

IMPLEMENTAÇÃO E AVALIAÇÃO DE ALGORITMOS DE CLASSIFICAÇÃO EM AMBIENTE INTEGRADO CLOUDLET-IOT (*INTERNET OF THINGS*).

Isaac Barros da Costa Moreira (Bolsista/Apresentador)¹ – Unifesspa

isaacbarros88@gmail.com

Warley Muricy Valente Junior (Coordenador(a) do Projeto)² - Unifesspa

vmvj@unifesspa.edu.br

Agência Financiadora: CNPq

Eixo Temático/Área de Conhecimento: Estratégias Sensíveis ao Contexto para Paradigmas de Computação Móvel na Borda.

1. INTRODUÇÃO

Os avanços tecnológicos em hardware e vida útil de bateria nos smartphones tem sido vagaroso, resultando em deficiências para suprir as necessidades das novas tecnologias (OTHMAN et al., 2013). Nesse sentido, uma infraestrutura de Computação Móvel na Nuvem, tal como *Cloudlets*, permite que recursos computacionais e de armazenamento de dispositivos móveis com hardware limitado possam ser estendidos para servidores de nuvem localizados próximos dos usuários para melhor disponibilidade e confiabilidade dos serviços prestados, além de economizar energia do dispositivo e melhorar o desempenho de aplicações móveis (PAN; MCELHANNON, 2017). Estes benefícios são garantidos através de operações de *offloading*, ou seja, códigos e dados que demandam poder computacional são migrados de dispositivos móveis para servidores remotos ricos em recursos.

Assim, o presente trabalho teve como objetivo realizar a avaliação de desempenho de um conjunto de algoritmos de classificação de tomada de decisão de *offloading* computacional para um ambiente integrado *cloudlet* e internet das coisas, visando a escolha de um ou mais algoritmos que tragam resultados satisfatórios no *tradeoff* entre consumo de energia e tempo de processamento.

2. MATERIAS E MÉTODOS

Artigos, seminários, *surveys*, artigos com propostas publicados em veículos reconhecidamente de excelência pela comunidade científica e suportados por sociedades e por editoras (PETERSEN et al., 2008) como Elsevier, Springer, além de outras, fizeram parte do referencial teórico da pesquisa (KITCHENHAM; CHARTERS, 2007). No decorrer do projeto foi optado pela utilização de Máquinas virtuais com sistema operacional derivado Linux, com diferentes configurações de hardware e hospedadas nos servidores do CTIC.

Para a execução das operações de descoberta dos recursos de nuvem, descoberta de serviços, e implantação de serviços, foi utilizada uma solução de sockets customizada para ser executada tanto em um dispositivo de IoT, quanto em uma plataforma de nuvem com sistema operacional Linux. Uma aplicação de detecção e reconhecimento facial foi utilizada como ferramenta de *benchmarking* para executar as operações de *offloading* computacional do dispositivo de IoT para a *cloudlet*. Com relação ao banco de dados utilizado para persistir os dados obtidos pelos experimentos, foi utilizado o sistema de gerenciamento de banco de dados MySQL. Por fim, a avaliação e comparação dos algoritmos será executada através da utilização da ferramenta WEKA (*Waikato Environment for Knowledge Analysis*).

3. RESULTADOS E DISCUSSÃO

Os resultados alcançados no período foram os seguintes: Elaboração e finalização da aplicação que se comunica entre IoT (um drone, no caso) e a nuvem: A Figura 1 apresenta a arquitetura desenvolvida, ilustrando

¹Graduando em Engenharia da Computação - Universidade Federal do Sul e Sudeste do Pará

²Doutor em Ciência da Computação - Professor adjunto Nível I na Universidade Federal do Sul e Sudeste do Pará (FACEEL/IGE/Unifesspa).

os relacionamentos e ações feitas pelas entidades, identificadas de forma cronológicas pelos índices numéricos de 1 a 8 (círculos verdes), no qual as relações descritas utiliza-se a seta pontilhada representando a transmissão de dados, e as setas sólidas representando comunicação de controle.

