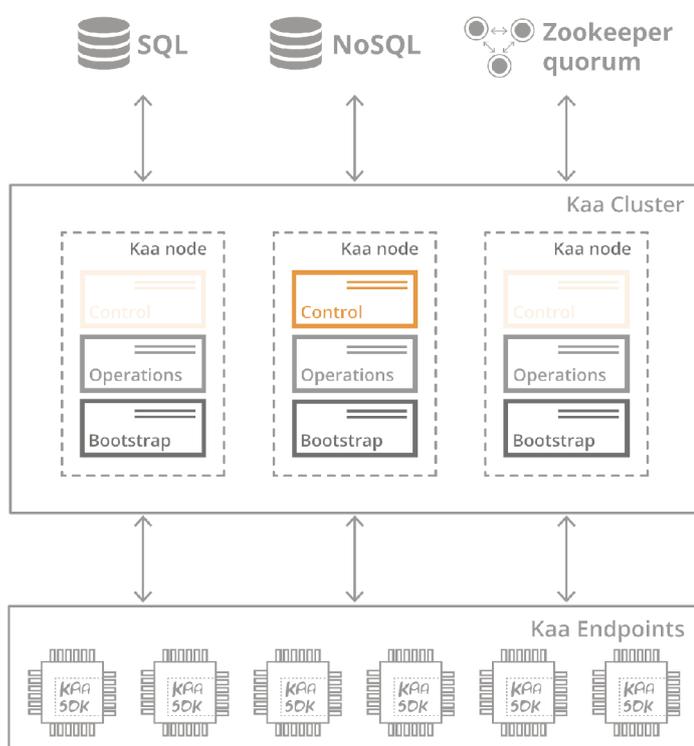


Figura 2 – Arquitetura da Plataforma Kaa. Fonte: CyberVision (2018)

5 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 armazenados e nas restrições de migração de serviços e aplicações entre diferentes provedores (prática chamada de *vendor lock-in*). (Saharan; Kumar, 2015)

Segundo Chiang; Zhang (2016), dos principais desafios encontrados quando se fala em *Internet of Things* em um cenário de nuvem, pode-se citar:

- a) Aplicações que requerem que a latência fim-a-fim seja ínfima;
- b) Aplicações que geram uma grande quantidade de dados em grande frequência, necessitando alta largura de banda;
- c) Dispositivos que possuem recursos computacionais extremamente limitados, cuja interação direta com a nuvem pode requerer algum processamento adicional, como operações criptográficas, por exemplo;
- 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 uma aplicação de coleta de dados a uma certa frequência de *upload*;

Visto os problemas encontrados na utilização tradicional de nuvens para *Internet of Things*, a Cisco Systems cunhou o termo *Fog Computing* para 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 desses dados, mais próxima de onde foram coletados diminui latência, tráfego de rede e permite maior controle sobre os dados que deverão sair da rede, gerando maior segurança. Qualquer dispositivo que permita processamento, armazenamento e comunicação em rede pode ser um nodo de *Fog*. (Cisco Systems, 2015)

A *Fog* permite rapidez de acesso aos dados para ambos os sentidos, tanto para a nuvem quanto aos dispositivos conectados. 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 a fonte dos dados, garantindo que seja possível suportar crescimento da quantidade de dispositivos e dados, evitar a saída de informações irrelevantes para fora da rede local (diminuindo assim a latência e economizando armazenamento) e descentralizar o processamento e análise desses dados. (Saharan; Kumar, 2015)

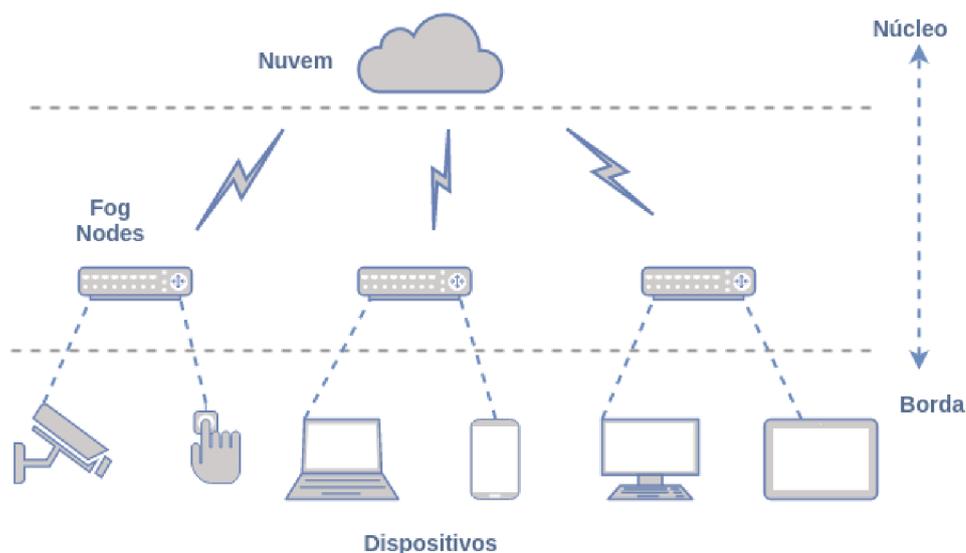


Figura 3 – Aproximação aos dispositivos por nodos de *Fog*. Fonte: Adaptação de (Saharan; Kumar, 2015)

Segundo Chiang; Zhang(2016), as estruturas de *Fog* e nuvem são complementares. A interdependência entre elas permite se beneficiar das vantagens presentes em cada uma. Algumas soluções podem ser gerenciadas pela nuvem e outras possuem melhor performance fazendo uso de *Fog*. Uma comparação entre características dessas estruturas é apresentada na Tabela 1.

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
<i>Delay Jitter</i>	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
Interações em tempo real	Suportadas	Suportadas
Tipo de Conexão <i>Last Mile</i>	Leased Line	Wireless
Mobilidade	Suporte Limitado	Suportada

Tabela 1 – Comparação entre nuvem e fog em diferentes parâmetros. Fonte: Saharan; Kumar (2015).

