



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA

Autorizada pelo Decreto Federal nº 77.496 de 27/04/76

Recredenciamento pelo Decreto nº 17.228 de 25/11/2016

PRÓ-REITORIA DE PESQUISA E PÓS-GRADUAÇÃO

COORDENAÇÃO DE INICIAÇÃO CIENTÍFICA



XXIII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIENTÍFICA E TECNOLÓGICA - 2019

COMUNICAÇÃO ENTRE INSTÂNCIAS DE NÉVOA EM AMBIENTE DE EMULAÇÃO MININET

Emerson Santos Lima¹ e Antonio Augusto Teixeira Ribeiro Coutinho²

1. Estagiário voluntário PEVIC/UEFS, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail: emerson.sl.1605@gmail.com
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail: augusto@ecomp.uefs.br

PALAVRAS-CHAVE: virtualização; emulação; computação em névoa.

INTRODUÇÃO

Atualmente as pessoas vivem conectadas a dispositivos como smartphones, smartwatches, e smart TVs. Além disto, casas são monitoradas por uma série de dispositivos como câmeras, sensores de presença (utilizado comumente em alarmes), sensores de temperatura, entre outros. Esses dispositivos têm a característica de utilizar uma rede para trocar informações com servidores e outros dispositivos onde, na grande maioria das aplicações, a rede usada é a internet. Este modelo se tornou popularmente conhecida como a Internet das Coisas (IoT).

Atualmente, as aplicações IoT adotam o modelo de nuvem. De acordo com o Instituto Nacional Americano de Padrões e Tecnologia (NIST, 2011), a computação em nuvem é um modelo baseado na demanda de recursos, como exemplo, servidores, armazenamento, aplicações ou serviços. Porém, quando se trata de escalabilidade, esse modelo é insuficiente para comportar o fluxo gerado pela IoT, devido a grande quantidade de dispositivos presentes hoje no mundo. A exigência de muita capacidade e processamento, aliada a alta latência (atraso na comunicação) na internet, são os principais problemas no modelo de computação em nuvem aplicado a IoT.

Em meio a essas dificuldades, surge o modelo de computação em névoa (*fog computing*) com o objetivo de sanar essas dificuldades. Segundo (Bonomi et al., 2012) computação em névoa é um sistema virtualizado que oferece serviços entre dispositivos finais e nuvem. A grande inovação da computação em névoa é a distribuição de recursos, de forma que os mais solicitados fiquem cada vez mais próximos dos dispositivos finais. Para isso, ela utiliza uma arquitetura hierárquica dividida em níveis, onde cada nível é responsável por fornecer algum recurso. A arquitetura mais simples possui três níveis: nuvem, névoa e dispositivos finais.

Em 2016, foi publicada a primeira arquitetura de referência em computação em névoa pelo OpenFog Consortium, um conjunto de empresas focada na padronização e desenvolvimento deste modelo. Essa arquitetura ainda possui pontos a serem esclarecidos, onde a maioria das propostas recentes focam em definir como as entidades devem interagir nesse modelo.

Por ser um conceito novo, a computação em névoa traz dificuldades aos pesquisadores, quando se refere a validação e testes de desempenho. Na tentativa de resolver esse problema,

foram adaptados alguns simuladores e emuladores de nuvem, mas que não conseguem ser integrados com sistemas reais. Por esse motivo (Coutinho et al., 2018) descreve a criação de um emulador para névoa, o FogBed. Um emulador é um software ou hardware com capacidade de reproduzir o comportamento de um sistema real, com desempenho muito semelhante, diferente de um simulador que foca na execução de etapas, onde o desempenho difere muito do sistema real.

O FogBed é um framework que permite aos desenvolvedores emular um ambiente de computação em névoa com arquitetura flexível. A versão descrita por (Coutinho et al., 2018) funciona em um computador local, que permite a adição de diversos dispositivos. No decorrer desse resumo serão apresentadas as ferramentas que foram utilizadas, para permitir que esse framework funcione de forma escalável, distribuindo-o em vários computadores. Ao final, são apresentados os resultados dos experimentos realizados.

METODOLOGIA

A solução do FogBed distribuído atende aos seguintes requisitos de baixo custo, suporte a serviços e protocolos do mundo real e escalabilidade. A seguir, serão descritas as tecnologias usadas para atender esses critérios.

