

# Distribuição de Conteúdo Sob Demanda Através da Alocação de Microserviços Dinâmicos na Borda e Núcleo da Rede

Derian Alencar<sup>1</sup>, Denis Rosário<sup>1</sup>, Eduardo Cerqueira<sup>1</sup>,  
Cristiano Both<sup>2</sup> e Rodolfo Antunes<sup>2</sup>

<sup>1</sup> Universidade Federal do Pará (UFPA), Belém, PA – Brasil

<sup>2</sup> Universidade do Vale do Rio dos Sinos (UNISINOS), São Leopoldo, RS – Brasil

derian.alencar@itec.ufpa.br, {denis,cerqueira}@ufpa.br,  
{cbboth,rsantunes}@unisinis.br

**Abstract.** *Video-on-Demand (VoD) content distribution is growing on popularity and will be the dominant application on the Internet. To manage traffic and provide QoE guarantees to VoD, microservices become an ideal model to explore the deployment of caching services at different levels of a Fog computing architecture. The use of microservices for video content distribution is key to enable the dynamic allocation of resources on fog nodes according to client demands. This paper presents Fog4MS, a mechanism for dynamic allocation of VoD caching microservices for Fog computing environments. The mechanism considers delay, microservice migration time, and utilization level of the fog node to determine the best location for caches considering a multi-layer fog computing architecture. Experiments in a simulated environment demonstrate the efficiency of the proposal in comparison of existing mechanisms when considering cost, time to migrate, fairness index, and QoE.*

**Resumo.** *A distribuição de conteúdo de vídeo sob Demanda (VoD) está se popularizando e será o principal tipo de aplicação utilizada na Internet. Para acomodar o tráfego de VoD e prover disseminação com Qualidade de Experiência (QoE), microserviços surgem como o modelo ideal para explorar a implantação de serviços de cache em diferentes níveis de uma arquitetura de computação em névoa. A utilização de microserviços na distribuição de conteúdo de vídeo é estratégica, pois permite alocar os recursos em nós da névoa de forma dinâmica e customizada com a demanda dos clientes. Este artigo apresenta Fog4MS, um mecanismo para alocação dinâmica de microserviços para cache de VoD em ambientes de computação em névoa. O mecanismo Fog4MS considera o atraso, tempo de migração do conteúdo e taxa de utilização do nó da névoa para calcular a melhor localização das caches considerando a arquitetura de computação em névoa multiníveis. Experimentos realizados em simulação mostram a eficiência da proposta comparada a mecanismos existentes quando considerando custo, tempo de migração do conteúdo, índice de justiça e QoE.*

## 1. Introdução

A distribuição de conteúdo de vídeo é o principal tipo de aplicação atualmente utilizada na Internet. Projeções mostram que, em 2022, aproximadamente 82% do tráfego da Internet será dominado pela distribuição de conteúdo multimídia, bem como serviços de vídeo sob demanda (*Video on Demand* - VoD), como YouTube, Netflix, dobrarão sua

utilização [Cisco 2019]. Nesse contexto, é necessário fornecer entrega de conteúdo de VoD com suporte à Qualidade da Experiência (QoE) [Araújo et al. 2019]. Para atender aos requisitos de QoE, as infraestruturas desses serviços devem oferecer mecanismos de distribuição de conteúdo capazes de suportar esse substancial crescimento.

Os principais provedores de conteúdo de VoD utilizam arquiteturas baseadas em redes de distribuição de conteúdo (*Content Delivery Networks* - CDNs). Atualmente, esses provedores já utilizam sistemas baseados em ambientes de nuvem para o armazenamento e distribuição eficiente de VoD [Salahuddin et al. 2018]. Tais arquiteturas atendem às demandas atuais por conteúdo de VoD na Internet, porém elas não exploram todo o potencial presente em arquiteturas de rede recentes, como por exemplo computação em névoa [Mao et al. 2017]. Além disso, tais arquiteturas apresentam recursos computacionais e de armazenamento que podem ser explorados para ampliar a capacidade disponível de distribuição de VoD dinamicamente, *i.e.*, de acordo com a demanda do usuário.

Microserviços são um modelo para o desenvolvimento de sistemas que vem ganhando destaque em função de sua flexibilidade [Tian et al. 2018]. Esse modelo utiliza menos recursos computacionais quando comparados ao modelo monolítico com máquinas virtuais, pois são baseados nos recentes avanços em infraestruturas de contêineres. Microserviços apresentam uma maior modularidade, conseqüentemente, reduzem a complexidade para implantação e elasticidade de aplicações. Em suma, o modelo de microserviços é ideal para explorar a implantação de mecanismos para distribuição de conteúdo de VoD, tal como serviços de *cache* em uma arquitetura de computação em névoa com múltiplos níveis, onde os clientes apresentam uma demanda dinâmica por um determinado conteúdo multimídia.

Nesse contexto, mecanismos de alocação de microserviços na computação em névoa são fundamentais para manter a QoE da aplicação na distribuição de conteúdo de VoD, o qual é formado por duas etapas: (i) decisão e alocação do microserviço e (ii) migração do conteúdo. A etapa de decisão é responsável por encontrar o melhor nó para implantação do microserviço em uma arquitetura de computação em névoa multiníveis, o qual deve prover menor atraso. Por sua vez, a migração refere-se à transferência do conteúdo da nuvem para o nó da névoa e redirecionamento das requisições de conteúdo, a qual deve ter o menor tempo de migração para não interferir na QoE. Tal microserviço permanece implantado e consumindo os recursos da computação em névoa apenas enquanto for justificável em função da demanda pelo conteúdo.

A literatura mostra que, na maioria dos casos, mecanismos de *cache* e taxa de bits adaptativa (*Adaptive BitRate* - ABR) não são integrados como microserviços em computação em névoa para distribuição de conteúdo de VoD, reduzindo assim a possibilidade de ganhos da arquitetura como um todo [Menchaca-Mendez et al. 2018]. Portanto, torna-se necessário explorar a combinação de diferentes mecanismos para otimizar a distribuição de VoD nas diferentes camadas da arquitetura. Mais especificamente, as requisições dos clientes devem ser gerenciadas dinamicamente para que sejam atendidas por nós da computação em névoa capazes de suprir as demandas de recurso de acordo com o requisito de QoE da aplicação [Jalali et al. 2016, Menchaca-Mendez et al. 2018].

