

Análise de Desempenho da Técnica de *Offloading* de Imagens com Drone em Ambiente de Computação em Névoa

João Barbosa¹, Isaac Barros¹, Wendel Serra¹, Warley Junior¹,
João Carmona¹, Hugo Kuribayashi¹

¹Núcleo de Estudos em Redes e Sistemas Inteligentes (NERSI)
Universidade Federal do Sul e Sudeste do Pará (UNIFESSPA)
Marabá – PA – Brasil

{joaobarbosa, isaacx, wendel.barros, wmvj, jvictor, hugo}@unifesspa.edu.br

Resumo. Apesar dos avanços recentes que tornaram o uso de UAVs (*Unmanned Air Vehicles*) em diversos cenários uma realidade, há vários desafios a serem enfrentados para explorar todo o seu potencial. O uso de *Fog Computing* se torna uma saída para auxiliar estes dispositivos móveis com energia e hardware limitados a terceirizar toda a capacidade de processamento para um recurso remoto na borda da rede. Este trabalho apresenta a avaliação de desempenho da técnica de *offloading* com uso de drone e servidor *cloudlet* em ambiente de *fog computing* para a transmissão e processamento de imagens através de algoritmo tradicional de detecção e reconhecimento facial em tempo real.

1. Introdução

Drone, por definição, compreende-se como um veículo aéreo não tripulado, também chamado de UAV (*Unmanned Aerial Vehicle*). Com sua arquitetura principal composta de sistemas de voo, controle, monitoramento, processamento de dados e de pouso, os drones atuais possibilitam várias funcionalidades, tais como: navegação, aquisição de dados, processamento de imagens e transferência de informações para estações em solo. A arquitetura de computação na névoa (*Fog computing*) vem para fazer uma ponte entre dispositivos de IoT (*Internet of Things*) e a nuvem, permitindo gerenciamento computacional, de rede, de armazenamento e de dados nos elementos da rede próximos ao dispositivo de IoT [Yousefpour et al. 2019].

Partindo desse pressuposto, é notável que o uso de UAVs é capaz de promover benefícios na área de segurança, possibilitando que forças policiais, baseadas na vigilância em tempo real provida pelos drones, utilizem esses dispositivos com o intuito de facilitar o reconhecimento e captura de criminosos [Mohamed et al. 2018].

Este trabalho apresenta a realização e avaliação de um estudo de caso da aplicação de *offloading* computacional com uso de Drone e servidor *cloudlet* em ambiente de *Fog computing*, para a transmissão de imagens e processamento a partir de um algoritmo de detecção e reconhecimento facial em tempo real. Além disso, serão exploradas questões como consumo energético, tempo de execução e taxa de transmissão de imagens. O presente artigo está organizado da seguinte forma: a Seção 2 contém os trabalhos relacionados, a Seção 3 descreve a arquitetura de monitoramento utilizada nos experimentos. Na Seção 4 são apresentadas a metodologia e avaliação bem como o ambiente experimental. Por fim, a conclusão e trabalhos futuros são apresentados na Seção 5.

2. Trabalhos Relacionados

Esta seção apresenta um levantamento sobre os principais trabalhos existentes na literatura que focam na utilização de sistemas de *offloading* computacional (juntamente com UAVs) com o objetivo de aumentar o desempenho das aplicações propostas. O trabalho de [Luo et al. 2015] baseia-se em algoritmos que avaliam parâmetros da rede antes de realizar *offloading* com o uso de um escalonador sensível ao contexto responsável pela decisão do envio dos dados. [Lima et al. 2018] explora a comunicação realizada a partir de um módulo embutido em uma câmera armadilha que processa e envia dados para um ponto de acesso embarcado em um drone, considerando fatores como consumo energético da câmera, autonomia da bateria do UAV e a eficiência da rede sem fio no ambiente florestal.

A proposta de [Loke 2015] utiliza *clusters* de drones para realização de *offloading* computacional, tendo como parâmetros o tamanho da tarefa, largura de banda e vazão da rede sem fio. [Mohamed et al. 2017] propõe a UAVFog, uma plataforma de *fog computing* que usa o UAV como um nó da névoa (*fog node*), possuindo poder computacional, de armazenamento e de comunicação, de acordo com a necessidade e viabilidade do cenário. [Messous et al. 2017] utiliza a técnica de *offloading* computacional considerando o *trade-off* entre tempo de execução e consumo de energia de uma frota de pequenos drones.