6 ESTUDOS DE CASO

6.1 CENÁRIOS DE AVALIAÇÃO

6.1.1 Cenário Um: Uma Estrutura Centralizada

O cenário Um descreve o ambiente elaborado para analisar uma estrutura centralizada em nuvem. Para isso, 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.

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 (2018)

6.1.2 Cenário Dois: Uma Estrutura Descentralizada

Com o intuito de reproduzir uma estrutura descentralizada de nuvem, o cenário Dois é composto de uma instalação *single-node* da plataforma Kaa na rede local dos dispositivos que originarão as requisições, a fim de observar a influência das especificações de *hardware* sobre a performance das aplicações de *Internet of Things*.

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.

6.2 AVALIAÇÃO DE DESEMPENHO

6.2.1 Métricas

Para analisar e quantificar propriedades das comunicações são estipuladas métricas de desempenho de rede. Essas métricas são definidas de acordo com metodologias específicas de coleta, padronização de unidades de medida e alguns critérios como citado na RFC 2330 (15), que estipula um *framework* para métricas de performance

em redes. Dentre as práticas citadas nesse 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 performance. Algumas dessas métricas de rede são parte integrante de muitos Acordos de Nível de Serviço (ou do inglês *Service Level Agreement - SLA*) feitas por fornecedores de serviços.

6.2.1.1 Latência

A latência total ou *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 performance principalmente para aplicações que precisam que o tempo de comunicação fim-a-fim seja ínfimo para funcionar de 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 do envio de um pacote pode ser diferente do caminho tomado para o envio da resposta e as medições obtidas em redes sob serviços que oferecem certo nível de qualidade de serviço (*Quality of Service - QoS*) é diferente das que não possuem essa garantia. (Almes et Al, 1999)

6.2.1.2 Taxa de Transferência

Outra métrica importante é a taxa de transferência, que representa a quantidade total de dados (normalmente em bits por segundo) transferidos em uma certa quantia de tempo. Taxas de transferência muito baixas são bons indicadores da existência de gargalos no caminho utilizado para comunicação em rede. Em cenários de *Internet of Things*, com uma alta carga de requisições, possuir uma taxa de transferência adequada é necessário para a fluidez das operações das aplicações.

6.2.2 Metodologia

A metodologia proposta é o desenvolvimento de testes que reproduzam o comportamento de aplicações reais de *Internet of Things*. Esses testes visam a observância

das métricas de performance de rede nos cenários definidos, para posterior análise.

Os testes compreendem uma aplicação de criação de amostras de temperatura e posterior envio para a plataforma. A aplicação gera uma amostra a cada segundo, porém, os envios são realizados a cada 5 segundos. A duração total de cada teste é de cinco minutos. O objetivo desse teste é medir a latência média dos envios das amostras para ambos os cenários. As especificações gerais dos testes seguem nas próximas seções e os recursos de código para replicação dos testes estão disponíveis nas referências desse trabalho.

6.2.2.1 Envio e armazenamento de Amostras na Plataforma

Em soluções de *Internet of Things*, bem como ambientes de sensores, monitoramento e coleta de dados, 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 para a sua viabilidade de implementação.

Para melhor visualização da importância da latência para algumas soluções, pode-se imaginar um conjunto de sensores responsáveis por aferir a temperatura de um ambiente de elementos extremamente sensíveis às condições do local (calor, umidade, luz, ...). Nesse ambiente hipotético, essas condições variam em uma taxa muito alta e há o interesse de armazenar essas informações com grande frequência, sem perda de dados, mantendo a acurácia de tempo da periodicidade dessas medições.

Para escolher um cenário ideal para a implementação dessa aplicação, deve-se realizar o armazenamento dos dados em uma estrutura fora da rede local dos sensores e em uma estrutura mais próxima à eles. A quantidade de dispositivos interagindo com essa estrutura também pode interferir na performance da aplicação, logo, testes devem ser realizados comparando a quantidade de dispositivos ativos.

Dado o cenário hipotético e utilizando a plataforma Kaa como base para realização desses testes, foi criada 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 5 (cinco) segundos. Para cada envio é calculada sua latência correspondente. A latência é calculada a partir de alterações no código da biblioteca Kaa SDK utilizada (mais especificamente do arquivo `kaa_logging.c` disponível nas referências desse trabalho), onde é registrado o tempo no

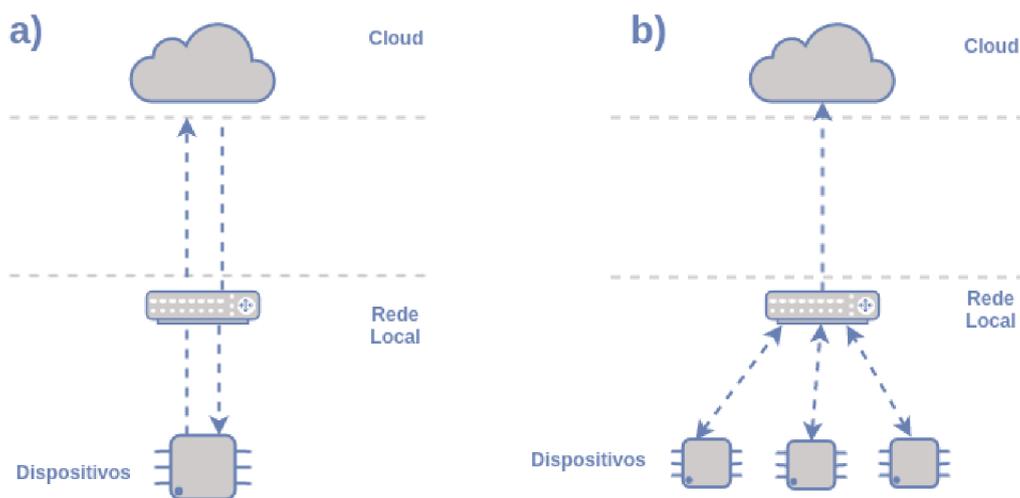


Figura 4 – Cenários da Aplicação: 1 e 100 dispositivos para as estruturas.

momento em que a amostra é enviada e o tempo de seu recebimento. A duração total do teste é de cinco minutos. O cenário dado, compreende duas modalidades de teste, demonstradas na Figura 4 e abaixo:

- 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*.

