

# MEC vs MCC: Análise do Desempenho de Aplicações Interativas e de Tempo Real

Micael Soares<sup>1</sup>, Pedro Pinto<sup>2,3,5</sup>, Jorge Mamede<sup>1,4,5</sup>

1000167@isep.ipp.pt, pedropinto@estg.ipv.pt, jbm@isep.ipp.pt

<sup>1</sup> Instituto Superior de Engenharia do Porto - Politécnico do Porto (ISEP-PP), Porto, Portugal

<sup>2</sup> Instituto Politécnico de Viana do Castelo (IPVC), Viana do Castelo, Portugal

<sup>3</sup> Instituto Universitário da Maia (ISMAI), Maia, Portugal

<sup>4</sup> GILT, Porto, Portugal

<sup>5</sup> INESC-TEC, Porto, Portugal

DOI: 10.17013/risti.37.116-131

**Resumo:** A evolução das redes de telecomunicações tem promovido o desenvolvimento de novas aplicações para dispositivos móveis. Algumas destas aplicações exigem requisitos computacionais e energéticos que vão para além das capacidades dos dispositivos móveis. Neste contexto, pode ser utilizada a arquitetura *Mobile Cloud Computing* (MCC), que permite executar as aplicações em *datacenters* na *cloud* e aliviar o processamento nos dispositivos móveis. No entanto, algumas aplicações mais exigentes, e.g. interativas e de tempo real, são mais sensíveis ao atraso no processamento e comunicação da informação. Para estas aplicações, a arquitetura *Mobile Edge Computing* (MEC) pode ser utilizada como uma tecnologia intermédia que disponibiliza recursos computacionais e de armazenamento a partir da periferia da rede. Este artigo apresenta um estudo que avalia o desempenho das arquiteturas MCC e MEC na execução de duas aplicações tomadas como representativas do espectro das aplicações interativas, de tempo real e de processamento intensivo: o *Fluid* e o *FaceSwap*. São apresentados resultados que permitem quantificar o desempenho destas arquiteturas em diferentes circunstâncias.

**Palavras-chave:** Mobile, Cloud, Edge, Computing, Real-time, Interactive, Applications.

## ***MEC vs MCC: Performance Analysis of Interactive and Real-time Applications***

**Abstract:** Telecommunication networks evolution is driving the development of new applications for mobile devices. Some of these applications are resource-intensive and push computational and energy demands of mobile devices beyond the mobile hardware capabilities. In this context, Mobile Cloud Computing (MCC) architecture emerges as a solution for offloading mobile devices that allows to execute

these applications in cloud datacenters thus reducing the processing demand in mobile devices. However, more demanding applications, e.g. interactive and real-time applications, are sensitive to processing and communications delay. For these applications, Mobile Edge Computing (MEC) can be used as an intermediary technology, providing computing and storage resources in the network edge. This paper presents a study carried out to evaluate the performance of MEC and MCC architectures when executing two applications, Fluid and FaceSwap, representative of real time and computing intensive applications. A set of scenarios were designed to quantify the performance of these architectures in different settings.

**Keywords:** Mobile, Edge, Cloud, Computing.

## 1. Introdução

Na última década, o *Cloud Computing* surgiu como um novo paradigma da ciência da computação, que fornece um conjunto de recursos partilhados e com alta disponibilidade, onde se incluem serviços centralizados de processamento, armazenamento e gestão, entre outros (Armbrust et al., 2009)(Zhang et al., 2010)(Araújo & Cota, 2016). Estes serviços são disponibilizados aos utilizadores finais e podem ser utilizados por aplicações e dispositivos que geram informação e depois utilizam a *cloud* para armazenar e/ou processar esta informação. Conforme referido em (Fortes et al., 2016), os serviços de cloud computing têm grande potencial e os seus fornecedores poderão promover a sua adoção de modo a aumentar o reconhecimento e a notoriedade social destas soluções tecnológicas. O crescimento na adoção destes serviços permite responder ao previsível aumento dos dispositivos terminais dos utilizadores. Com paradigmas como a *Internet of Things* (IoT), prevê-se que a quantidade de dispositivos ligada a estes sistemas seja cada vez maior, assim como a quantidade de dados que exigem ser analisados, processados, transformados, armazenados e respondidos em tempo útil.

Em (Dinh et al., 2013) foi proposta a arquitetura *Mobile Cloud Computing* (MCC), que permite que os dados gerados pelos dispositivos móveis sejam transmitidos para a *cloud* com servidores centrais distribuídos geograficamente utilizados para armazenamento e processamento. Com esta arquitetura, à medida que a distância física entre os dispositivos móveis e os servidores aumenta, a latência de transmissão e o atraso de processamento também aumenta, o que pode comprometer a qualidade do serviço e experiência de utilização de algumas aplicações. Em (Qing et al., 2013) os autores propõem uma arquitectura, o CACTSE, para disponibilização de conteúdos a partir de *cloudlets* localizadas em servidores próximos dos equipamentos dos utilizadores e explorando a cooperação entre diversos terminais.

Para estas aplicações mais exigentes em termos de latência de comunicação e atraso de processamento, foi proposta uma arquitetura distribuída designada por *Mobile Edge Computing* (MEC) (Patel et al., 2014). Esta arquitetura é baseada na colocação de servidores na periferia da rede que fornece a capacidade computacional, analítica e de armazenamento, para que os dispositivos móveis tenham esforço computacional distribuído e menor latência e atraso.

Neste artigo apresenta-se uma análise comparativa do desempenho das arquiteturas MEC e MCC, na implementação de duas aplicações consideradas representativas

das aplicações interativas, com requisitos de tempo real e processamento intensivo. São definidos cenários de teste que permitem analisar a influência da latência das comunicações e da capacidade de processamento dos servidores remotos em ambas as arquiteturas, a fim de quantificar o seu desempenho em várias vertentes.

Os resultados obtidos mostram de que forma o MEC pode melhorar a interação do utilizador e a qualidade da experiência, uma vez que coloca o processamento e a computação intensiva em servidores na periferia da rede, mais próximos dos dispositivos móveis.