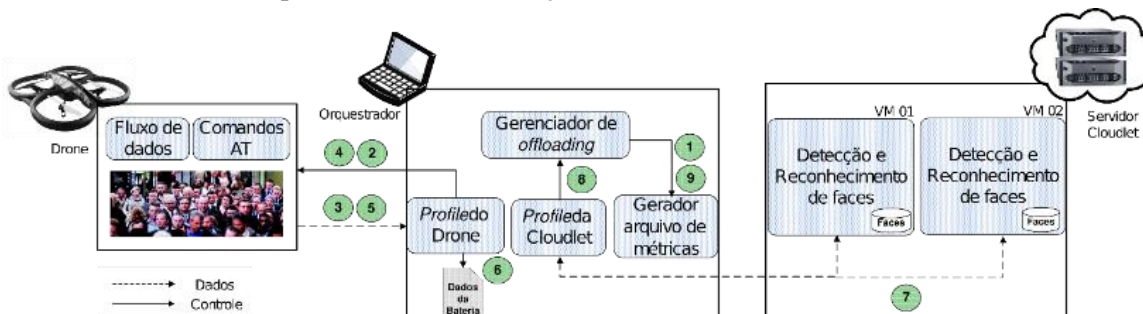


Figura 1. Arquitetura de monitoramento.

Orquestrador: Controla as ações e movimentos do drone, monitora as métricas de medição, gerencia e manipula as informações (dados e controle) coletadas e transmitidas durante todo o processo. Possui quatro componentes: o Gerador de Arquivo de Métricas, responsável pela criação do arquivo de métricas em (1) e inserção dos valores destas no arquivo 9, o *Profile* do Drone, que atua em conjunto com o drone para solicitar o envio das imagens contidas no mesmo (2), dos valores da vazão, tamanho da imagem, RSSI (*Received Signal Strength Indication*) e tempo de transferência 4. O drone também realiza a leitura do arquivo de log que contém o seu percentual de bateria (6). O *Profile* da *cloudlet* orquestra ações de envio e recebimento da imagem oriunda do drone juntamente com o servidor *cloudlet* (7) e o Gerenciador de *Offloading*, responsável pelo controle e recebimento das métricas coletadas ao longo de todo processo (8). **Drone:** Captura e transmite imagens de diferentes tamanhos. Possui dois componentes: Fluxo de Dados, que responde às solicitações do orquestrador para envio das imagens contidas no drone (3) e dos valores da vazão, tamanho da imagem, RSSI e tempo de transferência (5) e Comandos AT que controlam as ações e movimentos do drone por parte do orquestrador. **Servidor Cloudlet:** Plataforma de computação em nuvem. Possui duas VMs (*Virtual Machines*) para execução do algoritmo de detecção e reconhecimento facial, recebimento da imagem com as faces por parte do orquestrador e o envio da mesma imagem após a execução do algoritmo, juntamente com a informação da quantidade de faces reconhecidas (7). Em cada VM existe uma base de dados de 700 faces.

- Avaliação de desempenho:

A Tabela 1 mostra informações detalhadas dos parâmetros utilizados para a avaliação de desempenho do trabalho. Três métricas foram observadas: tempo total de execução remota, taxa de transferência das imagens (vazão), e drenagem de energia. Estas métricas foram mensuradas durante a execução de um total de 30 tarefas (uma tarefa representa uma tentativa de execução remota em uma das VMs no servidor *cloudlet*). As primeiras 10 tarefas foram executadas para o mesmo tamanho de imagem (1,5 MP), e as próximas 10 tarefas para um tamanho de imagem maior (7 MP), e assim por diante.

A avaliação de desempenho foi realizada em dois cenários distintos. O primeiro ocorreu em um ambiente interno sem barreiras, ou seja, não houve qualquer obstáculo entre o drone e o orquestrador. O segundo ocorreu em um ambiente interno com barreiras, ou seja, existe uma parede de obstáculo entre o drone e o orquestrador. Para ambos os cenários, a distância entre estes dois equipamentos foi aumentada de 5 metros (5, 10, 15 e 20 metros) para cada ciclo de 30 execuções das tarefas anteriormente mencionadas.

Tabela 1. Parâmetros do experimento.

Tamanho da imagem (MP)	Quantidade de faces	Algoritmo de detecção e reconhecimento	Número de execuções	Distância (m)
1,5	68	Haar Cascades	10	5
7				10
14				15
				20

Experimento 1 – Cenário interno sem barreiras

A Figura 2 ilustra um conjunto de gráficos variando as métricas de interesse (tempo total, vazão e consumo energético). É verificável o quanto a distância influencia nas três métricas supracitadas. Desse modo,

