

Um serviço escalável de virtualização de recursos para aplicações de cidades inteligentes

Vinício Vieira dos Santos¹, Fábio Moreira Costa²

¹Estudante ²Orientador

Instituto de Informática – Universidade Federal de Goiás – Goiânia – GO.

vvs50@discente.ufg.br¹, fmc@inf.ufg.br²

Resumo. Nos trabalhos anteriores vimos a implementação de protocolos para a comunicação e virtualização de sensores de baixo nível, assim como a criação de um protocolo de comunicação não centralizado entre tais sensores e a plataforma de cidades inteligentes InterSCity. Nesse artigo, focando na alocação eficiente e escalável de recursos virtuais na névoa, e prover uma pesquisa eficiente de recursos para serem virtualizados. O trabalho envolve o estudo de modelos de desempenho de sistemas computacionais distribuídos e alocação de recursos na névoa, a extensão da arquitetura do Virtualizador para gerenciamento dinâmico dos recursos e a avaliação de desempenho em cenários prováveis. Os resultados incluem um sistema de pesquisa com arquitetura hierárquica de recursos disponíveis, e a criação de recursos virtualizados a partir desse recursos, implementação do protótipo, e avaliação do desempenho.

1. Apresentação

1.1. Introdução

Com o grande crescimento no número de dispositivos conectados a rede no cotidiano das pessoas, juntamente com o grande número de sensores. A Internet das Coisas (IoT) também torna-se cada vez mais importante no cotidiano das pessoas, juntamente com o gradativo aumento da troca de dados entre esses dispositivos e com o enorme número de informações sendo gerados e precisando ser processados e armazenados. Nesse contexto, as tecnologias de computação em nuvem (*cloud computing*) surgiram como uma solução barata, eficiente, e flexível para o tratamento, processamento e armazenamento dos dados.

Entretanto, apesar dessas vantagens da *Cloud* ainda existem limitações, como aplicações que exigem baixa latência, respostas rápidas e processamento próximo à fonte dos dados. O atraso da transmissão e possíveis instabilidades de rede podem comprometer a eficiência e a confiabilidade de certas aplicações críticas.

Tendo isso em vista, veio o surgimento de computação em névoa (*fog computing*), propondo a utilização de dispositivos mais próximos do usuário ou da fonte dos dados, possibilitando o processamento local de informações antes de encaminhá-las à nuvem.

Nesse trabalho atual, com a utilização dos conceitos mencionados como a *InterScity* [de M. Del Esposte et al. 2019] na *Cloud*, a virtualização de recursos