Este artigo está estruturado em 5 secções. A primeira secção apresenta uma introdução ao trabalho realizado. Na Secção 2 apresentam-se trabalhos de outros autores relacionados com estudo reportado neste artigo. A Secção 3 descreve o cenário e os casos de teste das aplicações seleccionadas. A Secção 4 apresenta os resultados obtidos e as respectivas análises. A secção 5 conclui este artigo.

## 2. Trabalho Relacionado

O *cloud computing* tem sido utilizado como forma de ultrapassar as limitações de processamento e de armazenamento dos sistemas computacionais na periferia da rede.

Em particular, a arquitetura MCC tem como objetivo dar resposta aos desafios de mobilidade dos equipamentos móveis e às suas limitações de recursos como o armazenamento, processamento e baterias (Beck et al., 2014). No que respeita ao tráfego que estes dispositivos móveis geram e comunicam, pela estimativa apresentada em (Blockstrand et al., 2011), até 2020 deverão existir 50 mil milhões de dispositivos inteligentes ligados à rede, com o volume de dados a crescer exponencialmente. De acordo com (Skarpness, 2017), em 2020 é esperada a geração de um tráfego de 1,5 GB de dados por pessoa por dia.

Porém, existem aplicações que, por um lado, produzem grandes quantidades de dados e necessitam de capacidade computacional, mas por outro, exigem interação com o utilizador e são sensíveis à latência e atraso. O potencial dos terminais tem estimulado o desenvolvimento de aplicações cada vez mais interativas, inteligentes e que exploram a mobilidade dos utilizadores tais como as aplicações de realidade aumentada, condução autónoma, ou de jogos em tempo real, que requerem uma conectividade permanente e com débitos de acesso elevados. Este tipo de aplicações pode não ser compatível com o MCC e, é neste cenário, que surgem as arquiteturas de Edge Computing (EC) (Verbelen et al., 2012) (Patel et al., 2014) (ETSI, 2014) (ETSI, 2015) (GSMA Intelligence, 2014) (Mao et al., 2017).

Nas arquiteturas de *Edge Computing (EC)* os servidores são colocados próximo da rede de acesso e dos utilizadores, e são dotados de recursos de processamento, armazenamento e de débito de acesso, que permitem executar as tarefas computacionais mais exigentes, retirando-as dos equipamentos terminais. Existem várias arquiteturas de implementação do EC como, por exemplo, o *Fog Computing (FC)*, as *Cloudlets* e o MEC (*Mobile Edge Computing*).

A arquitetura do FC foi introduzida pela Cisco em 2012 (Bonomi et al., 2012) e foi considerada uma extensão do paradigma da *cloud* que fornece serviços de computação,

armazenamento e rede entre equipamentos terminais e os tradicionais servidores da *cloud*. O *Open Fog Consortium* definiu uma arquitetura distribuída de três camadas (utilizadores finais, nós de neblina e *clouds* centralizadas) em que os elementos comunicam e interagem entre si. O objetivo principal é aproximar os dados do utilizador final (Shenoy et al., 2015), o que reduz a latência, melhora a QoS (Suryawanshi & Mandlik, 2015) e fornece suporte para localização, reconhecimento de contexto e suporte à mobilidade (Aazam & Huh, 2014). O *FC* permite recolher dados e analisar a informação num contexto local, enquanto na *cloud* se coordena e realiza as análises globais para avaliação e satisfação dos requisitos das aplicações. Como a arquitetura *FC* é heterogênea, os serviços podem ser instalados na periferia da rede em servidores locais com débito elevado e acessíveis através de diferentes tecnologias sem fio (Bonomi et al., 2014).

A arquitetura *Cloudlet* foi desenvolvida por uma equipa da Universidade Carnegie Mellon (Satyanarayanan, 2009), que implementou um protótipo no âmbito do projeto Elijah (Elijah, 2009). Esta arquitetura assenta em três camadas compostas por dispositivos móveis, *cloudlets* e *cloud*. Nesta hierarquia, as *cloudlets* são a camada intermediária e consideradas como *datacenters* locais, que disponibilizam serviços de *cloud* localizados, e que proporcionam aos equipamentos dos utilizadores funcionalidades de alto desempenho computacional e de acesso mais rápido aos recursos da *cloud* (Klas, 2016) (Khan et al., 2014). As *cloudlets* são unidades servidoras projetadas para suportar aplicações de dispositivos móveis, executando as tarefas de computação intensiva de aplicações interativas, como, por exemplo, aplicações de tradução em tempo real (Achanta et al., 2012), aplicações de reconhecimento facial em tempo real (Soyata et al., 2012), aplicações de realidade aumentada (Abolfazli et al., 2013), jogos na *cloud* ou sistemas de assistência cognitiva. Nas *cloudlets*, as aplicações têm dois componentes principais, um que é executado no dispositivo móvel e outro na unidade servidora. Existem ainda outros trabalhos tal como (Koukoumidis et al., 2011) que propõem arquiteturas para cache dos serviços na *cloud* que são implementadas nos dispositivos móveis, utilizando uma *pocket cloudlet*.

Considerando as limitações computacionais e energéticas dos equipamentos terminais, a parte móvel da aplicação é a menos exigente em termos de recursos. Assim, é no servidor existente na *cloudlet* que se executam as tarefas da aplicação que necessitam de mais recursos computacionais e de armazenamento.

A filosofia da *cloudlet* foi seguida pelo *European Telecommunications Standard Institute (ETSI)*, que entretanto normalizou a arquitetura MEC (Satyanarayanan, 2013) (Ai et al., 2017). A arquitetura do MEC é baseada numa plataforma de virtualização que permite a execução de aplicações em servidores virtuais colocados na periferia da rede de acesso, ou seja, pretende reduzir o tráfego da rede movendo os recursos da *cloud* para periferia da rede (Shahzadi et al., 2017), explorando de uma forma eficaz uma infraestrutura de sistemas totalmente virtualizada (Simanta et al., 2012). Esta infraestrutura pode ser implementada em servidores virtualizados colocados em locais diferentes, como pontos de acesso sem fio (*AP - Access Points*), em controladores de rede de rádio (*RNC*) ou estações base LTE (eNodeB) das diversas tecnologias de acesso de rádio (*WLAN/3G/LTE*, respetivamente). Esta arquitetura surge como relevante no contexto das tecnologias mais recentes de telecomunicações como o 5G e permite explorar a escala no

contexto do IoT, da Internet Tátil ou da Internet de Mim. Para serviços virtualizados, a MEC traz vantagens como a redução de custos de implementação e a normalização dos procedimentos de gestão e orquestração.