A literatura também apresenta trabalhos que tratam de diferentes aspectos relacionados com a distribuição de conteúdo de VoD em arquiteturas baseadas em microserviços [Bhardwaj et al. 2019]. Especificamente, observam-se estudos que tratam de arquiteturas baseadas em computação em névoa para a distribuição de conteúdo de VoD. Entretanto, tais trabalhos apresentam limitações, em particular no que diz respeito ao me-

canismo de alocação de microserviços [Masip-Bruin et al. 2016, Rosário et al. 2018]. Outra limitação de trabalhos existentes na área de alocação de microserviços é que não realizam a migração do conteúdo com base nas requisições a conteúdos por parte dos usuários [Wang et al. 2015]. Serviços de distribuição de VoD apresentam uma demanda de recursos particularmente alta e, portanto, arquiteturas baseadas em computação em névoa e microserviços devem apresentar os mecanismos necessários para otimizar a disseminação de VoD [He et al. 2017]. Nesse sentido, alguns trabalhos também focam na alocação dinâmica de recursos com o objetivo de adaptar a capacidade do sistema ao estado atual da demanda dos clientes [Bhardwaj et al. 2019].

Para atender essas questões, o presente artigo apresenta um mecanismo de alocação dinâmica de microserviços de *cache* considerando uma arquitetura de computação em névoa multiníveis para distribuição de conteúdo de VoD com suporte QoE, denominado de Fog4MS. Para a etapa de decisão e alocação de microserviços, o mecanismo Fog4MS considera o atraso, tempo de migração do conteúdo e taxa de utilização de recursos do nó da névoa para calcular a melhor localização do microserviço de *cache* em uma arquitetura de computação em névoa multiníveis. Para ajustar o grau de importância de cada critério na tomada de decisão foi utilizado o método matemático *Analytic Hierarchy Process* (AHP). Simulações mostram a eficiência em termos de custo, tempo de migração do conteúdo, índice de justiça e QoE do mecanismo Fog4MS, comparado a mecanismos existentes na literatura. Por exemplo, os resultados de simulação mostraram que Fog4MS é capaz de reduzir em cerca de 25% o custo e diminuir o tempo de migração do conteúdo em aproximadamente 50% comparado com mecanismos existentes. Mais especificamente, as contribuições do presente artigo são as seguintes: (i) proposta de um mecanismo de alocação dinâmica de microserviços de *cache* de VoD, o qual é baseado nos conceitos de microserviços e arquitetura em névoa com multicamadas; (ii) introdução da decisão de alocação de microserviços na computação em névoa considerando atraso, tempo de migração do conteúdo e taxa de utilização de recursos do nó da névoa; e (iii) análise de desempenho do mecanismo Fog4MS conduzida através de experimentos em simulação baseado na arquitetura em névoa com multicamadas.

O restante do artigo está organizado conforme descrito a seguir. A Seção 2 apresenta uma visão geral sobre trabalhos que exploram propostas similares à aqui apresentada. A Seção 3 descreve o mecanismo Fog4MS para a distribuição de conteúdos de VoD. A Seção 4 explora o modelo de simulação desenvolvido para avaliar o mecanismo proposto, bem como os resultados obtidos. Por fim, a Seção 5 conclui o artigo com considerações finais e direções para trabalhos futuros.

## 2. Trabalhos Relacionados

Vários trabalhos propõem mecanismos para o redirecionamento de requisições baseado na taxa de chegada das mesmas. Tang, Tay e Wen exploram o redirecionamento de requisições para vários servidores buscando minimizar uma função de custo baseada em tempo de resposta, custos computacionais e capacidade da rede [Tang et al. 2014]. Siavoshani, Pourmiri e Shariatpanahi propõem o balanceamento de recursos armazenamento e comunicação utilizando o redirecionamento de requisições baseado em limitações no espaço de *cache* e proximidade de clientes ao servidor [Siavoshani et al. 2017]. Chunlin, Chuanli e Youlong buscam minimizar o tempo de execução e consumo de energia de serviços de distribuição de vídeo através do uso de parâmetros de QoS para a seleção de serviços [Chunlin et al. 2019]. Esses três trabalhos limitam-se à otimização de Qualidade de Serviço (QoS), os quais nem sempre afetarão diretamente a QoE do serviço de vídeo.

Xiao *et al.* modelam o redirecionamento dinâmico de requisições distribuídas geograficamente na forma de um problema de otimização estocástico, buscando minimizar os custos de aluguel de recursos de nuvem e melhorar a QoE de serviços de distribuição de vídeo [Xiao et al. 2015]. Zhang, Jiang e Xi propõem um mecanismo de posicionamento de conteúdo e encaminhamento de requisições para conteúdo de vídeo baseado em um processo de decisão de Markov, buscando maximizar os lucros do provedor de serviço [Zhang et al. 2016]. Ambos os trabalhos exploram a correlação de métricas de QoS com a QoE para distribuição de vídeo. Entretanto, a relação entre QoS e QoE é limitada, pois ambos possuem o foco na utilização de recursos ou custo de alocação na tomada de decisão, conseqüentemente, tal relação traz ganhos limitados para a QoE.