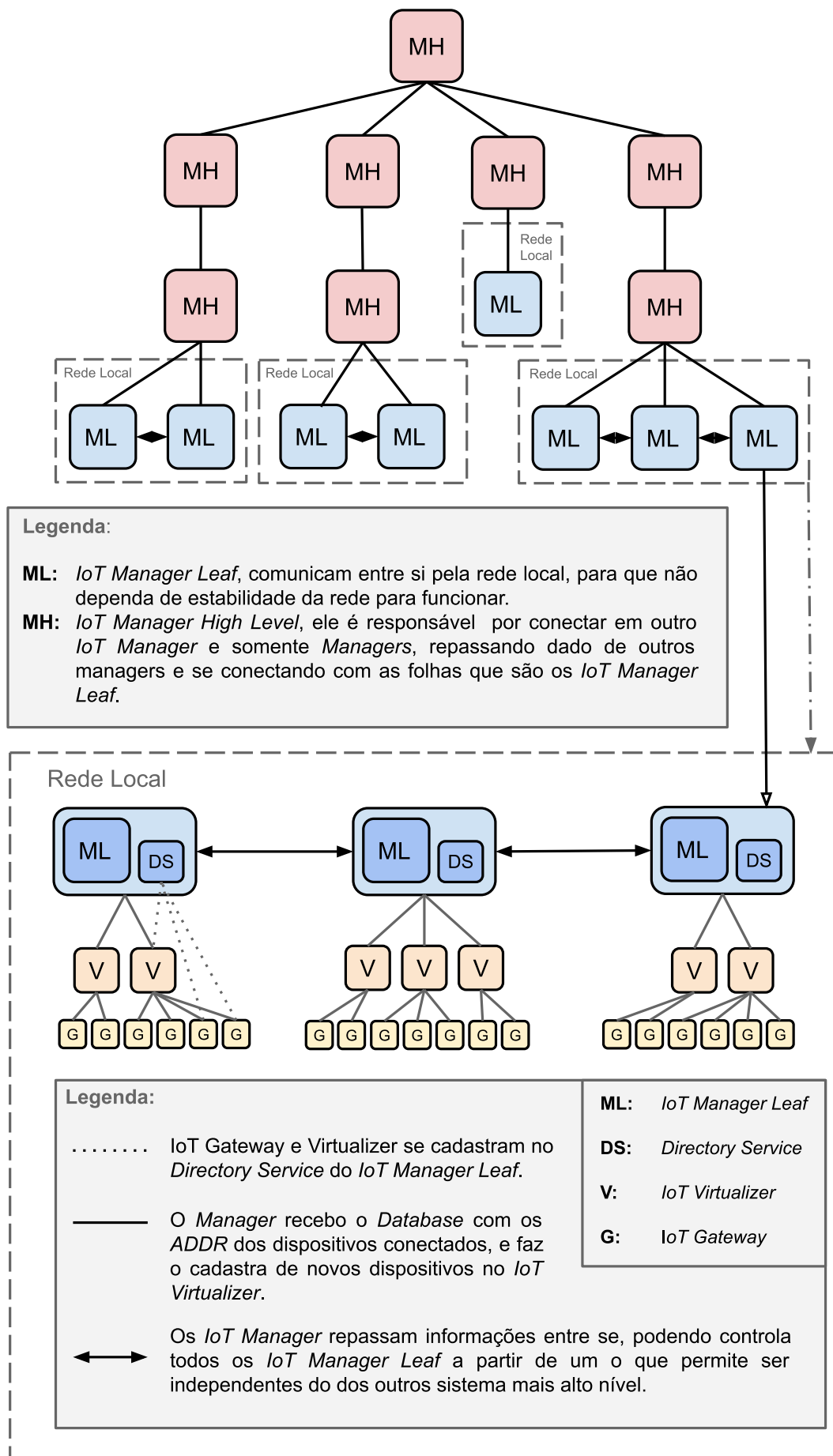


Figura 1. Protocolos dos IoT Managers

na *fog*, criamos um arquitetura para alocação de recursos e sensores eficiente, que possibilita sua orgânica, de maneira disponibilizar um gerenciamento dinâmico para criação de virtualizadores.

1.2. Referências conceituais

Como citado anteriormente, na *cloud*, temos a plataforma de cidade inteligente *InterSCity*, oferecem suporte para o desenvolvimento e integração de aplicativos de cidades inteligentes, sendo plataforma de código aberto baseada em microsserviços. A plataforma recebe os dados de sensores, e atuadores por meio do *IoTGateway* [R. Silva 2020] e *IoTVirtualizer* [R. Silva 2021], dados são rebebidos pelo *Resource Adaptor*, o microsserviço que cadastra, e é responsável pela consulta e envio de comandos aos atuadores.

Contudo, em um sistema real de cidade inteligente, seria muito complexo a pesquisa de recursos, então foi criado um sistema que gerencia esse *IoTGateway*, mas também que permite gerenciar *IoTVirtualizer*, permitindo também criação de novos recursos virtuais a a partir desse mesmo serviço.

Além do mais, outro desafio encontrado foi a complexidade de sensores em uma cidade inteligente, que pode ser divididos por exemplo em bairros, setores, prédios, e salas. Com isso, desensolvemos um “Gerente de gerente” e dissidimos nomear o gerente responsável diretamente com o *IoTGateway* e *IoTVirtualizer* de *IoTManagerLeaf* e os responsável por organizar esse *IoTManagerLeaf* de *IoTManagerHigh* e um exemplo desse protocolo pode ser visto na figura 1.

Conjuntamente na *fog*, como mencionado temos o *IoTGateway* e *IoTVirtualizer*. O *IoTGateway* é o componente que possibilita o envio de dados de um sensor, que podem possuir protocolos próprios, à Plataforma *InterSCity* usando padrão *REST*, e provendo um forma de cadastrar recursos na plataforma. O *IoTVirtualizer*, por sua vez, realiza a virtualização de recursos com base nos conceitos de *Resource* e *Capabilities* da plataforma *InterSCity*. Um *Resource* possui definições de um recurso de *IoT* cadastrado na plataforma, e sensores, que podem ser definidos a partir das primitivas contidas na plataforma. As *Capabilities* possuem os dados referentes ao cadastro de um recurso e juntamente com uma definição para processamento de dados, essa definição contem a operação a ser realizada, e o tipo de dados utilizados para esse cálculo. Tendo esses serviços, um desafio encontrado foi a pesquisa de de sensores, nessa caso *IoTGateway*, para criar virtualizes, anteriormente foi criado um serviço de diretórios para armazenamentos dos locais do *IoTGateway* [dos Santos 2022].

Contudo, outro desafio encontrado foi a complexidade de sensores em uma cidade inteligente, que pode ser divididos por exemplo em bairros, setores, prédios, e salas. Com isso, desensolvemos um “Gerente de gerente” e dissidimos nomear o gerente responsável diretamente com o *IoTGateway* e *IoTVirtualizer* de *IoTManagerHigh* e os responsável por organizar esse *IoTManagerLeaf* e outros *IoTManagerHigh* um exemplo desse protocolo pode ser visto na figura 1.