### 3. Configuração dos Cenários de Teste

Para simular as condições de periferia de rede, foi configurado um cenário onde se implementou um servidor numa *cloudlet* instalada numa rede local. Optou-se pelas *cloudlet* uma vez que, na altura dos testes, se tratava da implementação mais avançada e mais estável em termos de desenvolvimento, e aquela que mais se aproximava do modelo idealizado pela ETSI. Para reproduzir o serviço da *cloud*, foi instalado um servidor numa plataforma AWS. O terminal do utilizador interliga-se por wi-fi com a infraestrutura de rede local e é através desta que acede a uma infraestrutura cablada de acesso à *cloud*.

Duas aplicações foram escolhidas que se assumem como sendo aplicações com fortes requisitos de computação e interação em tempo real. A primeira aplicação é o *Fluid*, uma aplicação de teste utilizada em computação gráfica interativa. A segunda aplicação é o *FaceSwap*, uma aplicação em tempo real de reconhecimento facial e troca de imagens. Foram realizados dois conjuntos de testes: um para comparar o efeito da latência dos cenários de *cloudlet* e de *cloud* no funcionamento das aplicações consideradas, e um outro para avaliar até que ponto poderá um aumento da capacidade de processamento no servidor da *cloud* minimizar o efeito da latência no funcionamento das aplicações testadas.

A Figura 1 apresenta a arquitetura de teste utilizada para avaliar o desempenho global das soluções *cloudlet* e *cloud*.

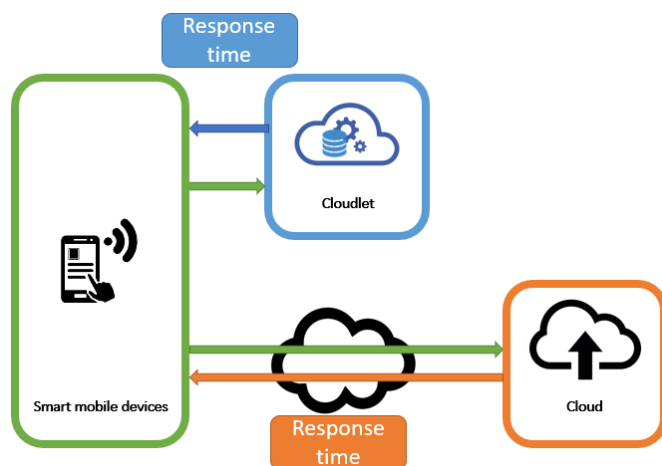


Figura 1 – Bancada de teste

A Tabela 1 resume as principais características dos sistemas utilizados para implementação da *cloudlet* e da *cloud* e do *smartphone* utilizados nos testes. Os dispositivos móveis foram ligados à mesma rede segundo a norma IEEE 802.11n.

Equipamento	Características
<i>Cloudlet</i> <i>Satellite L755-1DR</i>	Intel® Core™ i7 -2670 QM 2.20 GHz, 8 VCPUs 8 GB RAM, 30 GB Virtual Disk, 100 Mbps Ethernet
<i>Cloud</i> <i>Amazon AWS-West</i>	Intel+ R® Xeon E5-2680 v2 2.80 GHz, 32 VCPUs 32 GB RAM, 60 GB Virtual Disk, Amazon Enhanced Network
<i>Mobile Device</i> <i>Samsung S8</i>	Exynos 8895 2.3 GHz Quad + 1.7 GHz Quad 4 GB RAM, 64 GB Flash, 802.11a/b/g/n, WiFi

Tabela 1 – Configuração de Hardware

Para uma análise dos tempos de resposta conseguidos em cada cenário, os testes foram realizados com os servidores instalados em diferentes instâncias de máquinas virtuais na *cloudlet* e na *cloud*, conforme apresentado na Tabela 2.

Estes cenários permitiram a análise da influência do número de núcleos no desempenho da aplicação *Fluid*. Foram lançadas instâncias com 2 e 4 núcleos de CPU, na *cloudlet* e na *cloud*, e foi avaliado o desempenho em ambas as tecnologias. Para o efeito utilizaram-se diferentes instâncias de máquinas virtuais (*Virtual Machines* - VMs) na *cloudlet* e na *cloud*, tendo em conta que os servidores recomendados para a aplicação *FaceSwap* deveriam ter 4 núcleos e 8 GB de RAM (Wang, 2016). Foi considerada a variação do número de núcleos da CPU para permitir avaliar a influência da capacidade de processamento no tempo de resposta das aplicações. As instâncias foram iniciadas com 2, 4, 6 e 7 núcleos na *cloudlet* e com 1, 2, 4, 8, 16 e 32 núcleos na *cloud*. A mesma simulação foi realizada e os dados foram coletados para avaliar o desempenho as duas aplicações.

N.º vCPU/Instância	Cloudlet	Cloud
1 core	1 CPU - 2 GB RAM	1 CPU - 4 GB RAM
2 cores	2 CPU - 4 GB RAM	2 CPU - 8 GB RAM
4 cores	4 CPU - 6 GB RAM	4 CPU - 16 GB RAM
6 cores	6 CPU - 5 GB RAM	
7 cores	7 CPU - 6 GB RAM	
8 cores		8 CPU - 32 GB RAM
16 cores		16 CPU - 64 GB RAM
32 cores		32 CPU - 132 GB RAM

Tabela 2 – Configuração das instâncias

### 3.1. Fluid

*Fluid* é uma aplicação utilizada em demonstrações interativas de computação gráfica e um exemplo representativo de jogos em tempo real. É uma implementação simples de uma simulação de um fluido líquido, utilizando o método de Hidrodinâmica de Partículas Suavizadas (*Smoothed-Particle Hydrodynamics* - SPH) (Solenthaler & Pajarola, 2009). O modelo SPH é um método lagrangiano usado para modelar o fluxo

de um fluido, tratando cada partícula como um elemento discreto desse fluido. O *Fluid* transforma o ecrã do *smartphone* num recipiente onde as partículas se deslocam através dos movimentos provocados no dispositivo. O *Fluid* é dividido numa aplicação cliente executada no terminal móvel de um utilizador, e num servidor que executa a simulação dinâmica de cada uma das partículas do líquido.