A literatura também apresenta estudos sobre mecanismos para a alocação de recursos em ambientes de névoa, os quais são diretamente aplicáveis ao contexto de micros serviços. Aazam e Huh introduzem um modelo para o gerenciamento e alocação dinâmica de recursos em um ambiente de computação em névoa [Aazam and Huh 2015]. O modelo emprega parâmetros (*e.g.*, requisitos de armazenamento e rede) relativos à aplicação que será executada para gerenciar solicitações de alocação. Posteriormente, Aazam *et al.* estendem o modelo, incluindo diversos aspectos sobre o histórico de utilização de recursos pelos clientes [Aazam et al. 2016]. Os autores buscam favorecer alocações para clientes que fazem “bom uso” dos recursos solicitados, isto é, não solicitam recursos e os mantém desocupados. Ni *et al.* exploram um modelo de gerenciamento de recursos baseado no conceito de *Priced Timed Petry Nets* (PTPN), o qual permite implementar a predição do tempo e custo da execução de serviços no ambiente de computação em névoa [Ni et al. 2017]. Os autores utilizam o modelo proposto para implementar um mecanismo de alocação dinâmica de recursos em computação em névoa que leva em conta o tempo de utilização e a confiabilidade dos recursos disponíveis. Esses trabalhos exploram exclusivamente o problema da alocação de recursos em ambientes de computação em névoa, mas não exploram as características específicas da distribuição de VoD, em particular sua sensibilidade a um alto atraso e um longo tempo de bufferização, que ocasiona travamentos e demora na reprodução do conteúdo, resultando em um baixo QoE.

**Tabela 1. Resumo dos Mecanismos de Distribuição de VoD**

Trabalhos	Características						
	QoE	QoS	Cache	Requisição	Computação em névoa	DFA	DRA
[Tang et al. 2014]		✓		✓			
[Siavoshani et al. 2017]		✓		✓			
[Chunlin et al. 2019]		✓	✓	✓			
[Xiao et al. 2015]		✓		✓			
[Zhang et al. 2016]	✓		✓	✓			
[Aazam and Huh 2015]					✓		✓
[Aazam et al. 2016]	✓				✓		✓
[Ni et al. 2017]		✓				✓	✓

Com base na análise dos trabalhos relacionados é possível concluir que serviços de VoD implantados em ambientes de computação em névoa com múltiplos níveis exploram a unificação de recursos de diferentes dispositivos de rede a fim de ampliar sua QoE. Portanto, torna-se essencial direcionar as requisições para nós de computação em névoa adequados, assim evitando que usuários abandonem o serviço devido a uma baixa QoE. Dessa forma a alocação dinâmica de micros serviços proposta pelo Fog4MS visa prover distribuição de VoD com suporte e priorização do QoE, e assim a tomada de decisão

utiliza o atraso e o tempo de migração como as principais métricas para a escolha da alocação do microserviço, o que acarreta em ganhos perceptíveis para a QoE percebida pelo usuário. Além disso, observa-se que trabalhos que exploram a alocação dinâmica de recursos na computação em névoa não exploram as peculiaridades de serviços focados em distribuição de vídeo, por exemplo através do modelo de microserviços. Por fim, a Tabela 1 resume as principais características de trabalhos analisados destinados a fornecer distribuição de conteúdo em ambientes baseados em computação em névoa.

### **3. Mecanismo de alocação dinâmica de microserviços de *cache* de VoD em uma arquitetura de computação em névoa multiníveis (Fog4MS)**

Essa seção descreve a proposta do mecanismo Fog4MS para alocação dinâmica de microserviços baseado em atraso, tempo de migração de conteúdo e taxa de utilização de recursos na névoa para a etapa de decisão. Primeiro, é apresentada uma visão geral da arquitetura de computação em névoa multiníveis. Em seguida, é introduzida a tomada de decisão utilizada para a alocação de microserviços e recursos na infraestrutura disponível.

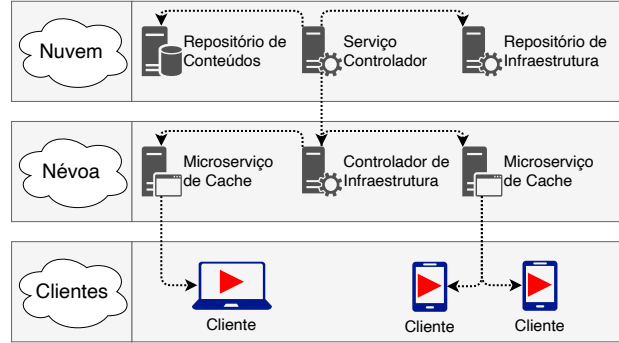
#### **3.1. Infraestrutura de computação em névoa multiníveis**

O mecanismo de alocação dinâmica de microserviço proposto considera uma arquitetura de computação em névoa multiníveis, onde cada nível possui um conjunto de funcionalidades para posicionar e servir os conteúdos o mais próximo possível dos clientes, também buscando reduzir o consumo de recursos necessários. A arquitetura tem seus principais componentes ilustrados na Figura 1. De forma geral, ela possibilita a distribuição do conteúdo de VoD para cada solicitação do cliente, bem como orquestra e controla os nós da névoa para distribuir de forma adaptativa e proativa o conteúdo de VoD.

Considera-se que os componentes de controle do mecanismo e todos os conteúdos disponíveis são mantidos em um ambiente de nuvem, o qual é representado na arquitetura pela camada homônima. O *Repositório de Conteúdo* presente nessa camada mantém a cópia original de todos os vídeos disponíveis. Dependendo das requisições dos usuários, esses conteúdos podem passar por um processo de transcodificação antes de serem transmitidos para outras camadas da arquitetura ou para os clientes. Nesse caso, presume-se que a camada de nuvem apresenta recursos computacionais suficientes para realizar tal processo sem interferir na distribuição e na qualidade de experiência do usuário. Também é possível que a camada de nuvem contenha múltiplos *datacenters*, de modo que as cópias dos conteúdos sejam movidas e armazenadas geograficamente para regiões mais próximas dos clientes. Por sua vez, o *Serviço Controlador* é responsável pelo controle de todos os componentes da arquitetura, incluindo as etapas de alocação de microserviços de *cache* e a migração de conteúdo para eles. O controlador utiliza as informações contidas no *Repositório de Infraestrutura* para selecionar qual nó da computação em névoa será alocado o microserviço de *cache*.