Com essa protocolo foi possível criar um sistema de pesquisa com arquitetura hierárquica de recursos de forma Mas também, outro desafio encontrado foi a complexidade de sensores em uma cidade inteligente, que pode ser divididos por exemplo em bairros, setores, prédios, e salas. Com isso, desensolvemos

um “Gerente de gerente” e decidimos nomear o gerente responsável diretamente com o *IoTGateway* e *IoTVirtualizer* de *IoTManagerLeaf* e os responsável por organizar esse *IoTManagerLeaf* de *IoTManagerHigh* e um exemplo desse protocolo pode ser visto na figura 1. Com essa protocolo foi possível criar um sistema de pesquisa com arquitetura hierárquica de recursos de forma eficiente e escalável.

1.3. Trabalhos relacionados

No artigo de [Ali et al. 2017], aborda a Qualidade de Serviço (QoS) como um elemento crucial para a capacidade de um sistema ou serviço de satisfazer os requisitos específicos dos usuários em termos de desempenho, confiabilidade, latência, largura de banda e outros parâmetros relevantes. Também propõem o uso de *Senaas*, sensor como um serviço, e o uso de *BSA - Backtracking Search Optimization Algorithm* como uma abordagem para melhorar o desempenho do *Senaas* em tempo real, *BSA* é uma abordagem de otimização que pode melhorar a QoS ao encontrar soluções que maximizam o desempenho do sistema de acordo com métricas específicas, como latência mínima ou máxima eficiência.

Também, outro artigo que tem o foco na QoS [Khansari et al. 2018], propõe um método de otimização baseado em algoritmos genéticos quânticos para selecionar e combinar serviços de IoT de maneira a maximizar a qualidade geral da composição, levando em consideração parâmetros como disponibilidade, confiabilidade, atraso e preço.

Nessa artigo ficamos da pesquisa na criação de virtualizadores, e uma arquitetura que facilita a pesquisa de recursos.

2. Metodologia

Inicialmente, realizará uma revisão bibliográfica com o objetivo de adquirir um maior conhecimento sobre os conceitos e o estado tecnológico da *fog* [Mahmud et al. 2018][Bellavista et al. 2019][Puliafito et al. 2019]. Em seguida, criamos a arquitetura do *IoTManagerLeaf*, e implementamos seu conceito, após isso, vimos a necessidade do *IoTManagerHigh*, elaboramos e implementamos sua arquitetura, e otimizamos e a melhor forma de interação com os serviços já existentes.

Com um sistema funcional, elaboramos alguns testes básicos de desempenho sobre a complexidade da árvore de *Managers*. E seguindo para realizará outra revisão bibliográfica, em sistemas similares, para podemos ter meio de comparação.

3. Resultados

3.1. Arquitetura

Na figura 2 temos o fluxo de dados entre *Gateway*, *Virtualizer* e *InterSCity*, nosso objetivo é criar um sistema para alocação de recursos, sensores e dispositivos de forma eficiente e dinâmica. Inicialmente, como visto no diagrama de sequência da figura 6, que foi o trabalho feito até [dos Santos 2022]. Para o *setup* iniciaremos o *IoTGateway* e *IoTVirtualizer* fornecendo o endereço do *Directory Service* que está contido no *IoT Manager Leaf*, outros *IoT Manager Leaf* em uma

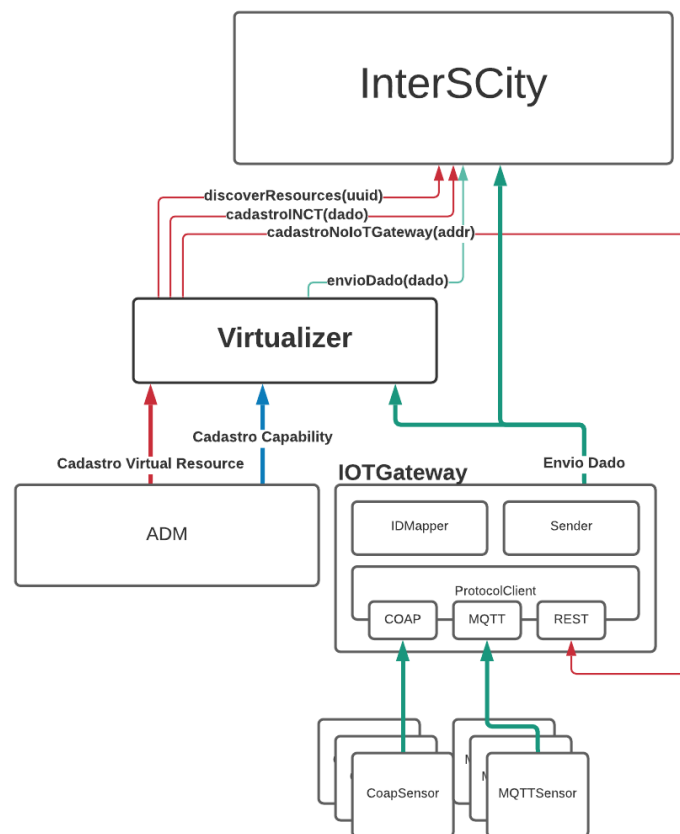


Figura 2. Fluxo entre *Gateway*, *Virtualizer* e *InterSCity*
[R. Silva 2021]