A Figura 2 mostra as partículas a oscilar entre os lados do ecrã do *smartphone* com velocidade e direção diferentes devido à interação e movimentos do utilizador. Os dados do acelerómetro do *smartphone* são enviados para um mecanismo gráfico no servidor. Do lado do servidor, a aplicação executa um simulador físico baseado na hidrodinâmica de cada uma das partículas. O resultado da simulação é processado periodicamente e dá a ilusão de um líquido a oscilar dentro do ecrã do terminal móvel.

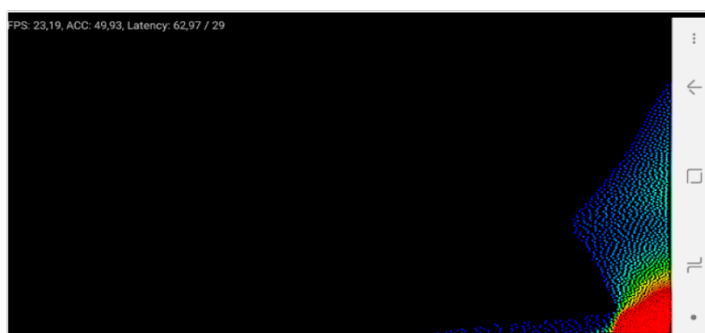


Figura 2 – Aplicação *Fluid*

Nesse cenário, o simulador executa o cálculo da trajetória de 2218 partículas com intervalos de tempo de 20ms, gerando até 50 Frames Por Segundo (FPS). A avaliação de desempenho mede o tempo de resposta do sistema utilizando as soluções de *cloudlet* e de *cloud*. Em ambos os casos, a medição começa quando o dispositivo envia os dados do sensor para o servidor e termina quando o servidor envia de volta a previsão estimada e essa é recebida no terminal. O ciclo de teste mede o tempo de resposta de várias iterações realizadas por um utilizador durante 5 minutos de interação, gerando 300 amostras. A experiência compara também os valores da taxa de FPS.

### 3.2. FaceSwap

O *FaceSwap* é uma aplicação Android que troca o rosto das pessoas em tempo real. O servidor do *FaceSwap* é composto por três camadas com base no rastreamento, deteção e reconhecimento de faces, conforme apresentado na Figura 3.

A aplicação cliente está constantemente a transmitir para o servidor imagens dos rostos no formato JPEG, com a dimensão 640x480. Do lado do servidor, se o reconhecimento do rosto for positivo, são enviadas caixas com as imagens dos rostos de substituição. A aplicação de reconhecimento de rosto detecta rostos nas imagens recebidas e tenta identificar o rosto a partir de uma base de dados pré-preenchida. As experiências realizadas consideram apenas a parte de reconhecimento num sistema

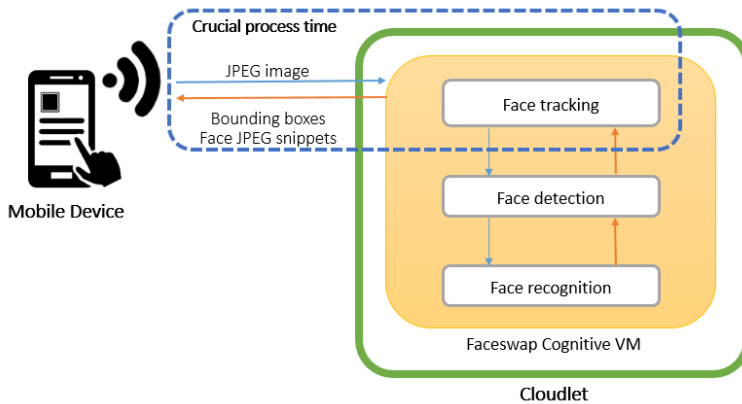


Figura 3 – Framework da aplicação cliente-servidor FaceSwap

treinado, possibilitando atingir tempos de resposta aceitáveis. Assim sendo, o treino dos classificadores e a população da base de dados são trabalhos feitos *offline*. Se as faces detectadas corresponderem às faces presentes na base de dados, uma resposta de comunicação bem-sucedida será enviada para permitir a troca das faces, conforme ilustrado na Figura 4.

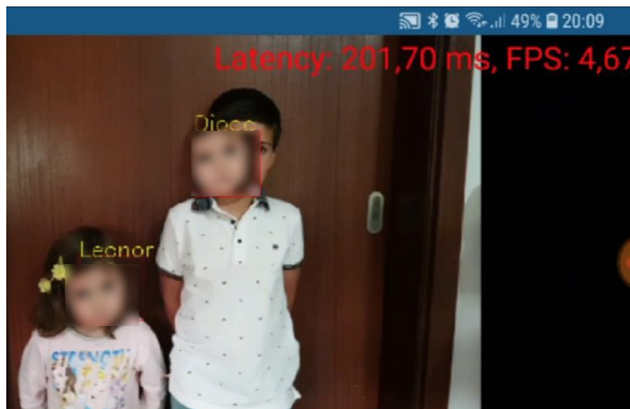


Figura 4 – Aplicação FaceSwap

Esta aplicação foi concebida para efetuar a troca de rostos de forma eficaz, pelo que um servidor FaceSwap deve ter pelo menos 4 núcleos e 8GB de RAM. Os testes do FaceSwap mostraram que a procura do rosto é executada em cerca de 15ms, sendo possível coletar dados a taxas elevadas de transferência de frames de imagens; e assim, obter uma resposta em tempo real. Por outro lado, como a deteção e o reconhecimento dos rostos demoram mais tempo, essas tarefas são efetuadas previamente (*offline*), permitindo que o objetivo principal ainda seja atingível. Quando os resultados ficam disponíveis, novas procuras podem ser iniciadas.