6.2.2.2 Envio e processamento de amostras

Reaproveitando o cenário descrito na seção anterior, é possível adicionar mais uma característica importante das soluções de *Internet of Things* a ele: o processamento dessas amostras.

Imaginando que deseja-se enviar uma quantia de amostras a plataforma, realizar cálculos sobre esse conjunto e retornar essas informações processadas aos dispositivos, a fim de que alguma ação seja realizada, por exemplo, dada uma média muito elevada de temperatura, é realizado o desligamento dos componentes evitando possíveis danos dado o superaquecimento.

Para a observância da latência para realizar essas operações, uma aplicação foi criada, utilizando as bibliotecas de desenvolvimento da plataforma Kaa. Essa aplicação

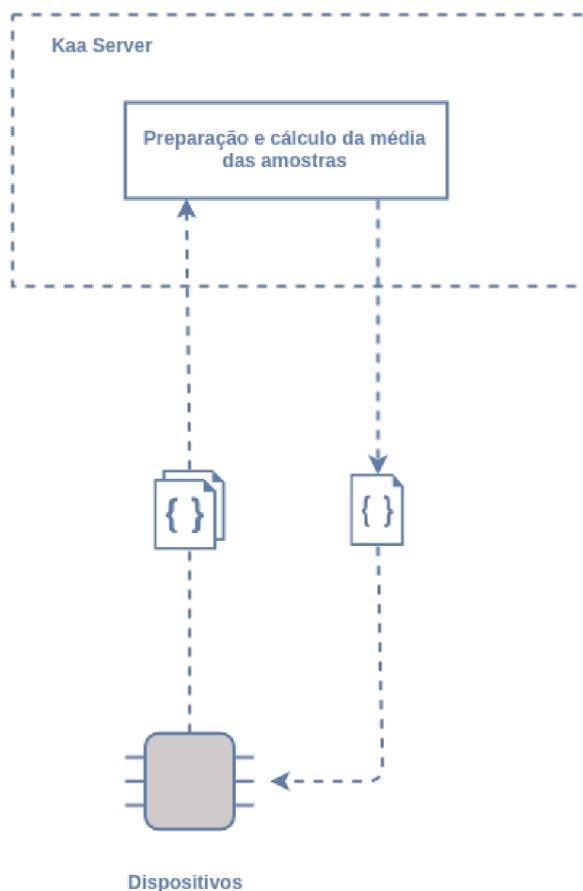


Figura 5 – Cenário do processamento das amostras.

envia um conjunto de 10 amostras de temperatura geradas de forma pseudo-aleatória para a plataforma. Na estrutura de servidor, esses dados são recebidos e armazenados. Para a realização do cálculo da média da temperatura das amostras é feita a leitura das amostras diretamente do banco de dados da plataforma (MongoDB), a filtragem dos dados importantes para o cálculo e a construção da resposta através de um *script* desenvolvido. Após isso, a média das amostras é enviada até o dispositivo que solicitou esse processamento. O tempo total de envio, processamento e retorno dessas informações ao dispositivo é o objetivo desse teste.

- 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*.

6.2.3 Resultados e Análise

Após a aplicação dos testes propostos e a reunião de seus respectivos resultados, é possível realizar uma comparação mais sólida entre o uso das estruturas propostas portando a plataforma Kaa. Nas seções seguintes, a discussão de cada resultado é feita separadamente.

6.2.3.1 Envio de amostras para a plataforma

6.2.3.1.1 Teste com um único dispositivo

Esse teste compreende as amostras enviadas tanto para a implantação da plataforma Kaa na estrutura de *Cloud* como para a de *Fog*. Foi realizado o cálculo da latência do envio de cada grupo de amostras, como especificado no capítulo anterior. Na tabela abaixo pode-se observar uma comparação que pôde ser realizada entre as duas estruturas graças ao teste aplicado.

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.

Para melhor compreensão dos resultados mostrados na Tabela 4 e no gráfico da Figura 6, é preciso explicitar as condições que podem ter afetado os dados colhidos. Como o ambiente de *Cloud* usado localiza-se a uma quantia considerável de saltos de distância a mais em comparação com a estrutura de *Fog*, é previsto que a latência entre os dispositivos e a estrutura de *Cloud* seria superior. O desvio padrão das amostras enviadas à estrutura de *Cloud* é 4,24 vezes maior que o 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 observações, podendo ser um conjunto de fatores do meio, assim como da própria estrutura de *Cloud*.

Como descrito anteriormente na seção 6.1, a estrutura utilizada para abrigar a plataforma é uma instância t2.micro da *Amazon Web Services*, localizada na região