rede local também podem se cadastrar no *Directory Service*, tendo como objetivo redundância na rede, para que mesmo com instabilidades de redes, os usuário possa acessar e gerenciar os dispositivos da rede local, nesse caso somente um *Directory Service* é iniciado na rede e no inicial do *IoT Manager Leaf* o usuário tem a opção de iniciar o *Directory Service* ou se cadastra em um já ativo.

Após a inicialização dos serviços, o *Manager Leaf* cria uma interface web usando *Flask*, um *framework web* em *Python* para criar aplicativos web, figura 3. Nessa página, é possível fazer consultas e acessar outras aplicações conectadas ao *Manager*, figura 4, e também registrar *Capabilities* e *Resources* nos *Virtualizer*. Por sua vez o *IoT Manager High* figura 5, somente possui opção de gerenciar outros *Managers* podendo consultar os “pais e filho” na estrutura na árvore e cadastrar novos filhos, criando uma estrutura como na figura 1.

Logo, o processo para o cadastro de no *Virtualizer* se dá como o diagrama de sequência da figura 7, pela interface ou requisições REST. A consulta de dados para registros podem ser feita no próprio *Manager Leaf*, das aplicações cadastradas no *Manager* ou outras aplicações cadastradas em outros *Manager* através do sistema de pesquisa. A figura 8 mostra o diagrama de sequência com consulta interna, (“Consultar ADDR” na interface da figura 3) a aplicação utiliza a biblioteca *PeeWee* do *Python* para acesso e manipulação de bancos de dados relacionais, recebendo a requisição consultado o banco de dados e retornados o HTML, a figura 4 mostra um exemplo de retorno, onde o usuário pode acessar aplicação e consultar os dados necessário. Para a consulta em outros *Manager*, como mostrado do diagrama de sequência da figura 9, é possível acessar outros *Manager* e outras aplicações nele cadastrados, para isso implementamos sistema de endereços, que é possível ser visto na figura 3 e 5 abaixo do nome do serviço, a partir desse endereço podemos pesquisar que retornará um *json* com, todos os serviços cadastrados.

3.2. Desempenho

A seção de desempenho visa avaliar a eficiência e a escalabilidade do sistema proposto, a latência é uma métrica crítica em sistemas em *fog* e, neste contexto, nossa análise concentra-se em entender o tempo de respostas dos *Manager*, durante uma pesquisa, dependendo da distância dos dispositivos e altura da árvore. A avaliação de desempenho foi realizada em rede local, criando uma árvore binária e medindo o tempo de resposta para cada folha dessa árvore. Nosso objetivo era entender como o sistema responde às com a evolução a complexidade do sistema e identificar eventuais gargalos. [Jain 1991]

Na Figura 10, criamos um gráfico com os resultados coletados, sendo uma árvore binária, usamos um *Manager Leaf* na extremidade da árvore para fazer as requisições de todas as outras folhas no sistema, em cada folha fizemos 10 requisições, com 3 segundos de intervalo, e fizemos uma média do tempo de resposta dessa folha e que corresponde ao *x* no gráfico. A partir desses dados montamos esse gráfico, com média e mediana do tempo de resposta e os intervalos com tempo máximo e mínimos possíveis.

Então, para um sistema real, consideramos que a árvore não passaria muito da altura 5, e que ele seria mais larga, contendo vários dispositivos cadastrados em cada nó em vez de somente 2, o que tornaria os tempos de resposta

IoTManager:

/1.1/2.1

Consultar ADDR:

Virtualizer

Gateway

Manager

Parent

Consultar Dados:

Gateway:

HOST:PORT

Consultar

Virtualizer:

HOST:PORT

Consultar

Registrar no Virtualizer:

Capability:

HOST:PORT

Nome

Description

Capability type

Association

Registrar

Recurso:

UUID

Capability 1

Capability 2

Status

Latitude

Longitude

Registrar

Figura 3. Interface web da página principal do *IoT Manager Leaf*

id	ipManager	portManager	registerTime		
1	172.23.57.140	9000	2023-05-29 07:38:54.497253	DELETE	OPEN »
2	172.23.57.140	9000	2023-05-29 07:51:08.917105	DELETE	OPEN
3	172.23.50.213	9000	2023-05-29 14:46:48.262285	DELETE	OPEN
4	172.23.50.213	9000	2023-05-29 14:47:48.626737	DELETE	OPEN