A plataforma Docker (Docker, 2018) é uma tecnologia que explora uma funcionalidade do *kernel* linux, a tecnologia de containerização. Um container tem tudo necessário para executar uma aplicação (código, bibliotecas, configurações, ferramentas). É um recurso semelhante a uma máquina virtual, porém nele é possível compartilhar o mesmo kernel entre diferentes containers e separar todas as camadas acima entre diferentes aplicações.

De acordo com (Wette et al., 2014) MaxiNet é um framework que usa como base o emulador de rede Mininet, com o adicional de funcionar em clusters (computadores físicos trabalhando em conjunto). No MaxiNet, é possível emular grandes redes em algumas máquinas físicas e muitos nós virtuais. Cada computador do *cluster* é chamado de *worker*, e executa uma instância do Mininet em um nó da rede emulada.

Em (Coutinho et al., 2018), é apresentado o FogBed, um framework também baseado no Emulador de rede Mininet. Seu objetivo principal é fornecer um ambiente de computação em névoa virtualizado. Os nós virtuais são inseridos através de containers Docker em diferentes configurações de rede. Através de uma API, são fornecidas ferramentas para adicionar ou remover containers da rede de forma dinâmica. Também é possível alterar configurações dos containers como quantidade de memória e tempo de CPU. Essas configurações permitem ao desenvolvedor obter resultados mais precisos para testes de desempenho em ambientes de névoa.

Para atender aos requisitos citados no primeiro parágrafo deste tópico, o framework do FogBed foi estendido de forma a possibilitar sua execução em várias máquinas físicas, através da sua integração com o framework MaxiNet.

RESULTADOS E/OU DISCUSSÃO (ou Análise e discussão dos resultados)

Ao reunir todas as tecnologias citadas no tópico anterior, o resultado foi a criação de uma arquitetura flexível e escalável. Toda a arquitetura pode ser sintetizada através da análise do

fluxo de execução do FogBed distribuído. A Figura 1 demonstra os passos realizados para uma emulação no FogBed:

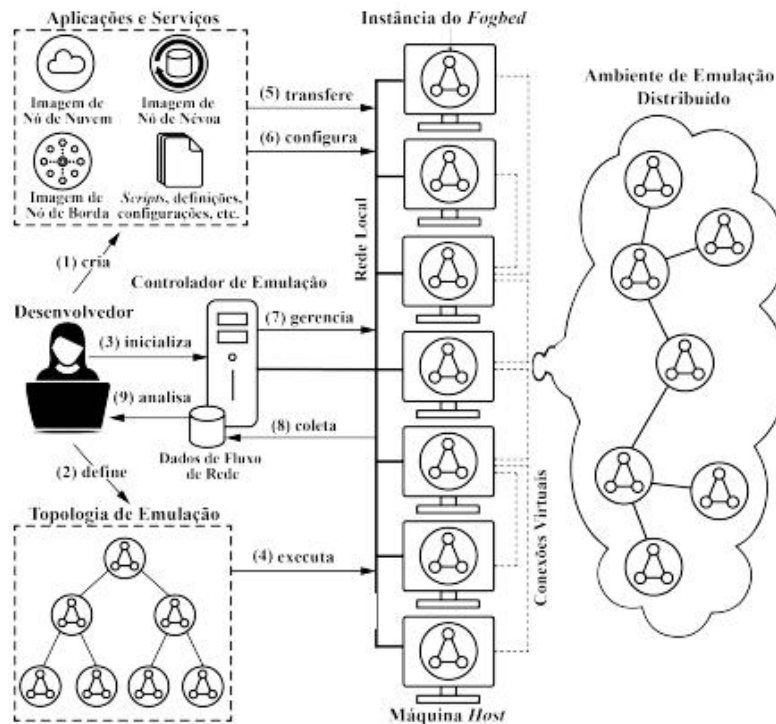


Figura 1: Fluxo de emulação FogBed

Por fim, foi aplicado um estudo de caso para comparar dois modelos: no modelo A, foi adotado a comunicação tradicional, uma aplicação de nuvem solicitando dados a dispositivos finais; no modelo B, foram usados nós de névoa intermediários com objetivo de minimizar o tempo de resposta e o tráfego de dados. Esses resultados podem ser observados nas Figuras 2 e 3, onde é possível perceber a vantagem do modelo B em relação ao modelo A comparando o tempo de resposta e o tráfego de dados.