mais especificamente na Figura 2a, houve um aumento acentuado no tempo total nas 3 últimas tarefas para a distância de 15 metros, enquanto na Figura 2b há uma concentração de dados para velocidade baixa de 200 a 400 kbps. Estes resultados mostram que o efeito da distância entre drone e orquestrador afeta diretamente na vazão e conseqüentemente no tempo de execução remota, uma vez que a imagem precisa ser enviada a partir do drone via rede Wi-Fi para o servidor *cloudlet*. A Figura 2c revela que, mesmo com uma distância um pouco maior (10 metros), o consumo energético do drone foi menor em comparação as demais distâncias - que é justificado pela ótima vazão apresentada na Figura 2b. As Figuras 2d, 2e e 2f mostram os resultados para a seqüência de execução das tarefas partindo da imagem de maior tamanho (14 MP) para a imagem de menor tamanho (1,5 MP). A Figura 2d mostra que houve maiores atrasos no tempo total quando o drone estava distante a 10 e 20 metros do orquestrador, que é justificado pela baixa vazão de ambos conforme ilustrado na Figura 2e. Por outro lado, a drenagem de energia do drone para estas mesmas distâncias também foi maior em comparação as demais.

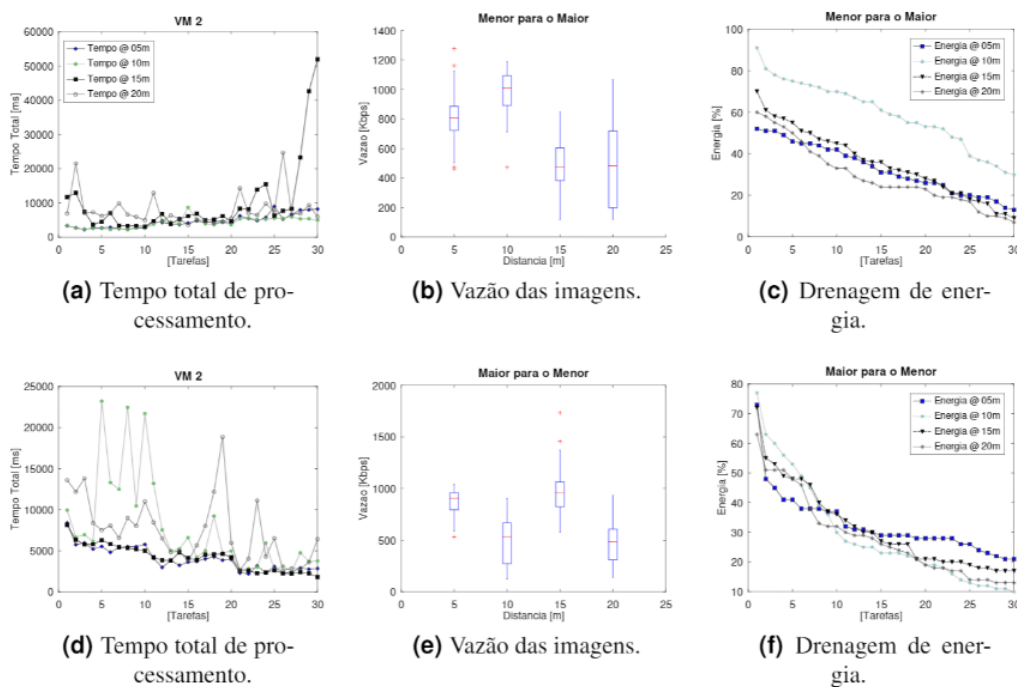


Figura 2. Tempo total, vazão e energia em ambiente interno sem barreiras.

Experimento 2 – Cenário interno com barreiras

As Figuras 3a, 3b e 3c exibem os resultados somente para distância de 5 metros. O gráfico da Figura 3a apresenta o efeito do aumento do tamanho da imagem em relação ao tempo total de execução para as duas VMs (ME-MA-VM1 e ME-MA-VM2) que foi bastante acentuada nas duas últimas tarefas (29 e 30).

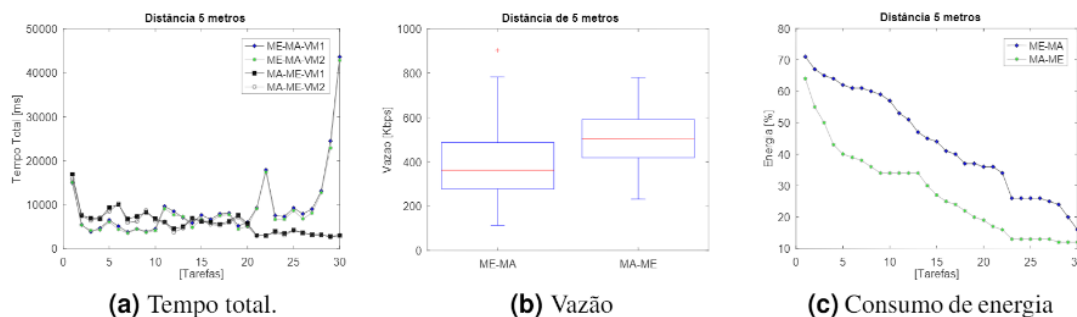


Figura 3. Tempo, vazão e energia em ambiente interno com barreiras.