Figura 4. Interfase web de consulta do *IoT Manager Leaf*

IoTManagerHigh:

/1.1

Consultar:

Parent

Childrens

Registrar:

Child:

IP

PORT

Description

Registrar

Figura 5. Interface web da página principal do *IoT Manager High*

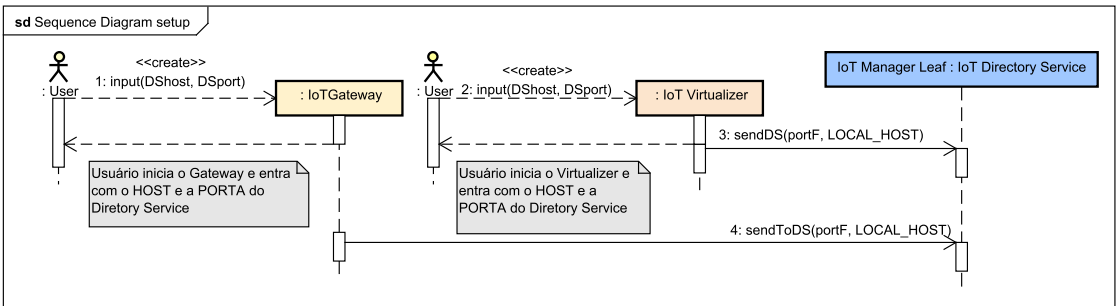


Figura 6. Diagrama de Sequência do *Setup*

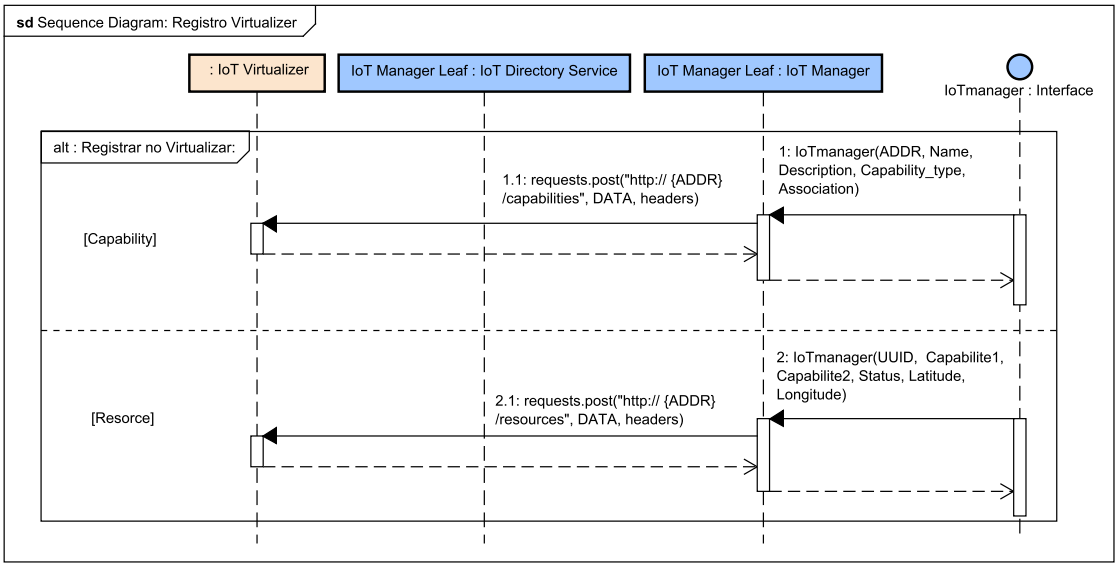


Figura 7. Diagrama de Sequência: Registro no *Virtualizer*

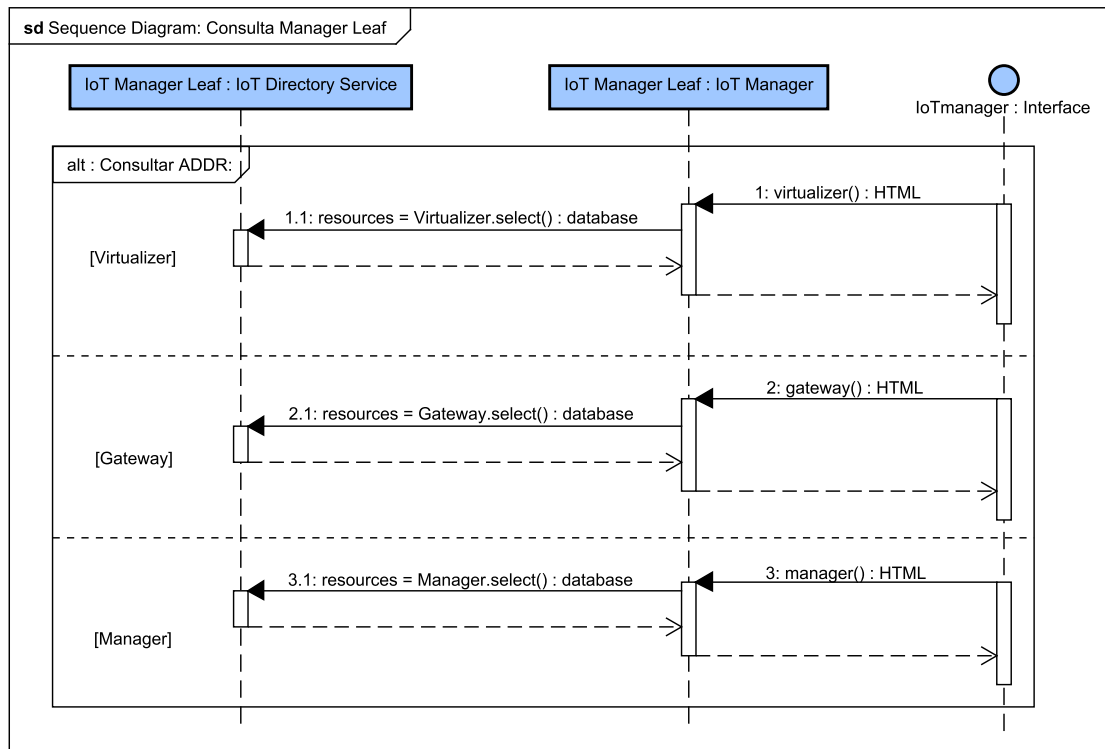


Figura 8. Diagrama de Sequência: Consulta Interna no *Manager*

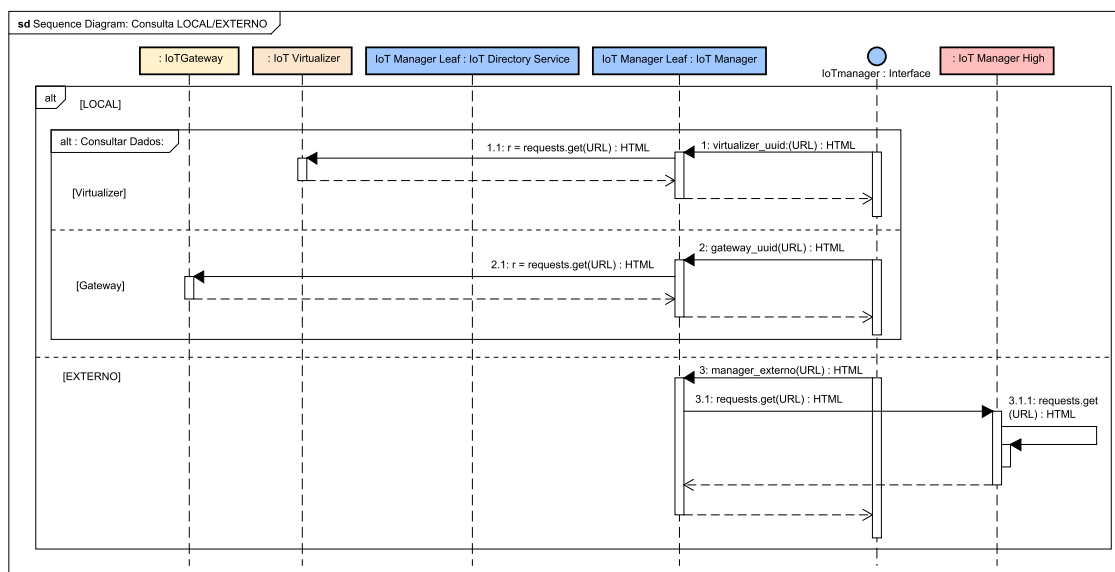