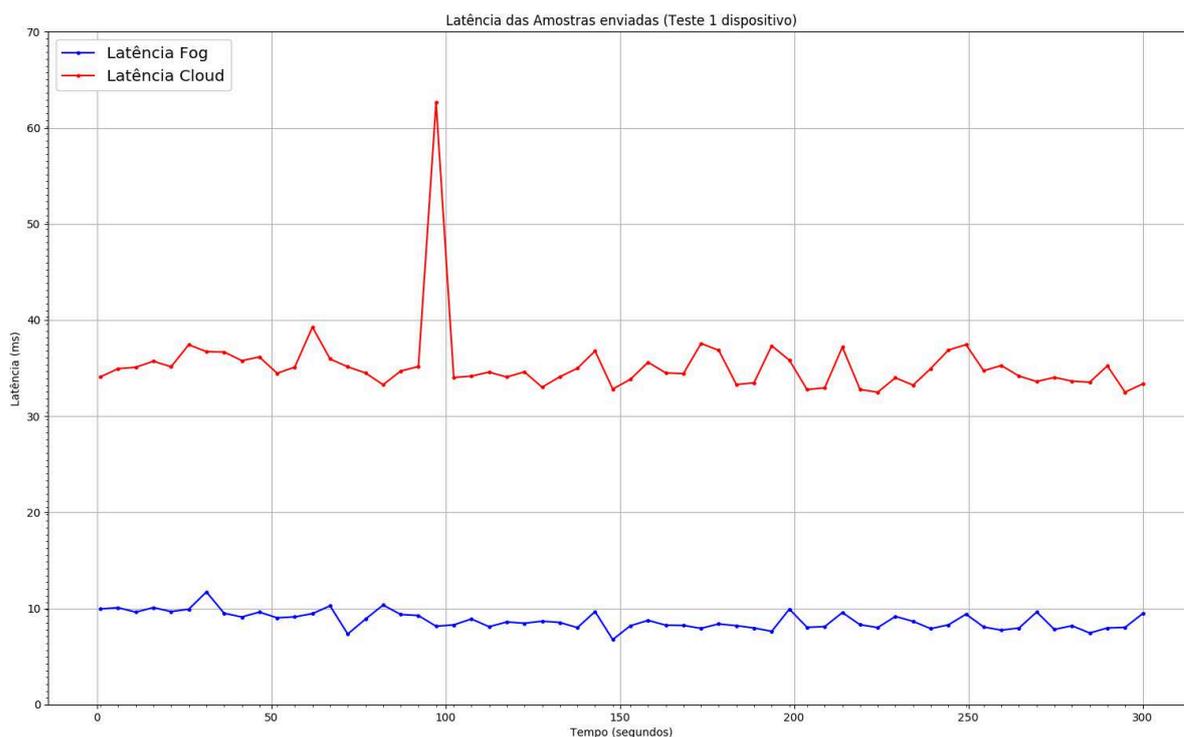


Figura 6 – Representação gráfica da latência de cada envio.

leste da América do Sul (*South America East - São Paulo*), há cerca de 25 saltos da rede que abriga os dispositivos que originam as amostras. Essa instância é parte do nível gratuito e é bem mais limitada em comparação às outras instâncias disponíveis.

O impacto de instâncias Amazon EC2 mais limitadas em testes de performance é abordado por Wang; Ng (2010). Segundo o autor, as instâncias Amazon EC2 utilizam virtualização *Xen* para a gerência dos servidores, graças a isso, possibilitando a existência de várias máquinas virtuais alocadas em um único servidor, dividindo recursos computacionais.

Um ponto importante a ser destacado para entendimento posterior é que essas instâncias são escalonadas para uso dos processadores físicos e, especificamente para o uso das interfaces de rede, essas máquinas virtualizadas (chamadas *Guest domains*, do inglês "domínios convidados") precisam se comunicar através de uma outra máquina virtual com maior privilégio (chamada *Driver Domain*) para acessar dispositivos físicos. Em operações de envio de pacotes, por exemplo, esses são enviados das *Guest Domains* para a *Driver Domain* através de suas interfaces virtuais e posteriormente (através dela) para a rede em si. (Wang; Ng, 2010)

Segundo Wang; Ng (2010), instâncias menores (*small* e *micro*) estão sempre dividindo processadores com outras instâncias e através de políticas de controle compu-

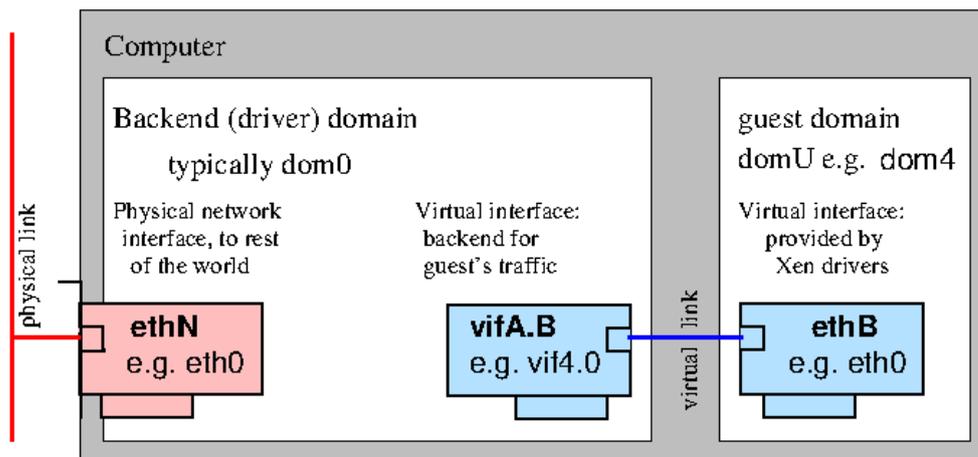


Figura 7 – Exemplo da estrutura de comunicação com virtualização Xen. Fonte: Project (2011)

tacional, as instâncias menores conseguem atingir apenas 50% do processador físico (mesmo com mais nenhuma instância ativa no mesmo servidor).

Graças a essas condições de competição por recursos, instâncias menores podem ter sua performance de rede afetada. Como resultado do estudo com instâncias Amazon, Wang; Ng(2010) conclui os problemas e possíveis fatores que os acarretam:

- 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 provável que a divisão de recursos o escalonamento de instâncias influencie no comportamento da rede, visto que, durante o envio de pacotes a instância que está enviando pode perder o processador.
- Alta oscilação do atraso fim-a-fim:
 - Causadas pela formação de filas de pacotes (*buffering*) no *Driver Domain* das instâncias.