A camada de névoa é composta por todas as instâncias disponíveis de microserviços de *cache* utilizados para acelerar a distribuição dos conteúdos. Considera-se que a camada de névoa seja composta por várias infraestruturas computacionais com características heterogêneas, onde cada nó da névoa é representado por  $f_i$  e  $i \in [1, n]$  identificador de cada névoa. O serviço de distribuição de conteúdo de VoD tem acesso a estas infraestruturas e pode alocar uma porção de seus recursos para instanciar um microserviço para *cache* e entrega de conteúdo ativos. Em função disso, a camada de computação em névoa possui uma ou mais instâncias do *Microserviço de Cache*, o qual é responsável por

entregar o conteúdo solicitado aos clientes. As instâncias de microserviços são controladas em cada infraestrutura de computação em névoa por um *Controlador de Infraestrutura*, o qual é gerenciado pelo provedor da computação em névoa. Por fim, a camada *Clientes* engloba todos os clientes do serviço de distribuição, que realizam as requisições para os conteúdos disponíveis.



**Figura 1. Infraestrutura para distribuição de vídeo através de microserviços**

O mecanismo de alocação de microserviços que estará instanciado no *serviço controlador* deve utilizar informações da demanda dos clientes, da disponibilidade de recursos nas infraestruturas de borda disponíveis, bem como a informação sobre a latência das conexões com clientes para as etapas de decisão e migração de conteúdo. A seguir é descrito o mecanismo Fog4MS para alocação dinâmica de microserviços

### 3.2. Mecanismo Fog4MS

O mecanismo Fog4MS encontra-se no módulo *Serviço Controlador*. Desta forma, o mecanismo é responsável pelas etapas de decisão (*i.e.*, posicionamento de microserviços na computação em um determinado nó da névoa) e migração de conteúdo (*i.e.*, transferência do conteúdo e o direcionamento de requisições para tal nó). Para a etapa de decisão, o mecanismo Fog4MS considera o atraso, tempo de migração do conteúdo e taxa de utilização de recursos na névoa, uma vez que tais métricas impactam na QoE para distribuição de conteúdo VoD.

A máxima vazão  $TCP_{MT}$  possível é calculada pela Eq. 1 [Padhye et al. 2000], na qual  $W_s$  é o tamanho da janela do TCP e o  $R_{TT}$  é o tempo de transmissão do pacote. Dessa forma, o tempo de migração  $t_m$  do conteúdo da nuvem para um determinado nó da névoa  $f_i$  é calculado com base na Eq. 2, definindo  $S$  como o tamanho do conteúdo a ser alocado. Dada uma modulação e codificação, a vazão máxima  $TCP_{MT}$  sempre será igual ou menor que a largura de banda do canal  $B_W$ , conforme a Eq. 3. O tempo de migração  $t_m$  mais baixo significa que o nó da névoa  $f_i$  está próximo ao usuário  $u$  que está requisitando o conteúdo de VoD, resultando em um melhor desempenho para disseminação.

$$TCP_{MT} = \frac{W_s}{R_{TT}} \quad (1)$$

$$t_m = \frac{S}{TCP_{MT}} \quad (2)$$

na qual,

$$TCP_{MT} \leq B_W \quad (3)$$

A métrica de taxa de utilização  $U$  de recursos de memória utilizados para alocação dos micros serviços de um determinado nó de computação em névoa é dada pela razão entre a quantidade de recursos alocados  $M_a$  e a quantidade total de recursos  $M_{max}$  de tal nó da névoa  $f_i$ , a qual é calculada conforme a Eq. 4. Essa métrica indica que quanto mais recursos disponíveis mais prioritário é o nó da névoa  $f_i$ , com o objetivo de prover balanceamento de carga na alocação de micros serviços.

$$U_i = \frac{M_{a,i}}{M_{max,i}} \quad (4)$$

O mecanismo Fog4MS calcula o custo  $S_i$  para alocação do micros serviço de *cache* na arquitetura baseado na Eq. 5. O cálculo considera diferentes pesos  $p_j$  para cada métrica  $M_j$ , *i.e.*, atraso, tempo de migração do conteúdo e taxa de utilização de recursos. O peso representa o grau de importância de uma métrica em relação as demais.

$$S_i = \sum_{j=1}^n p_j \times M_j \quad (5)$$

O mecanismo Fog4MS considera o método matemático AHP para estimar a melhor resposta de acordo com a importância de cada parâmetro em relação a outro. AHP provê uma técnica estruturada para a tomada de decisão de problemas com múltiplos critérios envolvidos, através da decomposição de um problema complexo em uma hierarquia de subproblemas mais simples, combinando fatores qualitativos e quantitativos para a análise, permitindo que o sistema encontre uma solução ideal quando houver vários critérios considerados no processo de entrega. Especificamente, AHP considera a comparação em pares entre os valores numéricos de cada parâmetro e seus graus de importância relativos, de forma a ajustar seus pesos em tempo de execução. Dessa forma, um valor numérico representa a comparação entre pares, que não devem se contradizer, por exemplo, se uma determinada métrica  $i$  for duas vezes mais importante que uma outra métrica  $j$ , então  $j$  é 1/2 vez tão importante quanto  $i$ . Foram definidos três níveis de importância para comparar cada par de parâmetros conforme mostrado na Tabela 2. Como resultado, quanto maior o peso, mais importante é o critério correspondente.

**Tabela 2. Escala de importância relativa para o AHP**

$C_{i,j}$	Grau de Importância
1	Duas métricas, <i>i.e.</i> , $i$ e $j$ , tem a mesma importância
2	Uma métrica $i$ é mais importante que outra métrica $j$
3	Uma métrica $i$ é muito mais importante que outra métrica $j$
1/2	Uma métrica $i$ é menos importante que outra métrica $j$
1/3	Uma métrica $i$ é muito menos importante que outra métrica $j$

O mecanismo Fog4MS constrói uma matriz para cada nó da névoa  $f_i$  a fim de comparar todos os pares de critérios. A matriz de comparação  $C_{i,j}$  da Eq. 6 representa o quão importante é um critério  $i$  comparado a outro,  $j$ , da matriz. Define-se  $D$  como atraso entre o usuário  $u$  e um determinado nó da névoa  $f_i$ ,  $t_m$  como o tempo de migração do conteúdo da nuvem para um dado nó da névoa  $f_i$ , e  $U$  como a taxa de utilização de recursos para alocação dos micros serviços em um dado nó da névoa  $f_i$ . Vale a pena realçar que se um critério é considerado duas vezes mais importante que outro, então o outro é