A Figura 3b evidencia a baixa vazão neste cenário (com barreiras) em relação ao anterior, mas ao mesmo tempo revela uma melhor qualidade de transmissão quando as imagens são processadas do maior



V Seminário de Iniciação Científica

Talentos da Ciência e Tecnologia em ação

Dias 26 e 27 de setembro de 2019

Auditório e Pátio - Unidade II

UNIFESSPA | PROPIT

tamanho para o menor, que em contrapartida, consome energia em menor tempo em comparação com a sequência de processamento partindo da menor imagem para a maior, conforme a Figura 3c.

- Início da inserção de um Raspberry Pi (modelo 3B+) no cenário:

Foi adquirido um Raspberry Pi (modelo 3B+) e iniciado o processo de adaptação ao nosso cenário geral. Após algumas semanas de testes e configuração do dispositivo, foi constatado que o código de reconhecimento facial (o mesmo que é executado nas VMs) necessitaria de adaptações (devido a performance do Raspberry Pi). Até o momento de escrita deste relatório, o código ainda passa por adaptações (o que explica o motivo de que o Raspberry Pi não foi inserido nos resultados deste relatório).

- Criação e alimentação do banco de dados com registros de cenários variados de execução do código de reconhecimento facial na nuvem (acima de 500 registros):

Com o objetivo de obter dados significativos e condizentes com a realidade, foi realizado um processo de construção e alimentação do banco de dados. Tal coleta de dados possui o objetivo de fornecer informações para o treinamento de um algoritmo responsável pela escolha de *offloading* computacional entre os diferentes locais de processamento, baseado nas métricas apresentadas. A Figura 4 demonstra o modelo do banco de dados utilizado para guardar os dados de cada uma das execuções realizadas no período de testes.

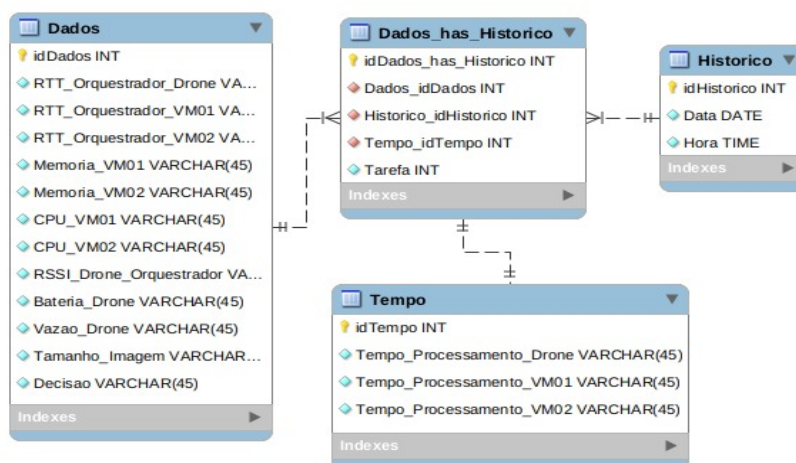


Figura 4. Modelo do banco de dados utilizado.

4. CONSIDERAÇÕES FINAIS

Portanto, é verificável a importância das métricas abordadas nesta pesquisa para todo o processo de *offloading* computacional. Assim, é notável a influência direta da distância no tempo total de processamento, vazão e consumo energético, sejam efeitos diretos na vazão ou na energia. No presente estado do projeto de pesquisa, pode-se concluir que o mesmo teve um bom desempenho, alcançando a grande maioria dos objetivos propostos no seu início e produzindo resultados satisfatórios. Como perspectivas futuras, pretende-se incluir de modo satisfatório o Raspberry Pi no cenário de execução para gerar mais dados condizentes com a realidade.

REFERÊNCIAS

OTHMAN, Mazliza et al. A survey of mobile cloud computing application models. **IEEE Communications Surveys & Tutorials**, v. 16, n. 1, p. 393-413, 2013.

PAN, Jianli; MCELHANNON, James. Future edge cloud and edge computing for internet of things applications. **IEEE Internet of Things Journal**, v. 5, n. 1, p. 439-449, 2017.

KITCHENHAM, Barbara; CHARTERS, Stuart. Guidelines for performing systematic literature reviews in software engineering. 2007.

PETERSEN, Kai et al. Systematic mapping studies in software engineering. In: **Ease**. 2008. p. 68-77.