Dadas as condições da estrutura de *Cloud* que abrigou os testes, pode-se ter uma visão mais clara dos possíveis fatores que levaram o teste a ter picos de latência que não foram observados na estrutura de *Fog*.

6.2.3.1.2 Teste com 100 dispositivos

Utilizando a mesma premissa do teste anterior, esse por sua vez possui a utilização de 100 dispositivos enviando as amostras para as estruturas abrigo a plataforma Kaa.

Assim como no teste anterior, houveram oscilações que influenciaram a medida de latência em tempos específicos durante o teste realizado. Para isso, abaixo seguem os resultados extraídos do teste e também após a remoção desses *outliers* das amostras.

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*).

Com auxílio do gráfico da Figura 8 é possível observar as amostras que possuem um afastamento drástico do resto da série:

Histograma de Latências

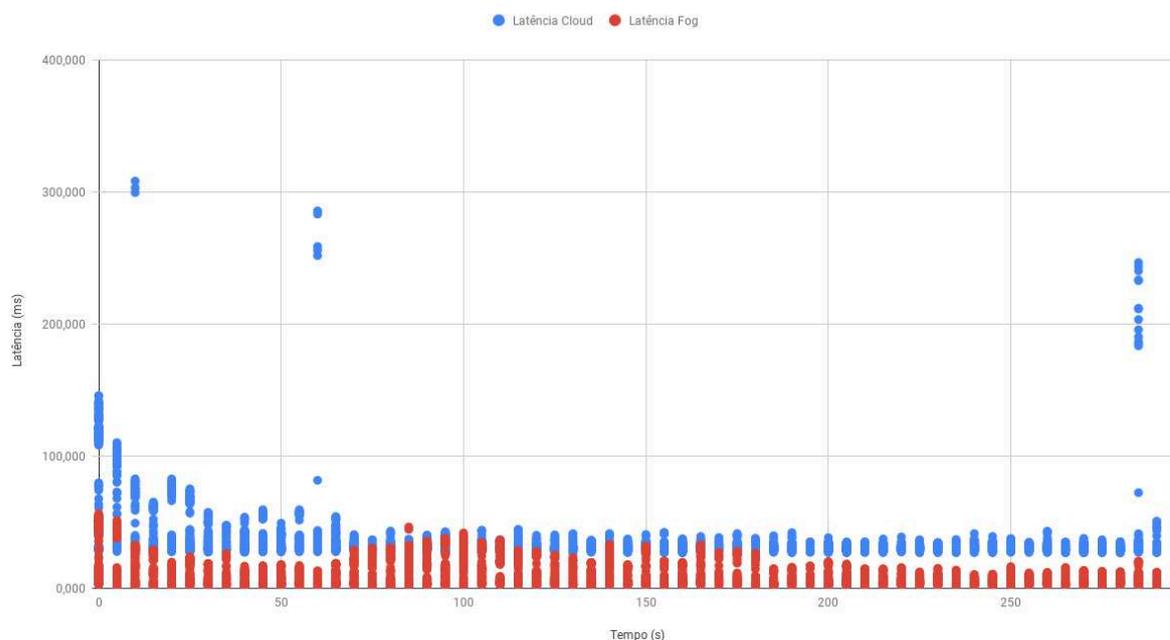


Figura 8 – Latências de 100 dispositivos ativos (sem remoção de *outliers*).

Com a remoção dos *outliers* do conjunto de amostras (de ambos os conjuntos), uma visão mais justa dos resultados é obtida, segundo a tabela 6:

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

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

Histograma de Latência Cloud

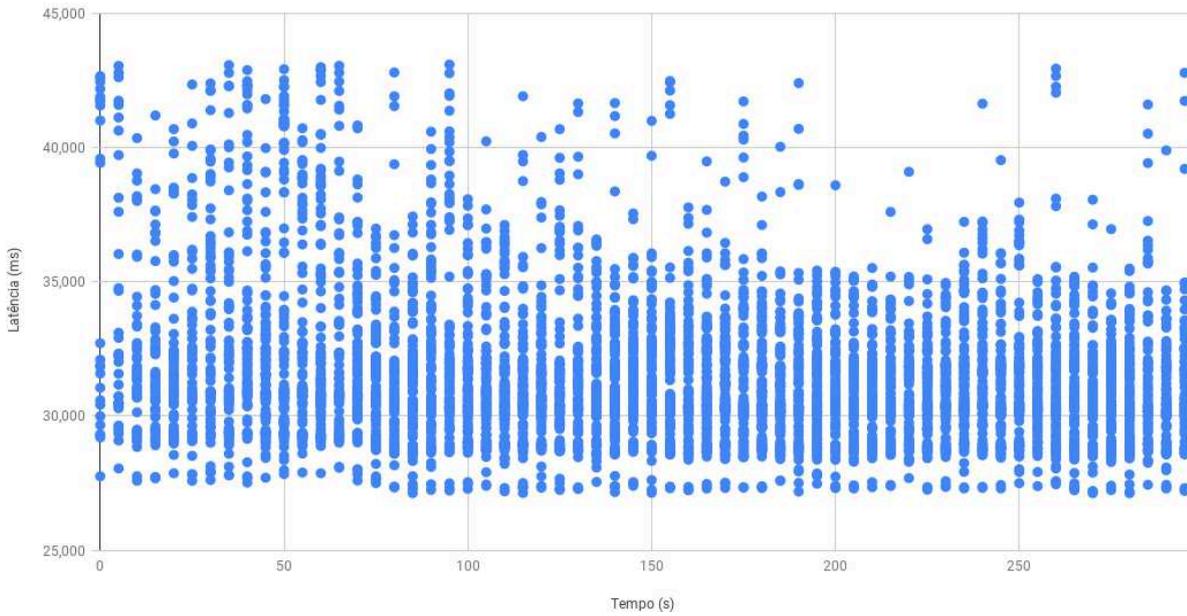


Figura 9 – Latência 100 dispositivos para *Cloud* (Após remoção).