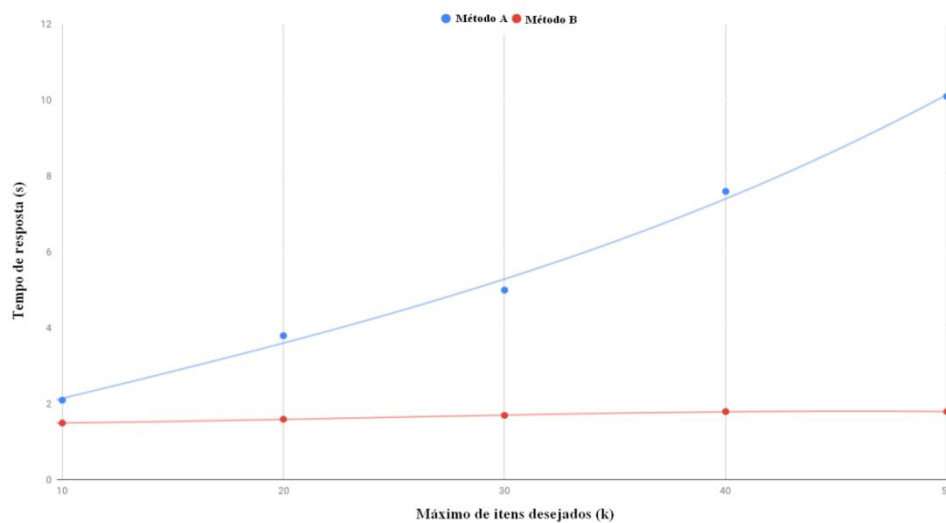


Figura 2: Evolução do tempo de resposta

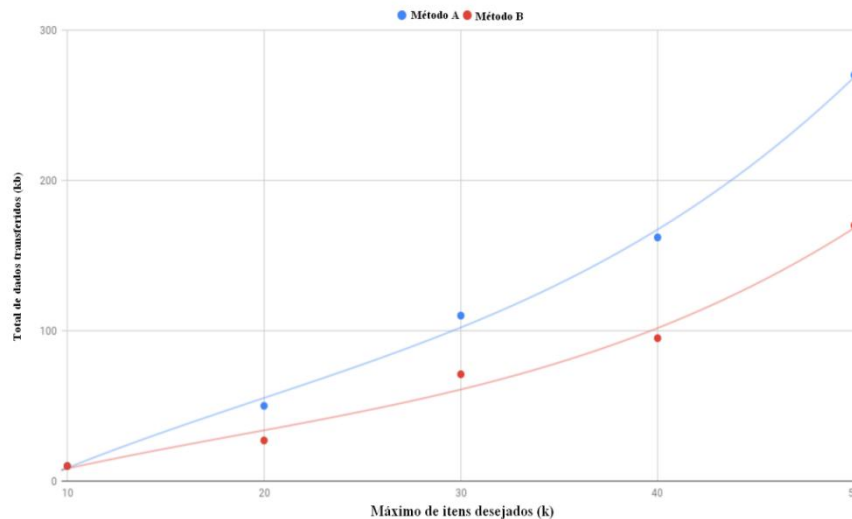


Figura 3: Evolução do total de dados transferidos

Conclusão

A Computação em névoa é uma tecnologia em ascensão com o objetivo de reduzir os problemas de escalabilidade e latência na rede. Ao longo deste resumo foi demonstrado brevemente o projeto de um emulador para computação em névoa, que atende aos requisitos de: baixo custo, escalabilidade, flexibilidade e desempenho. Para trabalhos futuros, deve-se considerar inserção de novas tecnologias na computação em névoa, como Blockchain, uma forma de aumentar segurança, usando a ferramenta FogBed distribuído para realizar experimentos e obter resultados mais precisos.

REFERÊNCIAS

NIST. 2011. The nist definition of cloud computing. Disponível em:

<http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>. Último acesso em: 26/07/2019

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. ACM, p. 13–16.

COUTINHO A.; GREVE F.; PRAZERES C.; CARDOSO J. 2018. Fogbed: *A rapid prototyping emulation environment for fog computing*. In: Communications Workshops (ICC Workshops), 2018 IEEE International Conference on. p. 1–7.

DOCKER. 2018. Página inicial do projeto. Disponível em: <https://www.docker.com/resources/what-container>. Último acesso em: 26/07/2019

WETTE P.; DRAXLER M.; SCHWABE A.; WALLASCHEK F.; ZAHRAEE M. H.; KARL H. 2014. Maxinet: *Distributed emulation of software-defined networks*. In: Networking Conference, 2014 IFIP, IEEE. pp. 1–9.

KIM H.; FEAMSTER N. 2013. Improving network management with software defined networking. In: IEEE Communications Magazine, vol. 51.