Analisando os trabalhos supracitados, percebe-se que nenhuma das propostas contempla o desenvolvimento de uma arquitetura de monitoramento e coleta de métricas em um ambiente integrado UAV-Fog, bem como avaliação da técnica de *offloading* de dados a partir de três perspectivas, qualidade de comunicação, consumo energético e processamento remoto.

3. Arquitetura de Monitoramento

A Figura 1 apresenta a arquitetura desenvolvida, ilustrando os relacionamentos e ações feitas pelas entidades, identificadas de forma cronológica 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.

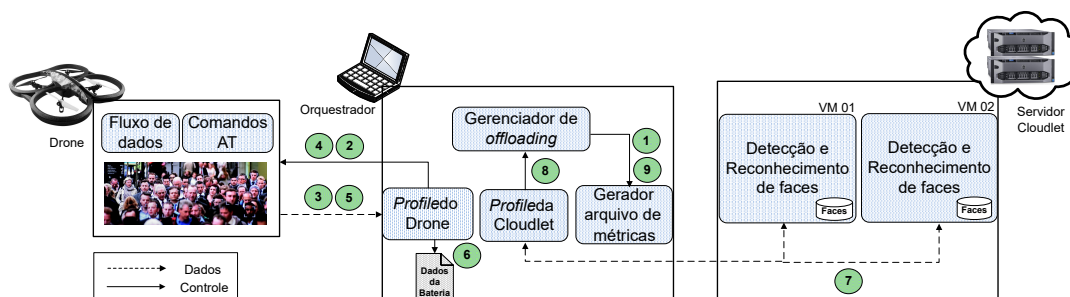


Figura 1. Arquitetura de monitoramento.

- **Orquestrador:** Controla as ações e movimentos do drone, monitora as métricas de medição, gerência e manipula as informações (dados e controle) coletadas e transmitidas durante todo o processo. Possui quatro componentes: o Gerador de Arquivo de Métricas, que é responsável pela criação do arquivo de métricas no disco ① e a inserção dos valores destas no respectivo arquivo ⑨, o *Profile* do

Drone, que atua em conjunto com o drone para solicitar o envio das imagens contidas no mesmo ②, e 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* que é o componente 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 as 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*) com a finalidade de execução do algoritmo de detecção e reconhecimento facial, o 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.

4. Análise de Desempenho

Esta seção apresenta o ambiente de desenvolvimento e da análise comparativa, como também dos resultados obtidos no presente trabalho.

4.1. Ambiente e Metodologia de Avaliação

A Tabela 1 detalha as configurações de hardware e software do experimento. Um destaque para o uso da biblioteca OpenCV. Através dela foi possível realizar o processo de detecção e reconhecimento facial das imagens. Nesse processo, a imagem quando chega em uma das VM's é automaticamente capturada de seu diretório por um *script* Java que, primeiramente, com o auxílio de um algoritmo classificador previamente treinado, realiza a detecção de todas as faces presentes naquela imagem. Ao realizar esse processo e armazenar todas as faces detectadas, um outro algoritmo classificador previamente treinado, realiza um primeiro processo de reconhecimento, conferindo a cada imagem armazenada um rótulo identificador.

A Tabela 2 mostra informações detalhadas dos parâmetros utilizados para a avaliação de desempenho do trabalho. Três métricas foram observadas: tempo total de execução remota (tempo de solicitação da imagem ao drone, resposta do drone e processamento na *cloudlet*), taxa de transferência das imagens (vazão), e drenagem de energia. Estas métricas foram mensuradas durante a execução de um total de 30 tarefas¹. 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,

¹Uma tarefa representa uma tentativa de execução remota em uma das VMs no servidor *cloudlet*.

Tabela 1. Configuração de hardware e software nos ambientes de execução.