Histograma de Latências Fog

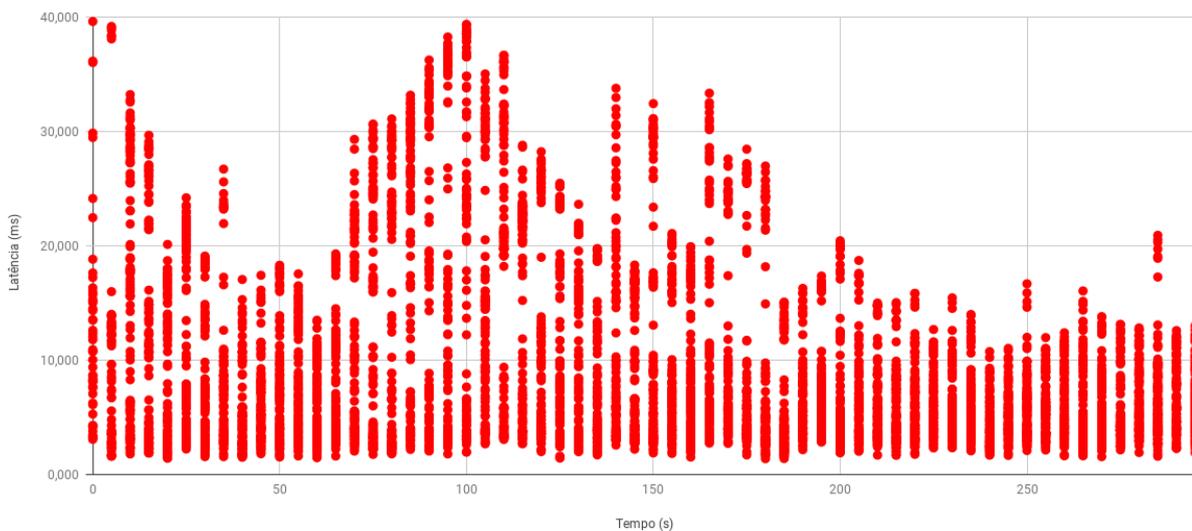


Figura 10 – Latência 100 dispositivos para *Fog* (Após remoção).

Os resultados com a remoção dos *outliers* permitem uma visão mais uniforme do comportamento das estruturas de *Fog* e *Cloud*, porém, deve-se levar em conta que com a utilização de uma estrutura fora da rede local, o meio entre os dispositivos e ela

está fora de controle das aplicações e que essas anomalias podem ocorrer, acarretando em latências maiores.

6.2.3.2 Envio e processamento de amostras

6.2.3.2.1 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 de amostras, os resultados de ambos os cenários fica menos discrepante.

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	11478,935	570,571
Fog	11367,141	441,213

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

Essa característica é notável quando é observado o gráfico dos testes, como mostra a Figura 11. Em alguns testes, a latência da estrutura de *Cloud* foi inferior a observada em *Fog*, mesmo tendo um *overhead* de rede muito maior do que a instância localizada na mesma rede local. A latência de processamento da estrutura de *Cloud* em alguns instantes supera o seu próprio *overhead* de rede, rendendo em uma latência total menor (em alguns instantes) do que a estrutura de *Fog*. Como já comentado anteriormente, por essa instância ser uma estrutura virtualizada, está em constante disputa pelos recursos da máquina que a abriga, há momentos em que a carga de instâncias pode ser menor, resultando em fatias de tempo de utilização desses recursos de forma mais frequente. Como detalhado na seção 6.1, mesmo tendo que concorrer aos recursos de processamento, a instância virtualizada é munida de características de *hardware* mais poderosas (por se tratar de uma estrutura para servidor) que a estrutura testada como *Fog*.

6.2.3.2.2 Teste com 100 dispositivos

Os resultados dos testes de latência (com processamento de amostras), pode ser observado na Tabela 8. Comparado com o teste anterior, que envolvia somente um

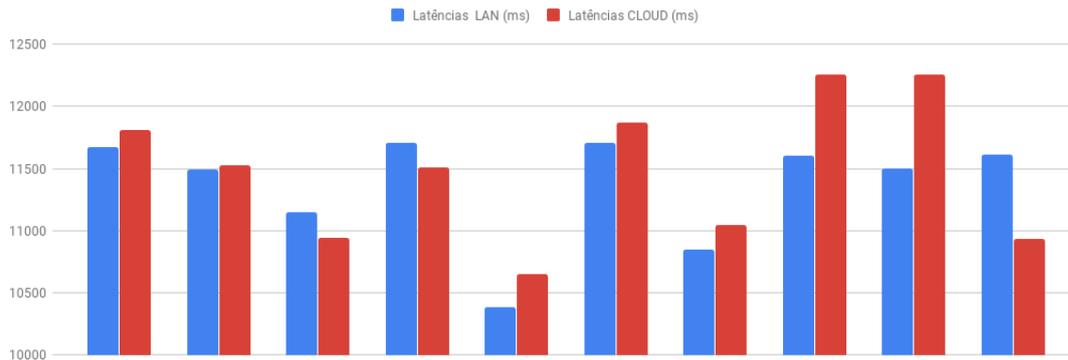


Figura 11 – Latência dos Testes com processamento.

dispositivo originando os dados, nesse o desvio padrão das medições para a estrutura de *Cloud* dobra.

Estrutura	Média das latências (ms)	Desvio Padrão
Cloud	14766,909	1232,360
Fog	12660,712	529,560

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

Testes com maior número de dispositivos (e consequentemente mais requisições para a estrutura) demonstram uma desvantagem da estrutura de *Cloud* proposta. Esse crescimento da variação pode ser associado às condições da rede no caminho entre os dispositivos e a estrutura e às próprias condições da instância utilizada para abrigar a plataforma. Para cenários de *Internet of Things*, onde o número de dispositivos é grande, a utilização de um ambiente centralizado sob influência de características de virtualização de instâncias, deve ser pensada levando em conta essas variações de latência. Aplicações dependentes de um tempo de resposta pequeno podem ter seu desempenho prejudicado com a adoção desse tipo de estrutura.