1/2 tão relevante quanto o primeiro, devido à comparação invertida. A diagonal principal da matriz sempre deve conter o valor 1, pois compara-se uma métrica com ela mesma.

$$C_{i,j} = t_m \begin{matrix} & D & t_m & U \\ D & \begin{pmatrix} 1 & 2 & 3 \\ 1/2 & 1 & 3 \\ 1/3 & 1/3 & 1 \end{pmatrix} \end{matrix} \quad (6)$$

Posteriormente, é necessário calcular o vetor próprio da matriz ( $C_{i,j}$ ), através da média da divisão de cada elemento pela soma de sua coluna para cada coluna da matriz. Por exemplo, pode-se obter o vetor  $P = [0.524 \ 0.334 \ 0.142]$ , significando que o atraso  $D$  terá um peso de 52%, o tempo de migração  $t_m$  terá peso de 33%, e a taxa de utilização  $U$  terá peso de 14% na tomada de decisão da AHP. O mecanismo Fog4MS realiza um produto entre o vetor  $P$  e um vetor que armazena os valores medidos  $M_j$ , obtendo o custo  $S_i$  para alocação do microserviço de *cache* na arquitetura de névoa multiníveis, onde os menores valores medidos  $M_j$  resultam em um menor custo  $S_i$ . Com base no custo  $S_i$  para alocação do microserviço em cada nó da névoa  $f_i$ , o mecanismo Fog4MS seleciona o nó da névoa com o menor custo  $S_i$ , uma vez que tal nó terá o menor tempo de migração do microserviço, proverá o menor atraso de comunicação e terá a menor taxa de utilização. Esses fatores possibilitam a distribuição de conteúdo de VoD com melhor QoE.

## 4. Avaliação de Desempenho

Essa seção apresenta a avaliação experimental do Fog4MS, incluindo o cenário, as métricas, a metodologia, e os resultados obtidos. O mecanismo foi avaliado através de um conjunto de experimentos baseados em um ambiente de simulação. Para isso, uma comparação foi feita a fim de avaliar o mecanismo de alocação proposto com outros mecanismos para a alocação do conteúdo. Na avaliação foi abordado, o tempo de migração do conteúdo, o tempo de *bufferização* do vídeo, o custo de alocação do microserviço na névoa e, por fim, o índice de justiça de cada método.

### 4.1. Metodologia de Simulação

O Fog4MS foi implementado no simulador de eventos discreto NS3 [Riley and Henderson 2010], disponível no GitHub<sup>1</sup>. Cada simulação foi executada 33 vezes com diferentes sementes de aleatoriedade e os resultados apresentam os valores com um intervalo de confiança de 95%. O cenário consiste em um número de requisições de microserviço variada (*i.e.*, 20, 40, 60, 80 e 100), modeladas de acordo com uma distribuição de Poisson. As requisições consistem em conteúdo de VoD, na qual os clientes podem escolher o vídeo a partir de um catálogo de 100 itens distintos e selecionar a qualidade pretendida do conteúdo. A seleção do vídeo e sua qualidade é dada pela distribuição de Zipf com  $\alpha$  igual a 0,7 de forma que a preferência por conteúdo seja melhor distribuída.

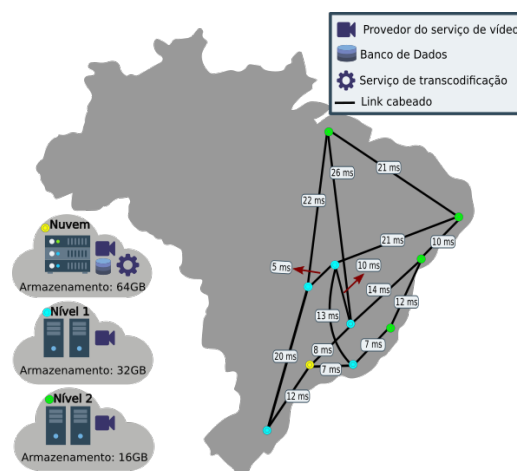
O cenário considera a topologia do projeto FIBRE<sup>2</sup> para atribuir os atrasos [Rosário et al. 2018], distribuir os pontos de névoa e de nuvem, conforme a Figura 2. Utiliza-se uma abordagem multicamada de névoa, havendo um nó de nuvem e nós de níveis 1 e 2 da névoa, os quais são distribuídos pelo cenário. O nó de nível 2 refere-se a um servidor local com menor capacidade de armazenamento, enquanto o nó de nível

<sup>1</sup><https://github.com/D3F3R4L/Fog4MS>

<sup>2</sup><https://www.fibre.org.br/>



1 representa um servidor regional com mais recursos. A nuvem representa um servidor com recursos suficiente para atender toda a demanda e onde todo o conteúdo do cenário está alocado.



**Figura 2. Cenário da topologia do projeto FIBRE**

Os clientes podem surgir em qualquer um dos locais, onde estão distribuídos os pontos de névoa, seguindo uma distribuição uniforme. A partir disso, é requisitado um conteúdo de VoD. Os vídeos são padronizados em duração de 30 minutos e foram codificados com taxa de bits conforme a Tabela 3<sup>3</sup>. A requisição é feita para a nuvem onde se encontra a Fog4MS, que seleciona o melhor ponto de névoa para alocar o microserviço, conforme a decisão da AHP, descrito na Seção 3, é importante ressaltar que o cenário possui tráfego de fundo entre os enlaces, onde até 60% da largura de banda pode estar sendo utilizada por outros serviços. Assim, a tomada de decisão do Fog4MS leva em consideração características variáveis da rede como o tráfego de fundo. O tempo total de simulação foi de 60 minutos, *i.e.*, suficiente para avaliar os algoritmos de disseminação.