Figura 9. Diagrama de Sequência: Consulta de outros Manager

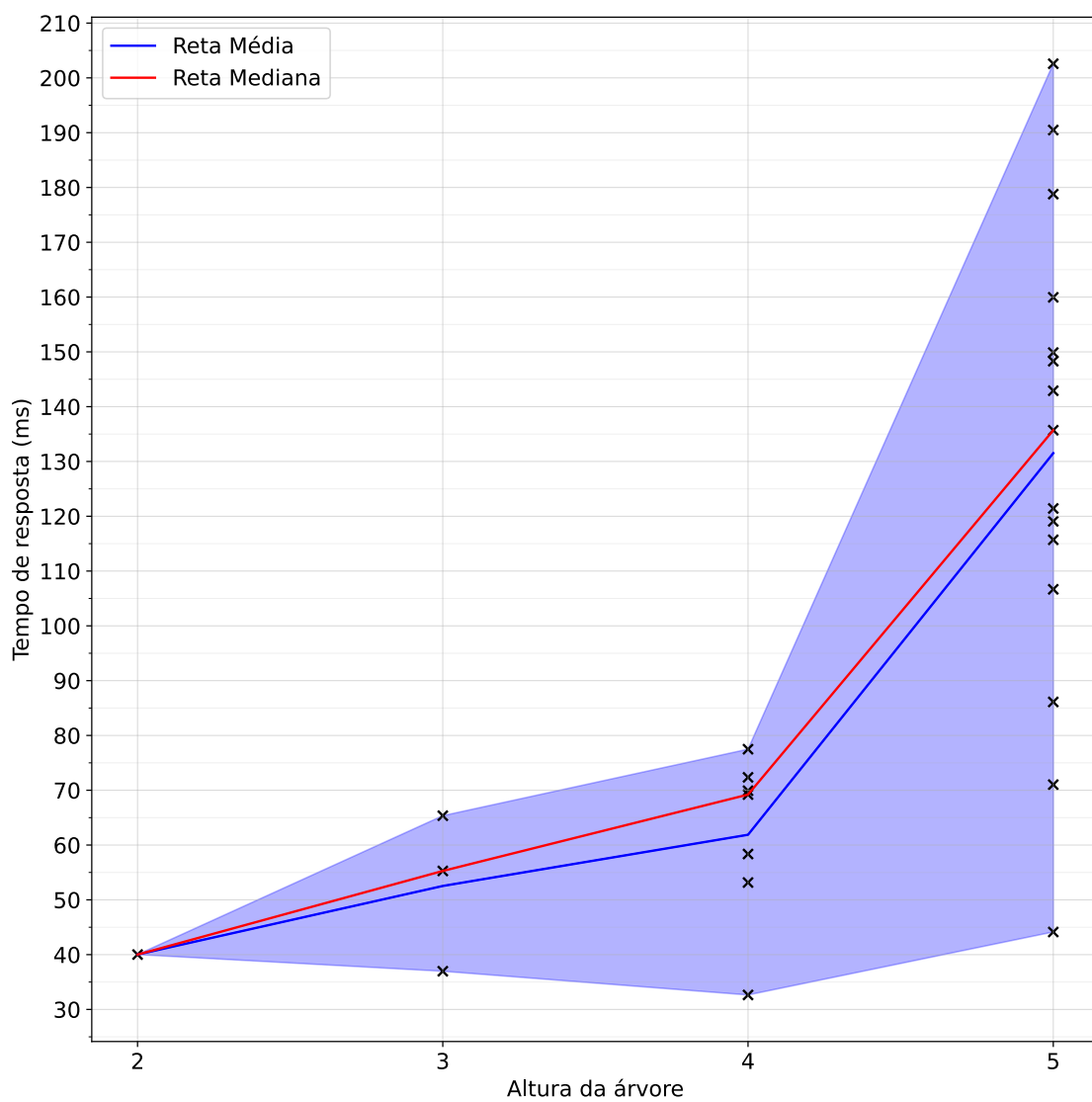


Figura 10. Gráfico do intervalo de tempo de resposta por altura da árvore de *Managers*

mais próximos da media, considerando também que em casos reais o a virtualização de recursos será feitos normalmente com recursos próximos o que e tenderá a tempos de pesquisa mais uniforme entre mesmo em arvores mais complexas.

4. Considerações finais e trabalhos futuros

O escopo inicial do projeto era para alocação sob demanda de múltiplas instâncias do Virtualizador, contudo se viu necessário a contação de gerencia de dispositivos, uma vez que para a alocação sob demanda é necessário um sistema escalonável e eficiente de criação de Virtualizador.

Também, como trabalho futuro, o serviço inicialmente esta ligado exclusivamente a espaços físicos, e não leva em conta serviços moveis sem ligação a um determinado espaço, e a alocação dinâmica de dispositivos móvel e muito importante no contexto de cidades inteligentes.

O código-fonte dos componentes pode ser encontrado em:
https://github.com/VVSRevolution/IC_code