Nos estudos de caso presentes nesse trabalho, os testes realizados continham até 100 dispositivos operando para maior controle dos testes e coleção dos resultados, porém, em ambientes reais de aplicações de *Internet of Things* o número de dispositivos é muito maior. Pensando em aplicações de cenários de ambientes de sensores e atuadores, cidades e casas inteligentes, esse número é facilmente superado e possui contínuo crescimento pela fácil integração de novos dispositivos às aplicações.

As plataformas e aplicações para esse fim devem possuir suporte e lidar de forma fluída com essa carga de dispositivos e requisições, porém, ao serem projeta-

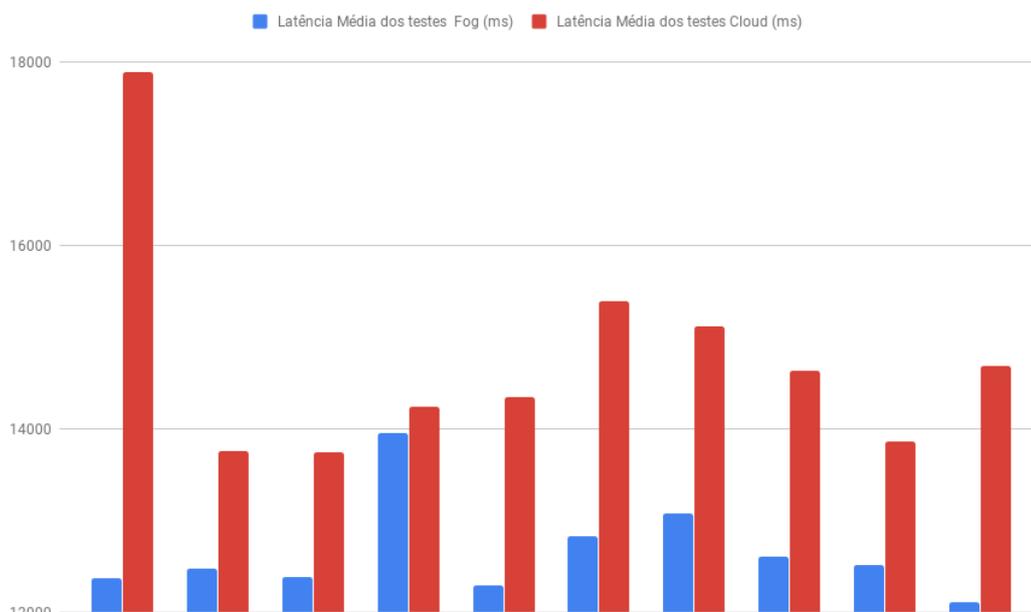


Figura 12 – Latência dos testes com processamento e 100 dispositivos.

das e implementadas, deve-se levar em conta todos os elementos que possam resultar em uma performance abaixo do esperado. Quando hospedadas em ambientes centralizados como os de *Cloud Computing*, o número de elementos que podem acarretar nessas interferências é maior. O meio de comunicação entre a rede que abriga os dispositivos e a estrutura pode ser um deles, sofrendo congestionamentos, possuindo maior número de saltos e sendo uma constante fonte de incertezas por não se ter um controle de qualidade sobre ele.

Outra característica a ser levada em conta em estruturas desse tipo são os recursos que elas oferecem e sob quais condições são oferecidos. Nesse trabalho, foi optado a utilização de uma estrutura mais reduzida em recursos para abrigar a plataforma escolhida, essa decisão pode ter influenciado nos resultados colhidos, dado que, por se tratar de uma instância gratuita fornecida pela Amazon, o nível de qualidade de serviço garantido é inferior a instâncias mais robustas e que são customizadas de acordo com as necessidades das soluções que serão abrigadas.

Com a utilização de uma estrutura localizada na rede local dos dispositivos, essas incertezas relacionadas ao caminho da comunicação são menores, pelo conhecimento das estruturas utilizadas e pela possibilidade de controlar como essa comunicação é realizada. Essa descentralização das capacidades da *Cloud*, traz a diminuição do atraso de comunicação, mas também uma maior segurança para as informações que são enviadas dos dispositivos para as plataformas. A escolha por descentralizar a co-

municação depende dos requisitos das soluções que serão desenvolvidas, aplicações que não dependem de forma crucial do atraso de comunicação podem se beneficiar muito bem da estrutura em *Cloud* sem a necessidade de nodos de *Fog*.

A presença de nodos de *Fog* abre um leque de cenários aplicáveis para *Internet of Things*. Há a possibilidade de escolha do local de acesso e armazenamento das informações baseado na localização dos dispositivos, pode-se diminuir a latência fim-a-fim armazenando e processando as informações na *Fog* e replicando-as periodicamente para a estrutura de *Cloud*, para que essas informações possam ser acessadas também fora da rede. Graças ao pré-processamento na rede local, a capacidade de estruturar dados antes de serem armazenados na *Cloud*, filtrá-los enviando somente dados que realmente são necessários, realiza uma economia de recursos de armazenamento, diminui a frequência de envio de requisições para fora da rede e acarretando em taxas de transmissão maiores.

7 CONCLUSÃO

Para a evolução e aplicabilidade do paradigma de *Internet of Things*, as tecnologias atuais devem ser adaptadas, para que os requisitos das aplicações sejam viáveis. Dentre esses requisitos, lidar com uma carga grande de dispositivos e requisições, grande frequência de geração de informações e a necessidade de uma menor latência de comunicação 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* é pensado.