Equipamento	Hardware utilizado	Software utilizado
Drone	Processador: ARM Cortex A8 1 GHz 32-bit RAM: DDR2 1 GB @ 200 MHz; Wi-Fi b/g/n; Bateria: 1000mAh	Linux 2.6.32
Orquestrador	Processador: Intel (R) Core i5-4300M CPU @ 2.60GHz x 4; RAM: 4GB;	Ubuntu 18.04.2 LTS NetBeans 8.2 CV Drone
Cloudlet	VM1: vCPU E5-2623 v3 @ 3.00GHz x 2; RAM: 4GB;	Ubuntu 18.04.1 LTS OpenCV 3.2.0
	VM2: vCPU E5-2623 v3 @ 3.00GHz x 4; RAM: 8GB;	Ubuntu 18.04.1 LTS OpenCV 3.2.0

ou seja, existe uma parede de obstáculo entre o drone e orquestrador. Para ambos os cenários, a distância entre estes dois equipamentos foi aumentando de 5 metros (5, 10, 15 e 20 metros) para cada ciclo de 30 execuções das tarefas anteriormente mencionadas.

Tabela 2. 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

4.2. Experimento 1 - Interno sem barreiras

A Figura 2 ilustra um conjunto de gráficos variando as métricas de interesse (tempo total, vazão e consumo energético). Nas Figuras 2a, 2b e 2c, a sequência de execução das tarefas foi partindo da imagem de menor tamanho para a imagem de maior tamanho, ou seja, no intervalo de 1 a 10 a imagem de tamanho 1,5 MP foi executada 10 vezes, e assim sucessivamente para as demais imagens.

É possível entender o quanto a distância influencia nas três métricas mencionadas. Mais especificadamente na Figura 2a, houve um aumento acentuado no tempo total nas 3 últimas tarefas para a distância de 15 metros, enquanto que 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 consequentemente 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 sequência de execução das tarefas partindo da imagem de maior tamanho (14 MP) para a imagem de menor tamanho

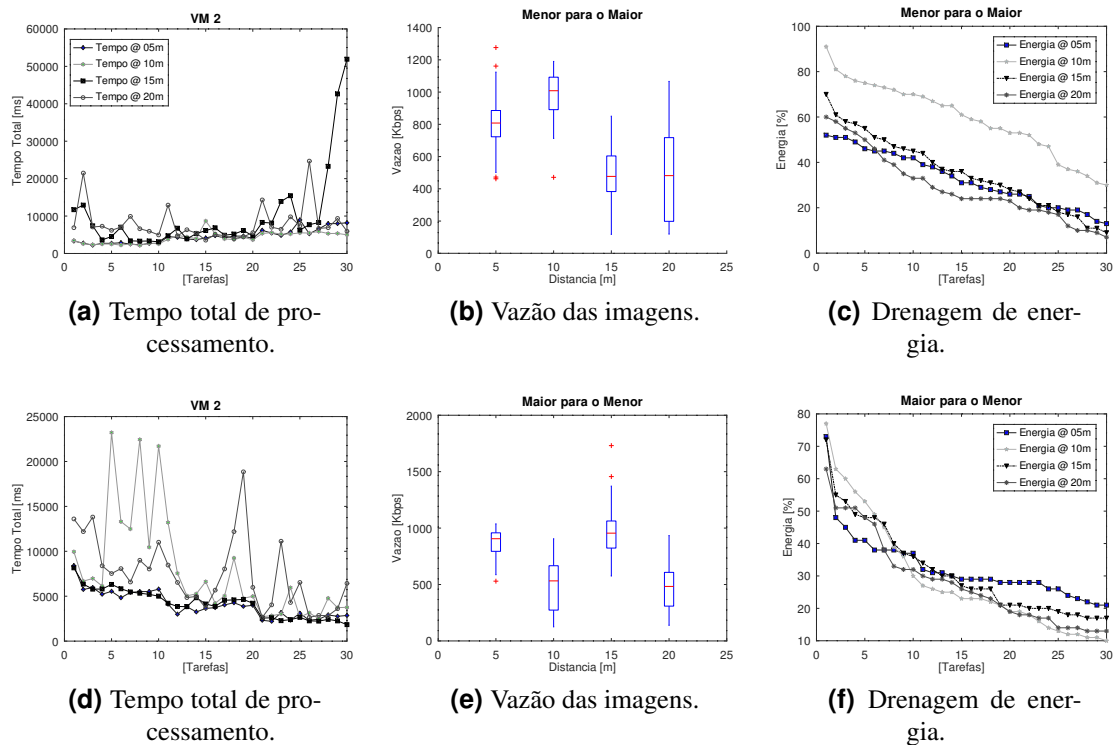


Figura 2. Tempo total, vazão e energia em ambiente interno sem barreiras.