Referências

- [Ali et al. 2017] Ali, Z., Ali, H., and Badawy, M. (2017). A new proposed the internet of things (iot) virtualization framework based on sensor-as-a-service concept. *Wireless Personal Communications*, 97.
- [Bellavista et al. 2019] Bellavista, P., Berrocal, J., Corradi, A., Das, S. K., Foschini, L., and Zanni, A. (2019). A survey on fog computing for the internet of things. *Pervasive and Mobile Computing*, 52:71–99.
- [de M. Del Esposte et al. 2019] de M. Del Esposte, A., Santana, E. F., Kanashiro, L., Costa, F. M., Braghetto, K. R., Lago, N., and Kon, F. (2019). Design and evaluation of a scalable smart city software platform with large-scale simulations. *Future Generation Computer Systems*, 93:427–441.
- [dos Santos 2022] dos Santos, V. V. (2022). Uma arquitetura descentralizada para virtualização de sensores em plataforma de cidades inteligentes. Relatório Final do Programa Institucional de Bolsa de Iniciação Científica 2021-2022, Universidade Federal de Goiás (UFG), Instituto de Informática.
- [Jain 1991] Jain, R. (1991). *The art of computer systems performance analysis - techniques for experimental design, measurement, simulation, and modeling*. Wiley professional computing. Wiley.
- [Khansari et al. 2018] Khansari, M. E., Sharifian, S., and Motamedi, S. A. (2018). Virtual sensor as a service: A new multicriteria qos-aware cloud service composition for iot applications. *J. Supercomput.*, 74(10):5485–5512.
- [Mahmud et al. 2018] Mahmud, R., Kotagiri, R., and Buyya, R. (2018). *Fog Computing: A Taxonomy, Survey and Future Directions*, pages 103–130. Springer Singapore, Singapore.
- [Puliafito et al. 2019] 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).

- [R. Silva 2020] R. Silva, R. (2020). Uso de um testbed de rede de sensores para teste e avaliação de aplicações de cidades inteligentes. Relatório Final do Programa Institucional de Bolsa de Iniciação Científica 2019-2020, Universidade Federal de Goiás (UFG), Instituto de Informática.
- [R. Silva 2021] R. Silva, R. (2021). Adaptabilidade e Tolerância a Falhas No Uso De Recursos De IoT Em Uma Plataforma De Cidades Inteligentes. Relatório Final do Programa Institucional de Bolsa de Iniciação Científica 2020-2021, Universidade Federal de Goiás (UFG), Instituto de Informática.



Certificamos que **VINICIO VIEIRA DOS SANTOS**, participou da palestra **“Ciência Aberta e a Propriedade Intelectual: O mais aberto quanto possível, o mais fechado quanto necessário”**, atividade do **Programa Diálogos em Pesquisa e Inovação**, promovida pela **Pró-Reitoria de Pesquisa e Inovação da Universidade Federal de Goiás**, no dia 17 de Novembro de 2022 com 02 (duas) horas de duração.

Goiânia, 17 de Novembro de 2022

Prof.ª Dr.ª Helena Carasek Cascudo

Pró-Reitora de Pesquisa e Inovação



Certificamos que **VINICIO VIEIRA DOS SANTOS**, participou da palestra **“Pesquisa e Inovação em comunidades tradicionais: um dedo de prosa”**, atividade do **Programa Diálogos em Pesquisa e Inovação**, promovida pela **Pró-Reitoria de Pesquisa e Inovação da Universidade Federal de Goiás**, no dia 18 de Maio de 2023 com 02 (duas) horas de duração.

Goiânia, 18 de Maio de 2023.

Prof.ª Dr.ª Helena Carasek Cascudo

Pró-Reitora de Pesquisa e Inovação



Certificamos que **VINICIO VIEIRA DOS SANTOS**, participou da palestra “**Mineração de dados textuais: uma análise temática das teses e dissertações**”, atividade do **Programa Diálogos em Pesquisa e Inovação**, promovida pela **Pró-Reitoria de Pesquisa e Inovação da Universidade Federal de Goiás**, no dia 26 de Maio de 2023 com 02 (duas) horas de duração.

Goiânia, 26 de Maio de 2023.

Prof.ª Dr.ª Helena Carasek Cascudo

Pró-Reitora de Pesquisa e Inovação



Certificamos que **VINICIO VIEIRA DOS SANTOS**, participou da palestra “Uma história que começou nas humanidades”, atividade do **Programa Diálogos em Pesquisa e Inovação**, promovida pela **Pró-Reitoria de Pesquisa e Inovação** da **Universidade Federal de Goiás**, no dia 28 de Abril de 2023 com 02 (duas) horas de duração.

Goiânia, 28 de Abril de 2023.

Prof.ª Dr.ª Helena Carasek Cascudo

Pró-Reitora de Pesquisa e Inovação

II WORKSHOP - ETAPA 1: FUNDAMENTOS - VISÃO GERAL DA WEB 3.0

CERTIFICADO

Vinício Vieira dos Santos

participou da palestra **“II Workshop sobre a Visão Geral da Web 3.0: Revisão Ampliada dos Conceitos e Perspectivas de Desenvolvimento - Etapa 1: Fundamentos”** com carga horária de 4 horas, realizada no dia 11 de agosto de 2023.

Goiânia, 17 de agosto de 2023



PROF. DR. ELIOMAR ARAÚJO DE LIMA

Diretor do Instituto de Informática - UFG