O presente trabalho possibilitou visualizar as vantagens e aplicabilidade do uso de um cenário de *Fog Computing* utilizando a plataforma *Kaa IoT Platform* como suporte para o desenvolvimento dos estudos de caso. Os dados colhidos mostraram certa vantagem da estrutura descentralizada quando se há a necessidade de aplicações que possuam um tempo de resposta menor. Estruturas localizadas fora da rede local estão sob total interferência do estado do caminho traçado para se comunicarem com as plataformas. Diminuir a frequência com que as informações precisam sair da rede dos dispositivos permite às aplicações o alcance de um melhor desempenho, sem inutilizar as estruturas de *Cloud*.

Como futuras oportunidades de pesquisa dentro desse tema, esse estudo pode ser replicado utilizando instâncias mais robustas (tanto para a estrutura de *Cloud* como para *Fog*), a fim de equalizar as observações. A exploração de mais métricas e diferentes outros tipos de aplicações presentes nos cenários de *Internet of Things* podem ser contribuições relevantes.

REFERÊNCIAS

- 1 ALMES, G.; KALIDINDI, S.; ZEKAUSKAS, M. **A Round-trip Delay Metric for IPPM**. [S.l.], set. 1999. p. 1–19. Disponível em: <<https://tools.ietf.org/html/rfc2681>>.
- 2 AMAZON WEB SERVICES, Inc. **Tipos de instância do Amazon EC2**. 2018. Disponível em: <<https://aws.amazon.com/pt/ec2/instance-types>>. Acesso em: 23 ago. 2018.
- 3 AUGER, A.; EXPOSITO, E.; LOCHIN, E. Sensor observation streams within cloud-based IoT platforms: Challenges and directions. In: 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN). [S.l.: s.n.], mar. 2017. p. 177–184. DOI: 10.1109/ICIN.2017.7899407.
- 4 CHIANG, M.; ZHANG, T. Fog and IoT: An Overview of Research Opportunities. **IEEE Internet of Things Journal**, v. 3, n. 6, p. 854–864, dez. 2016. ISSN 2327-4662. DOI: 10.1109/JIOT.2016.2584538.
- 5 CISCO SYSTEMS, Inc. **Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are**. [S.l.: s.n.], 2015. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. Acesso em 4 Abr 2018.
- 6 CYBERVISION, Inc. **Kaa IoT Platform**. [S.l.: s.n.], 2018. <https://kaaproject.github.io/kaa/docs/v0.10.0/Welcome/>. Acesso em 03 Mar 2018.
- 7 KANG, B.; KIM, D.; CHOO, H. Internet of Everything: A Large-Scale Autonomic IoT Gateway. **IEEE Transactions on Multi-Scale Computing Systems**, v. 3, n. 3, p. 206–214, jul. 2017. DOI: 10.1109/TMSCS.2017.2705683.
- 8 LAURENT, J. et al. Computing in the Fog with Reconfigurable Gateways. In: 2018 IEEE International Symposium on Circuits and Systems (ISCAS). [S.l.: s.n.], maio 2018. p. 1–4. DOI: 10.1109/ISCAS.2018.8351774.
- 9 PROJECT, Xen. **Xen Networking**. 2011. Disponível em: <https://wiki.xenproject.org/wiki/Xen_Networking>. Acesso em: 13 nov. 2018.
- 10 RAJ, Pethuru; RAMAN, Anupama C. **The Internet of Things**. 1. ed. New York: Auerbach Publications, fev. 2017. ISBN 9781498761291.

- 11 ROMAN, Rodrigo; ZHOU, Jianying; LOPEZ, Javier. On the features and challenges of security and privacy in distributed internet of things. **Computer Networks**, v. 57, n. 10, p. 2266–2279, 2013. Towards a Science of Cyber Security Security and Identity Architecture for the Future Internet. ISSN 1389-1286. DOI: <https://doi.org/10.1016/j.comnet.2012.12.018>. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128613000054>>.
- 12 SAHARAN, K. P.; KUMAR, Anuj. Article: Fog in Comparison to Cloud: A Survey. **International Journal of Computer Applications**, v. 122, n. 3, p. 10–12, jul. 2015. Full text available.
- 13 SALAMI, A.; YARI, A. A framework for comparing quantitative and qualitative criteria of IoT platforms. In: 2018 4th International Conference on Web Research (ICWR). [S.l.: s.n.], abr. 2018. p. 34–39. DOI: 10.1109/ICWR.2018.8387234.
- 14 SILVA BONETTI, Fernanda da; SPOHN, Marco Aurélio. **Kaa Application Scripts**. 2018. Disponível em: <https://doi.org/10.5281/zenodo.1640766>>. Acesso em: 28 nov. 2018.
- 15 V. PAXSON, G. Almes; MAHDAVI, J.; MATHIS, M. **Framework for IP Performance Metrics**. [S.l.], maio 1998. p. 1–40. Disponível em: <https://tools.ietf.org/html/rfc2330>>.
- 16 VAQUERO, Luis M.; RODERO-MERINO, Luis. Finding Your Way in the Fog: Towards a Comprehensive Definition of Fog Computing. **SIGCOMM Comput. Commun. Rev.**, ACM, New York, NY, USA, v. 44, n. 5, p. 27–32, out. 2014. ISSN 0146-4833. DOI: 10.1145/2677046.2677052. Disponível em: <http://doi.acm.org/10.1145/2677046.2677052>>.
- 17 WANG, G.; NG, T. S. E. The Impact of Virtualization on Network Performance of Amazon EC2 Data Center. In: 2010 Proceedings IEEE INFOCOM. [S.l.: s.n.], mar. 2010. p. 1–9. DOI: 10.1109/INFOCOM.2010.5461931.
- 18 ZHANG, Qi; CHENG, Lu; BOUTABA, Raouf. Cloud computing: state-of-the-art and research challenges. **Journal of Internet Services and Applications**, v. 1, n. 1, p. 7–18, maio 2010. ISSN 1869-0238. DOI: 10.1007/s13174-010-0007-6. Disponível em: <https://doi.org/10.1007/s13174-010-0007-6>>.