**Tabela 3. Resolução do vídeo e sua respectiva taxa de bits**

	Parâmetros							
Resolução do vídeo	180p	360p	360p	540p	540p	720p	1080p	1080p
Taxa de bits (Kbps)	400	650	1000	1500	2250	3400	4700	6000

Em cenários com grande número de clientes, é importante levar em consideração o quanto a tomada de decisão do mecanismo Fog4MS é justa em termos de uso dos recursos de armazenamento de conteúdo. Para isso, considerou-se o índice *Jain Fairness F* [Jain et al. 1984] para expressar o quanto os mecanismos são igualitários na alocação de conteúdo de VoD na névoa. O índice é calculado pela Eq. 7, na qual  $x_i$  representa o uso do armazenamento da névoa  $f_i$ .

$$F = \frac{[\sum_{i=1}^n (x_i)]^2}{n \sum_{i=1}^n (x_i)^2} \quad (7)$$

<sup>3</sup><https://docs.microsoft.com/bs-latn-ba/azure/media-services/previous/media-services-mes-preset-h264-multiple-bitrate-1080p-audio-5.1>

Avaliou-se o tempo de migração do conteúdo da nuvem para a névoa, bem como a QoE através do tempo de *bufferização* do vídeo, que é o período antes do vídeo começar a ser reproduzido. Normalmente, inclui o tempo necessário para baixar a página HTML (ou arquivo de manifesto), carregar o *plug-in* do cliente de vídeo e reproduzir a parte inicial do vídeo. Por fim, considerou-se o custo para o uso dos serviços na névoa, baseado na calculadora AWS da Amazon<sup>4</sup>. Os valores obtidos e a configuração dos pontos de névoa e nuvem possuem a mesma configuração disponíveis para uso na Amazon. Nesse padrão as máquinas são divididas em níveis referentes ao poder computacional, conforme esses níveis foi adotado a distinção da multicamada para os pontos de névoa e nuvem onde a nuvem possui mais potência computacional. Dessa forma, os valores e a configuração adotada para os pontos de névoa e nuvem são mostrados na Tabela 4.

**Tabela 4. Custo monetário de CPU por hora na Amazon ECS**

Nós	CPU Cores	Memória (GB)	Armazenamento (GB)	Custo (USD)
Nuvem	16	64	64	1,224
Nível 1	8	32	32	0,612
Nível 2	4	16	16	0,306

Finalmente, o mecanismo Fog4MS foi comparado com outros mecanismos para a alocação do conteúdo. Mais especificamente, um método de decisão Gulosa que toma as decisões com base na latência; um método de decisão aleatório que escolhe qualquer um ponto da névoa randomicamente; e ao cenário padrão que aloca todo o conteúdo na nuvem, denotado como nuvem nos resultados. Os principais parâmetros usados na simulação estão sumarizados na Tabela 5.

**Tabela 5. Parâmetros da simulação**

Parâmetros	Valores
Número de microserviços	20, 40, 60, 80 e 100
Topologia	10 nós (9 névoas e 1 nuvem)
Popularidade do conteúdo	Zipf com $\alpha$ 0.7
Número de conteúdos	100
Versões do conteúdo	8
Tempo de Vídeo	30 min
Tempo de simulação	60 min
Janela TCP para migração	1024 Kbits
Capacidade dos links	100 Mbps

## 4.2. Resultados

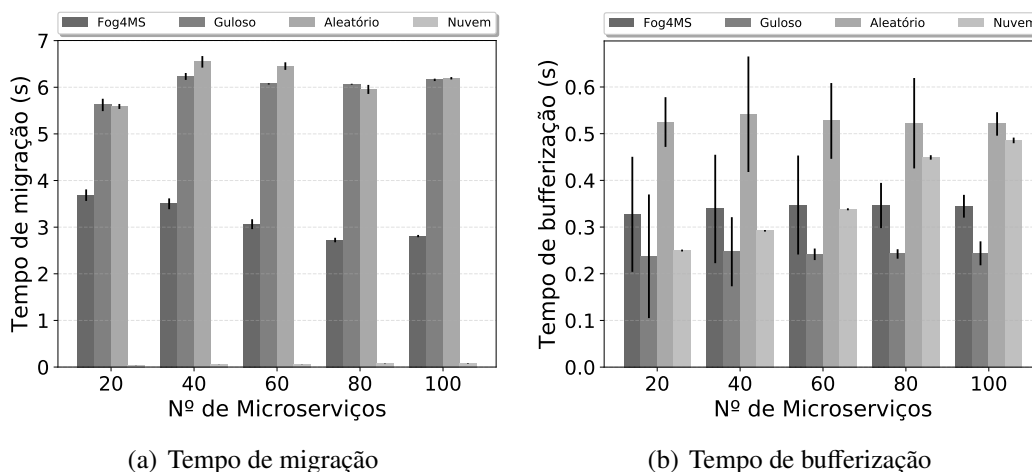
As Figuras 3(a) e 3(b) apresentam o tempo médio de migração do conteúdo e de *bufferização* do vídeo ao se alocar o microserviço na névoa, respectivamente. A *bufferização* do vídeo tem impacto direto na QoE percebida pelos usuários de VoD, pois quanto menor o tempo de *bufferização*, maior será a QoE do usuário e menor será a taxa de abandono do vídeo [Juluri et al. 2016]. Dessa forma, é imprescindível a observação de métricas como o tempo de migração do serviço e o tempo de *bufferização* para a análise de QoE.

A Figura 3(a) mostra que a nuvem possui um tempo de migração inexistente, pois o conteúdo já está armazenado, sendo apenas necessário instanciar o serviço. Entretanto,

<sup>4</sup><https://awstccalculator.com/>

pode ser visto na Figura 3(b), que o tempo de *bufferização* inicial do vídeo aumenta conforme a quantidade de microserviços. Tal resultado ocorre pois a nuvem recebe todas as requisições, assim sua carga aumenta conforme cresce a quantidade de microserviços o que, conseqüentemente, degrada seu desempenho. Por sua vez, o mecanismo guloso possui uma decisão baseada na latência entre o cliente e o ponto de névoa, dessa forma sempre possui um tempo de *bufferização* baixo. Entretanto, esse mecanismo apresenta um baixo desempenho em relação ao tempo de migração do conteúdo, pois muitas vezes aloca o vídeo em pontos remotos da rede. O mecanismo aleatório possui uma decisão randômica o que faz seus resultados variarem muito, além disso sempre com um desempenho insatisfatório. Por fim, Fog4MS possui uma tomada de decisão inteligente que considera a latência e tempo de migração, obtendo o menor tempo de migração do conteúdo, mesmo que tenha um pequeno aumento no tempo de *bufferização*, *i.e.*, o custo/benefício do mecanismo é plenamente justificável.