A avaliação de desempenho mede o tempo de resposta do sistema utilizando a *cloudlet* em comparação com a *cloud*, isto é, a diferença de tempo que demora a troca entre a face substituída e a face original. O ciclo de teste possibilita a medição do tempo de resposta e da taxa de frames até chegar às 100 interações. No final de cada ciclo, foram registados os valores da latência e da taxa de FPS. O tempo de resposta determina a qualidade dos gráficos e deveria apresentar valores máximos de 200ms de forma a permitir usufruir de uma aplicação fluida; sendo que, quanto maior for, pior a experiência do utilizador.

## 4. Resultados e Análise

Das experiências realizadas foram obtidos e analisados os resultados. Dentre as plataformas AWS testadas (EC2-Europe, EC2-Asia, EC2-West e EC2-East) escolhida foi o EC2-West (situada em Oregon), uma vez que esta apresenta um tempo médio de ida e volta (*Round Trip Time - RTT*) de 101,5ms e disponibilizou as condições necessárias para a realização dos diferentes testes num contexto académico. A largura de banda média testada foi de 68Mbps. As experiências avaliam a latência e a qualidade do utilizador implementando ciclos de interação que registam diferentes fenómenos que podem ocorrer em condições reais, como limitações de largura de banda, saturação de Wi-Fi e congestionamento, instabilidade de encaminhamento, falha do fornecedor de serviços. Esses ciclos permitem a apresentação e comparação dos resultados com um intervalo confiável sendo que estes resultados são apresentados utilizando a Função de Distribuição Cumulativa (*Cumulative Distribution Function - CDF*).

### 4.1. Análise aos Resultados do Fluid

A Figura 5 e a Figura 6 apresentam os resultados dos testes realizados à aplicação *Fluid* na *cloudlet* e na *cloud*. Um movimento fluido das partículas é obtido se a latência apresentar um valor máximo abaixo de 100ms. Se for superior, o movimento pode parecer mais lento, deteriorando a experiência do utilizador (Ha et al., 2013). Os valores obtidos no cenário da *cloudlet* são melhores, pois os valores medianos do tempo de resposta são de 32 a 34ms, comparativamente com os valores de 124ms a 145ms para a *cloud*, como ilustra a Figura 5.

Na Figura 6 pode verificar-se que a taxa de frames apresenta valores medianos de 44-47 FPS para a *cloudlet* e 18-22 FPS para a *cloud*. Como o servidor continua a gerar até 50 FPS e altera apenas os estados da simulação de acordo com os dados recebidos do acelerómetro, o cenário da *cloudlet* permite uma taxa de FPS duas vezes maior que a *cloud*. Os recursos obtidos permitem constatar que o uso da *cloudlet* pode ter um grande impacto reduzindo o tempo de resposta e aumentando substancialmente a qualidade da experiência do utilizador.

A análise de desempenho com o aumento dos recursos de CPU foi também considerada, através da execução de instâncias de VMs com maior capacidade de processamento (maior número de núcleos) e RAM. No entanto, verificamos que para o caso da aplicação *Fluid* não há uma diminuição significativa nos tempos de resposta, com 1 vCPU, 2 vCPUs ou 4 vCPUs. Para instâncias de VM na *cloudlet* com 1 vCPU e 4 vCPUs, verificamos que

o tempo de resposta reduz cerca de 6% e aumenta da taxa de FPS em 4%. Os valores obtidos na *cloudlet* cumprem com êxito os requisitos da aplicação.