(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.

4.3. Experimento 2 - 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).

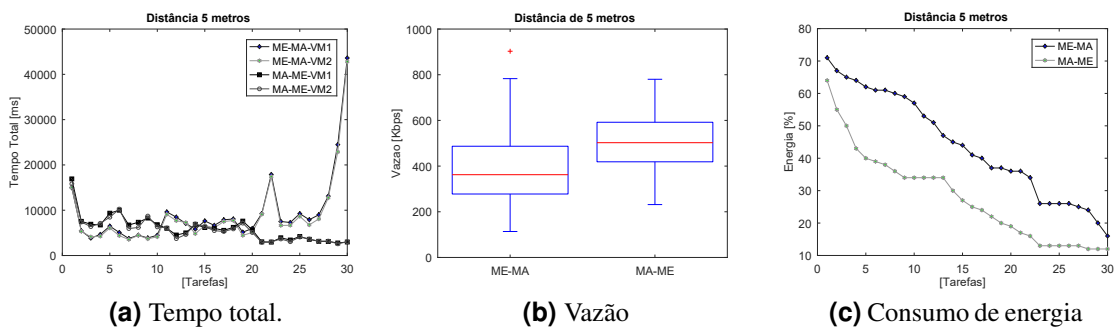


Figura 3. Tempo, vazão e energia em ambiente interno com barreiras.

A Figura 3b evidencia a baixa vazão neste cenário com barreira em relação ao anterior, mas ao mesmo tempo revela uma melhor qualidade de transmissão quando as

imagens são processadas do maior 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.

Os resultados dos testes realizados para as distâncias de 10, 15 e 20 metros, não foram possíveis de serem plotados para este artigo, devido a perda (ou qualidade) de comunicação entre o drone e o orquestrador, ocasionado pelo rápido esgotamento da energia do drone antes de o ciclo das 30 execuções das tarefas serem finalizadas.

5. Conclusão

Este artigo apresentou um estudo sobre a aplicação da técnica de *offloading* com a utilização de drone e servidor cloudlet em ambientes de *fog computing* para a transmissão e processamento de imagens de tamanhos diferentes e aplicação de algoritmo tradicional de detecção e reconhecimento facial. O trabalho mostrou que a eficiência do uso de recursos da cloudlet em diferentes cenários, pode apresentar resultados satisfatórios em relação ao consumo energético do drone e o tempo de processamento das faces das imagens. Como trabalhos futuros pretende-se melhorar a arquitetura do orquestrador com a alimentação do banco de dados para auxiliar nas tomadas de decisão de onde processar as imagens com as faces (MicroFog, Nuvem Pública), baseados em informações contextuais.

Referências

- Lima, C. M. d. A., da Silva, E. A., and Velloso, P. B. (2018). Avaliação de desempenho do ieee 802.11 para coleta de grande volume de dados em florestas com auxílio de drones. In *17^o Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance 2018)*, volume 17. SBC.
- Loke, S. W. (2015). The Internet of Flying-Things: Opportunities and Challenges with Airborne Fog Computing and Mobile Cloud in the Clouds. *IEEE Internet of Things Journal*, pages 1–5.
- Luo, C., Nightingale, J., Asemota, E., and Grecos, C. (2015). A uav-cloud system for disaster sensing applications. In *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pages 1–5. IEEE.
- Messous, M.-A., Sedjelmaci, H., Houari, N., and Senouci, S.-M. (2017). Computation offloading game for an uav network in mobile edge computing. In *2017 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- Mohamed, N., Al-Jaroodi, J., Jawhar, I., Idries, A., and Mohammed, F. (2018). Unmanned aerial vehicles applications in future smart cities. *Technological Forecasting and Social Change*, (May):0–1.
- Mohamed, N., Al-Jaroodi, J., Jawhar, I., Noura, H., and Mahmoud, S. (2017). Uavfog: A uav-based fog computing for internet of things. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation*, pages 1–8. IEEE.
- Yousefpour, A., Fung, C., Nguyen, T., Kadiyala, K., Jalali, F., Niakanlahiji, A., Kong, J., and Jue, J. P. (2019). All one needs to know about fog computing and related edge computing paradigms: a complete survey. *Journal of Systems Architecture*.