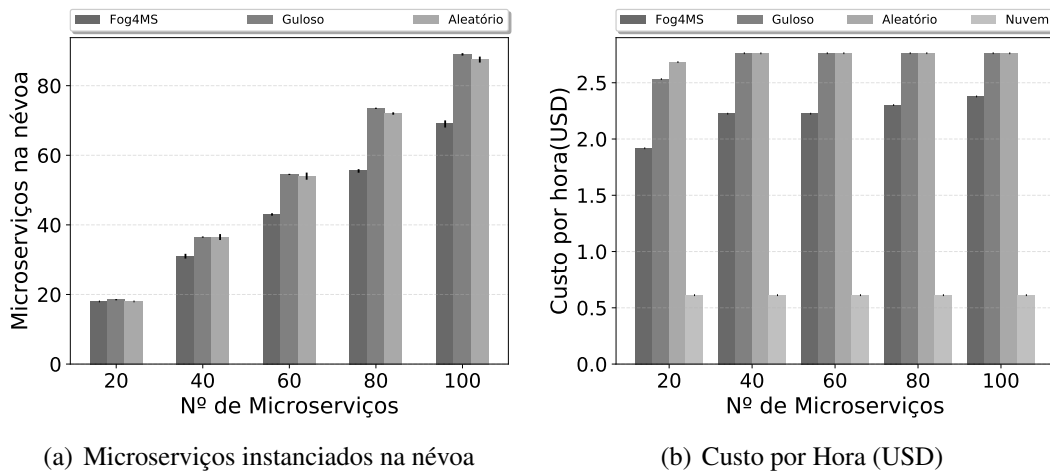
É importante destacar que Fog4MS é o único método que consegue diminuir o tempo de migração do conteúdo com o aumento do número de microserviços. Em comparação com o mecanismo guloso e aleatório, Fog4MS consegue reduzir o tempo de migração em cerca de 48% e 49% respectivamente, melhorando o tempo de inicialização do vídeo e, conseqüentemente, a QoE dos usuários. Isso é possível pois um maior número de requisições aumenta a chance de que clientes solicitem o mesmo conteúdo, de modo que a tomada de decisão inteligente do Fog4MS aloque esses microserviços em um mesmo ponto da névoa. Deste modo, o mecanismo utiliza a mesma cópia de um conteúdo para várias requisições e, assim, diminui o tempo de migração. Ademais, é importante salientar que a *bufferização* sofre maior variância por conta de fatores relacionados a rede e a especificidade do conteúdo requisitado.



**Figura 3. Tempo médio de migração e bufferização do conteúdo os microserviços**

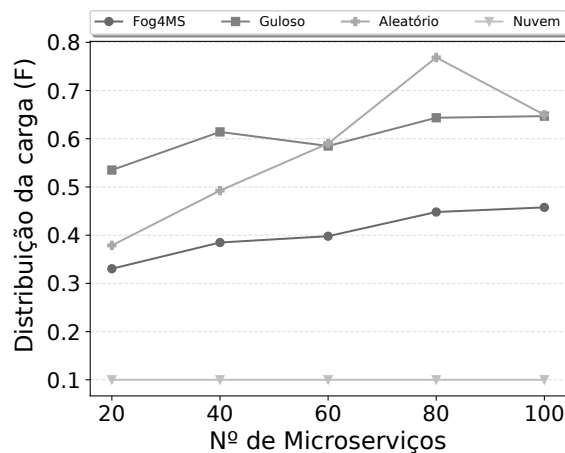
A Figura 4 mostra a quantidade de microserviços alocados na névoa em conjunto com o custo monetário do uso dos pontos de névoa para a alocação dos microserviços. A nuvem representa o cenário padrão, onde todos os microserviços são alocados nela, por isso não está presente na Figura 4(a). Além disso, esse cenário tem o menor custo possível como visto na Figura 4(b). A Figura 4(a) demonstra a quantidade de microserviços na névoa e assim pode-se relacionar ao custo presente na Figura 4(b). Dessa forma, é possível ver que tanto o método guloso quanto o aleatório alocam cerca de 90% dos microserviços na névoa, enquanto o Fog4MS aloca em média apenas 70% dos microserviços na névoa. Deste modo, economiza-se no custo geral em relação as demais, que utilizam vários pon-

tos da névoa para atender as requisições, enquanto o Fog4MS aproveita recursos da nuvem sem comprometer a eficiência e o desempenho. Por conseguinte, os mecanismos guloso e aleatório possuem um custo mais elevado, em média 23% e 25% maiores em relação ao Fog4MS, respectivamente. É importante ressaltar que o custo aumenta com um maior número de microserviços, porque a AHP considera a questão de distribuição de carga dos microserviços na rede e, conseqüentemente, instância mais microserviços na névoa.



**Figura 4. Custo médio dos pontos da névoa e a quantidade de microserviços instanciados na névoa para cada quantidade de microserviço**

A Figura 5 apresenta o índice de justiça para o uso dos recursos dos pontos de névoa no cenário, calculado pela Eq. 7. O índice expressa o quanto a carga está sendo balanceada na rede. O uso apenas da nuvem mostra o pior caso, *i.e.*, quando se aloca os microserviços apenas em um local. Os mecanismos guloso e aleatório tem uma distribuição semelhante, pois no aleatório a chance para a alocação em uma névoa é a mesma dentre todas as possíveis. No guloso, como as requisições aparecem no cenário em pontos seguindo uma distribuição uniforme, a alocação desse conteúdo acaba seguindo o mesmo princípio. Fog4MS possui um índice baixo em comparação aos demais, com um decréscimo de 33% em relação ao guloso e de 30% em relação ao aleatório.