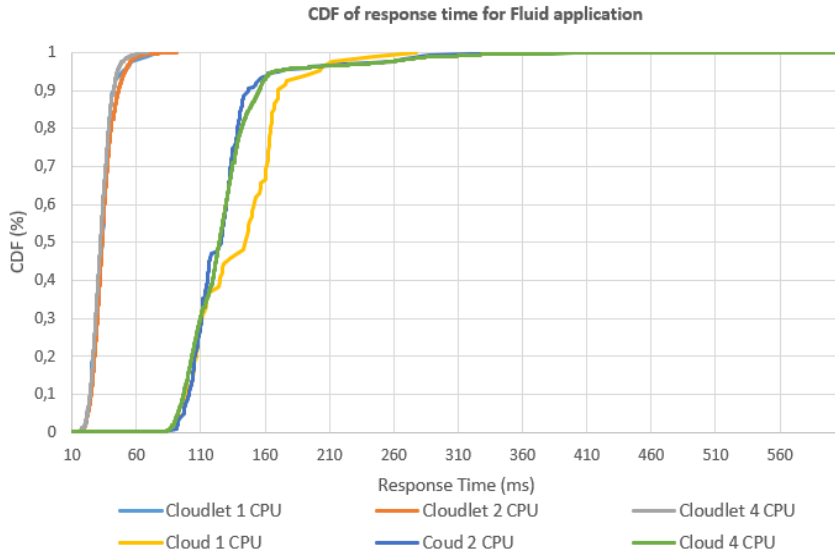


Figura 5 – Comparação dos tempos de resposta - *Fluid*

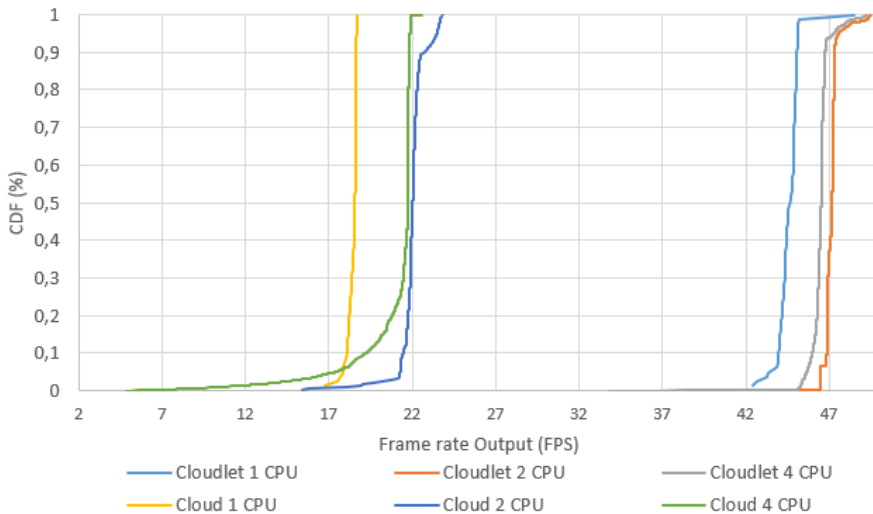


Figura 6 – Comparação das taxas de FPS - *Fluid*

#### 4.2. Análise aos Resultados do FaceSwap

Os testes realizados com o *FaceSwap* permitiram avaliar diretamente os valores do tempo de resposta e da taxa de FPS na *cloudlet* e na *cloud* (AWS EC2-Oregon) com instâncias de VMs com características aproximadas. Por um lado, a plataforma AWS apenas possibilita a utilização de instâncias com valores fixos de processamento (vCPU) e RAM. Por outro lado, as instâncias iniciadas na *cloudlet* apresentam limitações próprias do hardware utilizado. Portanto, não foi possível iniciar instâncias exatamente com a mesma capacidade de processamento (núcleos CPU) e de memória RAM. A experiência analisa também a melhoria do desempenho com o aumento da capacidade de processamento das instâncias da VM.

O processo inicia-se com o lançamento das instâncias da VM do servidor *FaceSwap*. Depois disso, é iniciada a aplicação cliente *FaceSwap* no *smartphone* e selecionado o servidor da *cloudlet*. Note-se que os testes realizados pressupõem que em instâncias anteriores da *cloudlet*, foi executado o treino prévio da aplicação de reconhecimento de rostos e desse modo preenchida uma base de dados com os rostos. A seguir, é selecionada a funcionalidade de troca de rostos, rostos que são listados na base de dados previamente treinados e armazenados.

Na Figura 7 são apresentados os valores dos tempos de resposta da aplicação *FaceSwap* quando executada nos servidores da *cloudlet* e da *cloud*. Os valores obtidos permitem constatar que a performance no cenário *cloudlet* é consideravelmente superior, pois os valores medianos do tempo de resposta do *cloudlet* estão em torno de 38 a 43ms e, para o *cloud*, os valores medianos estão entre os 205ms e os 618ms. Para instâncias de VM com características semelhantes de processamento (2 e 4 CPU), os valores dos tempos de resposta da *cloudlet* são quase 5 vezes inferiores aos valores medidos na *cloud* (AWS EC2 Oregon).

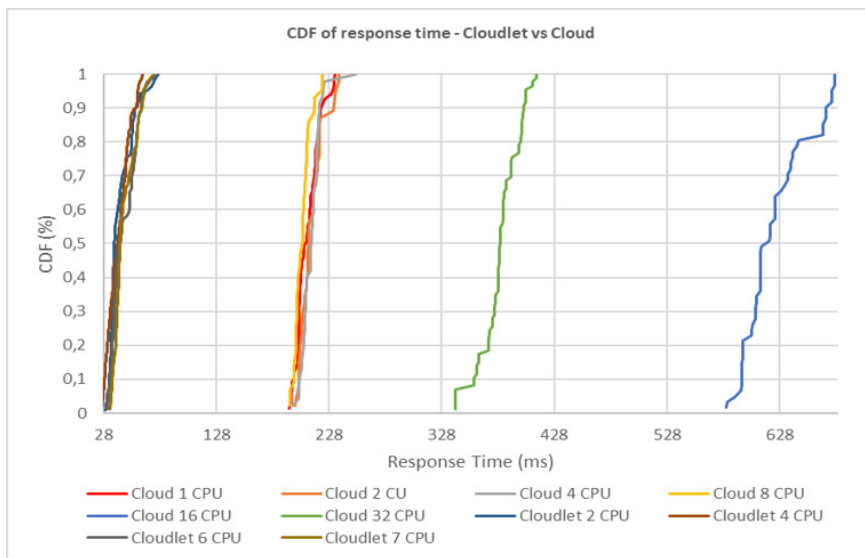


Figura 7 – Comparação dos tempos de resposta - *FaceSwap*

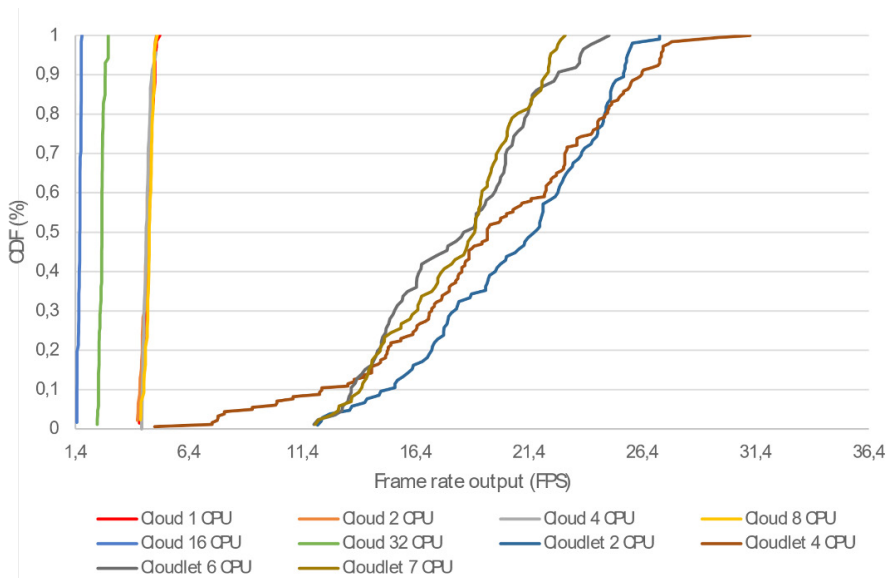


Figura 8 – Comparação das taxas de FPS - *FaceSwap*

Curiosamente, foi possível constatar que nos testes realizados com servidores de *cloud* com maior capacidade computacional, como nos sistemas de 16 e 32 vCPU, se assistiu a um aumento considerável no tempo de resposta da aplicação. Considerando que a capacidade de processamento foi aumentada, tal reflexo poderá estar associado a uma maior distância entre o cliente no *smartphone* e o servidor da *cloud* utilizado para a prestação do serviço alojamento com as características referidas. Repare-se, porém, que para os servidores *FaceSwap* de *cloud* com 16 vCPUs se obtiveram valores médios do tempo resposta em torno dos 600 ms, e que o aumento do número de vCPUs para 32 permitiu reduzir de forma muito significativa esse valor para cerca de 360 ms.

O tempo utilizado para o reconhecimento de rosto do servidor *cloudlet* é notoriamente inferior. A taxa de *frames* apresenta valores medianos de 19 a 22 FPS no *cloudlet* e de 2 a 5 FPS na *cloud*, como ilustra a Figura 8.

Nos testes realizados foi possível constatar que o uso da *cloudlet* garante uma taxa de FPS pelo menos 4 vezes maior do que a *cloud*, satisfazendo desta forma os requisitos de experiência do utilizador desta aplicação.

Da análise dos resultados obtidos foi possível constatar que estes estão em consonância com os obtidos nos testes realizados por J. Wang (Wang, 2016). Adicionalmente, foi efetuada uma comparação com outros trabalhos utilizando a aplicação *FaceSwap* (Ha et al., 2013) (Hu et al., 2016) (Chen et al., 2017), e nestes, os resultados obtidos nos testes realizados apresentam menor tempo de resposta nas soluções *cloudlet*.

## 5. Conclusões

Este artigo apresenta um estudo de análise e comparação das tecnologias *Mobile Edge Computing* (MEC) e *Mobile Cloud Computing* (MCC) relativamente ao tempo de resposta de aplicações com fortes requisitos de processamento em tempo real em dispositivos móveis. Com este estudo pode concluir-se que o MEC pode assumir um papel preponderante ao complementar o funcionamento das soluções assentes na *cloud* (MCC).

Em aplicações com requisitos de processamento intensivo em tempo real, a parte sensível ao atraso do aplicativo pode ser executada no servidor MEC, enquanto a parte intensiva, mais tolerante a atraso do aplicativo pode ser executada no servidor na *cloud*. O MEC permite que os dispositivos móveis conectados executem aplicativos de processamento intensivos e de tempo real na periferia da rede, libertando os terminais dos clientes utilizadores dessas tarefas. Para além de que, a maior proximidade entre servidor e dispositivo cliente permite reduzir significativamente o tempo de resposta das aplicações e, deste modo, melhorar a experiência do utilizador.

Este artigo apresenta os resultados de comparação de implementações do MEC com o MCC recorrendo a duas aplicações com requisitos de processamento em tempo real, o *Fluid* e o *FaceSwap*.

Dos resultados obtidos, mostra-se que o MEC com a implementação via *cloudlet* permite melhorar a qualidade e a experiência do utilizador. Os valores dos tempos de resposta medidos na aplicação *Fluid* fixam-se sempre abaixo de 100 ms, sendo aproximadamente quatro vezes inferiores aos valores apresentados utilizando a *cloud*. Além disso, pela utilização da *cloudlet*, o movimento das partículas é mais fluido e não se degrada facilmente como na *cloud*.

Os valores medidos na utilização do *FaceSwap* apresentam um tempo de resposta cerca de cinco vezes inferior, e sempre abaixo dos 100ms, quando comparada com os valores de uma implementação de *cloud*. Relativamente à taxa de FPS de imagens, o *cloudlet* apresenta valores quatro vezes mais elevados do que na *cloud*, o que evidencia a melhoria da qualidade de experiência do utilizador.

Em suma, quando comparado aos servidores remotos na *cloud*, verifica-se que, nas aplicações com as características seleccionadas, o MEC permite menor tempo de resposta e uma maior taxa de frames do que o MCC. Os resultados permitem concluir que a arquitetura MEC proporciona um melhor desempenho e melhor experiência ao utilizador, quando comparada com a *cloud*.

## Referências

- Aazam, M., & Huh, E.N. (2014). Fog computing and smart gateway-based communication for cloud of things. In *2014 International Conference on Future Internet of Things and Cloud*, (pp. 464–470). doi: 10.1109/FiCloud.2014.83.
- Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., & Buyya, R. (2013). Cloud-based augmentation for mobile devices: motivation, taxonomies, and open challenges. *IEEE Communications Surveys & Tutorials*, 16(1), 337–368. doi: 10.1109/SURV.2013.070813.00285

- Achanta, V., Sureshababu, N., Thomas, V., Sahitya, M., & Rao, S. (2012). Cloudlet-based multilingual dictionaries. In *Third International Conference on Services in Emerging Markets*, (pp. 30–36). doi: 10.1109/ICSEM.2012.12
- Ai, Y., Peng, M., & Zhang, K. (2017). Edge cloud computing technologies for internet of things: A primer. doi: 10.1016/j.dcan.2017.07.001
- Araújo, V. M., Cota, M. P. (2016). Software como um Serviço: uma visão holística. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (19), 145–157. doi: 10.17013/risti.19.145–157
- Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., & Stoica, I. (2009). Above the clouds: A berkeley view of cloud computing. Berkeley: Dept Electrical Eng and Computer Sciences, University of California.
- Beck, M. T., Werner, M., Feld, S., & Schimper, S. (2014). Mobile edge computing: A taxonomy. In *Proceedings of the Sixth International Conference on Advances in Future Internet*, (pp. 48-55).
- Blockstrand, M., Holm, T., Kling, L.-O., Kog, R., & Wallin, B. (2011). *More than 50 billion connected devices – taking connected devices to mass market and profitability*. Retrieved from: [https://www.akos-rs.si/files/Telekomunikacije/Digitalna\\_agenda/Internetni\\_protokol\\_Ipv6/More-than-50-billion-connected-devices.pdf](https://www.akos-rs.si/files/Telekomunikacije/Digitalna_agenda/Internetni_protokol_Ipv6/More-than-50-billion-connected-devices.pdf)
- Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog computing: A platform for internet of things and analytics. In *Big data and internet of things: A roadmap for smart environments*, (pp. 169-186). Cham: Springer International Publishing. doi: 10.1007/978-3-319-05029-4\_7.
- Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. In *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, (pp. 13-16). ACM. doi: 10.1145/2342509.2342513
- Chen, Z., Hu, W., Wang, J., Zhao, S., Amos, B., Wu, G., & Satyanarayanan, M. (2017). An Empirical Study of Latency in an Emerging Class of Edge Computing Applications for Wearable Cognitive Assistance. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, (pp. 14:1–14:14). ACM. doi: 10.1145/3132211.3134458
- Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless communications and mobile computing*, 13(18), 1587-1611. doi: 10.1002/wcm.1203
- Elijah, G. (2009). Cloudlet-based Edge Computing. Retrieved from: <http://elijah.cs.cmu.edu/>.
- ETSI. (2014). *Mobile-Edge Computing*. Introductory Technical White Paper. [https://portal.etsi.org/Portals/o/TBpages/MEC/Docs/Mobile-edge\\_Computing\\_-\\_Introductory\\_Technical\\_White\\_Paper\\_V1%2018-09-14.pdf](https://portal.etsi.org/Portals/o/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf)
- ETSI. (2015). *Mobile Edge Computing (MEC) Service Scenarios V1.1.1*.