**Figura 5. Índice de distribuição de carga dos microserviços**

Esse comportamento se dá pelo fato de que Fog4MS prioriza o desempenho e eficiência para a alocação do microserviço, sendo a métrica de utilização usada com menor impacto no cálculo do custo. Por exemplo, a utilização tem um peso de apenas 14% na tomada de decisão, sendo menos relevante na escolha da névoa para alocação. Além disso, Fog4MS ainda é capaz de distribuir a carga na rede quando necessário, conforme mostrado com o aumento do número de microserviços. Dessa forma, Fog4MS provê maior eficiência na distribuição de conteúdo, uma vez que a sua tomada de decisão prioriza eficiência em prol do balanceamento da rede.

## 5. Considerações Finais

Este artigo apresentou o mecanismo Fog4MS para alocação dinâmica de microserviços de *cache* em ambiente de computação em névoa para a distribuição de conteúdo VoD. O mecanismo Fog4MS leva em consideração atraso, tempo de migração do conteúdo e taxa de utilização na névoa para alocação de microserviços e recursos na infraestrutura de computação em névoa de múltiplos níveis. Para tomada de decisão, o mecanismo Fog4MS utiliza AHP para ajustar o grau de importância de cada critério.

Foram realizadas simulações com os mecanismos Fog4MS, guloso, aleatório e nuvem para disseminação de conteúdo de VoD. Os resultados obtidos mostraram que o mecanismo Fog4MS pode reduzir por quase a metade o tempo de migração do conteúdo e com um custo acerca de 25% menor comparados aos mecanismos guloso, aleatório. Como trabalhos futuros, pretende-se levar em consideração outras métricas para análise do QoE, ponderar mais parâmetros para a tomada de decisão da AHP, considerar DASH sobre a demanda de vídeo e instanciar o nível da névoa de acordo com a necessidade.

## 6. Agradecimentos

O presente trabalho foi realizado com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico – Brasil (CNPq).

## Referências

- Aazam, M. et al. (2016). MeFoRE: QoE based resource estimation at Fog to enhance QoS in IoT. In *23rd International Conference on Telecommunications (ICT)*, pages 1–5.
- Aazam, M. and Huh, E.-N. (2015). Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT. In *29th International Conference on Advanced Information Networking and Applications*, pages 687–694.
- Araújo, F., Rosário, D., Cerqueira, E., and Villas, L. A. (2019). A hybrid energy-aware video bitrate adaptation algorithm for mobile networks. In *15th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 146–153. IEEE.
- Bhardwaj, K. et al. (2019). Addressing the Fragmentation Problem in Distributed and Decentralized Edge Computing: A Vision. In *IEEE International Conference on Cloud Engineering (IC2E)*, pages 156–167.
- Chunlin, L. et al. (2019). Optimal media service selection scheme for mobile users in mobile cloud. *Wireless Networks*, 25(6):3179–3192.
- Cisco (2019). Cisco visual networking index: Forecast and trends, 2017–2022. Technical report, Cisco.
- He, Q. et al. (2017). Fog-based transcoding for crowdsourced video livecast. *IEEE Communications Magazine*, 55(4):28–33.

- Jain, R. K., Chiu, D.-M. W., and Hawe, W. R. (1984). A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*.
- Jalali, F. et al. (2016). Fog computing may help to save energy in cloud computing. *IEEE Journal on Selected Areas in Communications*, 34(5):1728–1739.
- Juluri, P., Tamarapalli, V., and Medhi, D. (2016). Measurement of Quality of Experience of Video-on-Demand Services: A Survey. *IEEE Communications Surveys & Tutorials*, 18(1):401–418.
- Mao, Y. et al. (2017). A Survey on Mobile Edge Computing: The Communication Perspective. *IEEE Communications Surveys Tutorials*, 19(4):2322–2358.
- Masip-Bruin, X. et al. (2016). Foggy Clouds and Cloudy Fogs: a Real Need for Coordinated Management of Fog-to-Cloud Computing Systems. *IEEE Wireless Communications*, 23(5):120–128.
- Menchaca-Mendez, R. et al. (2018). Opportunistic Mobile Sensing in the Fog. *Wireless Communications and Mobile Computing*, 2018:1–18.
- Ni, L. et al. (2017). Resource allocation strategy in fog computing based on priced timed petrinets. *IEEE Internet of Things Journal*, 4(5):1216–1228.
- Padhye, J. et al. (2000). Modeling tcp reno performance: a simple model and its empirical validation. *IEEE/ACM transactions on Networking*, 8(2):133–145.
- Riley, G. F. and Henderson, T. R. (2010). The ns-3 network simulator. In Wehrle, K., Günes, M., and Gross, J., editors, *Modeling and Tools for Network Simulation*, pages 15–34. Springer.
- Rosário, D. et al. (2018). Service Migration from Cloud to Multi-tier Fog Nodes for Multimedia Dissemination with QoE Support. *Sensors*, 18(2).
- Salahuddin, M. A. et al. (2018). A survey on content placement algorithms for cloud-based content delivery networks. *IEEE Access*, 6:91–114.
- Siavoshani, M. J., Pourmiri, A., and Shariatpanahi, S. P. (2017). Storage, communication, and load balancing trade-off in distributed cache networks. *IEEE Transactions on Parallel and Distributed Systems*, 29(4):943–957.
- Tang, J., Tay, W. P., and Wen, Y. (2014). Dynamic request redirection and elastic service scaling in cloud-centric media networks. *IEEE Transactions on Multimedia*, 16(5):1434–1445.
- Tian, Y. et al. (2018). A new live video streaming approach based on Amazon S3 pricing model. In *IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 321–328.
- Wang, S. et al. (2015). Dynamic service migration in mobile edge-clouds. In *IFIP Networking Conference (IFIP Networking)*, pages 1–9.
- Xiao, W. et al. (2015). Dynamic request redirection and resource provisioning for cloud-based video services under heterogeneous environment. *IEEE Transactions on Parallel and Distributed Systems*, 27(7):1954–1967.
- Zhang, Z.-H., Jiang, X.-F., and Xi, H.-S. (2016). Optimal content placement and request dispatching for cloud-based video distribution services. *International Journal of Automation and Computing*, 13(6):529–540.