- Fortes, N., Pereira, J. H., & Costa, J. F. da. (2016). A adoção de serviços cloud computing pelas empresas portuguesas: O papel dos esforços de marketing. *RISTI - Revista Ibérica de Sistemas e Tecnologias de Informação*, (18), 33-48. DOI: <https://dx.doi.org/10.17013/risti.18.33-48>
- GSMA Intelligence, (2014). *Understanding 5G: Perspectives on future technological advancements in mobile*. White paper, 1-26.
- Ha, K., Pillai, P., Lewis, G., Simanta, S., Clinch, S., Davies, N., & Satyanarayanan, M. (2013). The impact of mobile multimedia applications on data center consolidation. In *2013 IEEE international conference on cloud engineering (IC2E)*, (pp. 166–176). doi: 10.1109/IC2E.2013.17
- Hu, W., Gao, Y., Ha, K., Wang, J., Amos, B., Chen, Z., Pillai, P., & Satyanarayanan, M. (2016). Quantifying the Impact of Edge Computing on Mobile Applications. In *Proceedings of the 7th ACM SIGOPS Asia-Pacific Workshop on Systems*, (pp. 5:1–5:8). ACM. doi: 10.1145/2967360.2967369
- Khan, K., Wang, Q., Luo, C., Wang, X., & Grecos, C. (2014). Comparative study of internet cloud and cloudlet over wireless mesh networks for real-time applications. In *Real-Time Image and Video Processing 2014*, (pp. 91390K). doi: 10.1117/12.2052474
- Klas, G. (2016). Fog computing and mobile edge cloud gain momentum open fog consortium, etsi mec and cloudlets. In *SoftNet 2016 Conference*, August 21, 2016, Rome.
- Koukoumidis, E., Lymberopoulos, D., Strauss, K., Liu, J., & Burger, D. (2011). Pocket cloudlets. In *ACM SIGARCH Computer Architecture News*, (pp. 71–184). doi: 10.1145/2248487.1950387
- Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322-2358. doi: 10.1109/COMST.2017.2745201
- Patel, M., Naughton, B., Chan, C., Sprecher, N., Abeta, S., & Neal, A. (2014). Mobile-edge computing introductory technical white paper. White paper, mobile-edge computing (MEC) industry initiative, (pp. 1089-7801).
- Qing, W., Zheng, H., Ming, W., & Haifeng, L. (2013). Cactse: Cloudlet aided cooperative terminals service environment for mobile proximity Content delivery. *China Communications*, 10(6), 47–59. doi: 10.1109/CC.2013.6549258.
- Satyanarayanan, M. (2013). Cloudlets: At the Leading Edge of Cloud-mobile Convergence. In *Proceedings of the 9th International ACM Sigsoft Conference on Quality of Software Architectures (QoSA '13)*, (pp. 1–2). doi: 10.1145/2465478.2465494
- Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The Case for VM-Based Cloudlets in *Mobile Computing IEEE Pervasive Computing*, 8(4), 14–23. doi: 10.1109/MPRV.2009.82

- Shahzadi, S., Iqbal, M., Dagiuklas, T., & Qayyum, Z. (2017). Multi-access edge computing: open issues, challenges and future perspectives. *Journal of Cloud Computing*, 6(1), 30. doi: 10.1186/s13677-017-0097-9
- Shenoy, K., Bhokare, P., & Pai, U. (2015). Fog computing future of cloud computing, *International Journal of Science and Research*, 4(6), 55-56.
- Simanta, S., Ha, K., Lewis, G., Morris, E., & Satyanarayanan, M. (2012). A Reference Architecture for Mobile Code Offload in Hostile Environments. In *Joint Working IEEE/IFIP Conference on Software Architecture and European Conference on Software Architecture*, Helsinki, 2012, (pp. 282-286). doi: 10.1109/WICSA-ECSA.212.46.
- Skarpness, S. (2017). Keynote: Beyond the Cloud: Edge Computing. Recuperado de: <https://www.youtube.com/watch?v=mCDXnls6pQk>
- Solenthaler, B., & Pajarola, R. (2009). Predictive-corrective Incompressible SPH. *ACM SIGGRAPH 2009 papers*, 28(3), 40:1-40:6. doi: 10.1145/1576246.1531346
- Soyata, T., Muraleedharan, R., Funai, C., Kwon, M., & Heinzelman, W. (2012). Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In *2012, IEEE symposium on computers and communications (ISCC)*, (pp.000059-000066). doi: 10.1109/ISCC.2012.6249269
- Suryawanshi, R., & Mandlik, G. (2015). Focusing on mobile users at edge and internet of things using fog computing. *Int. J Sci. Eng. Tech. Res*, 4(17), 3225-3231.
- Verbelen, T., Simoens, P., De Turck, F., & Dhoedt, B. (2012). Cloudlets: Bringing the cloud to the mobile user. In *Proceedings of the third ACM workshop on Mobile cloud computing and services*, (pp. 29-36). ACM. doi: 10.1145/2307849.2307858
- Wang, J. (2016). FaceSwap (Internet) Retrieved from: <https://cmusatyalab.github.io/faceswap/dev-guide/>
- Zhang, Q., Cheng, L., & Boutaba, R. (2010). Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1), 7-18. doi: 10.1007/s13174-010-0007-6