



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

XXVII SEMINÁRIO DE INICIAÇÃO CIENTÍFICA DA UEFS SEMANA NACIONAL DE CIÊNCIA E TECNOLOGIA - 2023

Balanceamento de Carga em Névoa das Coisas com Blockchain

**João Erick Barbosa Teixeira da Silva¹; Antonio Augusto Teixeira Ribeiro
Coutinho**²;

1. Estagiário PEVIC, Graduando em Engenharia de Computação, Universidade Estadual de Feira de Santana, e-mail:
jsilva@ecomp.uefs.br
2. Orientador, Departamento de Tecnologia, Universidade Estadual de Feira de Santana, e-mail:
augusto@ecomp.uefs.br

PALAVRAS-CHAVE: Internet das Coisas; Computação em Névoa; Blockchain

INTRODUÇÃO

É notória a evolução demonstrada pelo paradigma de Computação em Nuvem (*Cloud Computing*) ao longo dos anos. Devido a sua capacidade de armazenamento e processamento, de acordo com Aazam et al. (2014), esse modelo é o mais adotado em aplicações relacionadas a Internet das Coisas (*Internet of Things*, IoT), as quais produzem um grande volume de dados, e, por isso, demandam grandes capacidades de processamento e armazenamento (Moura e Hutchison, 2016).

Apesar do que foi citado anteriormente, segundo Coutinho et al. (2016), a Computação em Névoa (*Fog Computing*) traz consigo características que são vantajosas para aplicações IoT, quando comparada com a Computação em Nuvem, dentre elas, tem-se: maior proximidade dos usuários finais, mobilidade e localidade. Ainda que possua capacidades de processamento inferiores à Computação em Nuvem, são superiores aos nós/dispositivos que estão na borda da rede.

Realizar a integração entre borda/névoa e nuvem, de acordo com Castilho e Kamienski (2018), resulta em um aumento do desempenho da rede, beneficiando em aspectos de escalabilidade geral dos serviços, baixa latência, entre outros. Entretanto, devido às características descentralizadas deste modelo, é necessário utilizar tecnologias que forneçam serviços de maneira eficiente, autônoma e segura (Coutinho et al., 2016).

É nesse cenário que tecnologias de livro razão distribuído (*Distributed Ledger Technologies*, DLT) como *blockchain*, podem ser aplicadas, pois fornecem serviços distribuídos, sem a necessidade de uma autoridade central; e de maneira segura, pois faz uso de criptografia e é governada por mecanismos de consenso (Greve et al., 2018).

Em trabalhos anteriores, foi desenvolvido um algoritmo de balanceamento de carga na borda da rede, que faz uso das tecnologias Internet das Coisas e *blockchain*. Este trabalho busca integrar a camada de névoa, englobando mais de uma camada durante o processo de balanceamento buscando melhorias no desempenho da rede.

METODOLOGIA

Para atingir o objetivo proposto, inicialmente foram realizadas pesquisas e revisões bibliográficas a respeito dos conceitos sobre DLT e *blockchain*, principalmente para a DLT *Tangle* (Popov, 2018). Além disso, foram realizadas implementações visando o desenvolvimento da plataforma SOFT-IoT (Prazeres e Serrano, 2016), usada em trabalhos anteriores e adaptada neste trabalho às necessidades do balanceamento.

Posteriormente, foram iniciados estudos e testes sobre o algoritmo de balanceamento de carga desenvolvido em trabalho anterior. O objetivo foi entender o seu funcionamento visando adequações e migração do algoritmo da borda da rede para o ambiente de desenvolvimento em névoa no presente trabalho.

Por fim, foram configurados e executados experimentos sobre o estudo de caso proposto em um ambiente virtual no Laboratório de Redes e Sistemas Distribuídos (LARSID) do DTEC/UEFS. O objetivo dos experimentos foi validar se o novo algoritmo, juntamente com o novo módulo (*bundle*) proposto resultaram em melhorias no desempenho da rede. Para isso, os *gateways* e dispositivos da arquitetura foram emulados como containers *Docker*¹ usando dados de sensores simulados durante sua execução.

RESULTADOS E DISCUSSÕES

O desenvolvimento do estudo de caso envolveu o desenvolvimento de um *bundle* responsável pelo balanceamento de carga multicamada. Para manter o mesmo padrão, e facilitar o entendimento e futuras manutenções, foi acrescentado o prefixo *UP* para as solicitações do *gateway* sobrecarregado, e *DOWN* para as respostas de um *gateway* com disponibilidade presente em uma camada superior, resultando nas seguintes quatro novas mensagens:

1. *UP_LB_ENTRY*: Mensagem enviada por um *gateway* que se encontra sobrecarregado, solicitando que ocorra o balanceamento de carga;
2. *DOWN_LB_ENTRY_REPLY*: Resposta de um *gateway* não sobrecarregado de uma camada superior, enviada em resposta a um pedido de balanceamento de carga feito por outro *gateway*. Esta mensagem só é enviada se o *gateway* não ficar sobrecarregado ao aceitar um novo dispositivo. Caso contrário, um novo balanceamento de carga seria necessário, reiniciando o ciclo.
3. *UP_LB_REQUEST*: Quando um *gateway* sobrecarregado recebe uma mensagem do tipo *DOWN_LB_ENTRY_REPLY*, ele envia essa mensagem contendo os dados necessários para a realocação do dispositivo para o *gateway* disponível na camada superior;
4. *DOWN_LB_REPLY*: É uma mensagem de resposta enviada pelo *gateway* da camada superior que recebeu os dados do dispositivo, indicando que a realocação foi concluída com sucesso para o *gateway* que estava sobrecarregado anteriormente.

A Figura 1 ilustra um fluxograma geral do funcionamento do processo do balanceamento de carga multicamada. Como todo o sistema é descentralizado, inicialmente, todos os *gateways* verificam e publicam seu estado atual na rede, indicando se estão sobrecarregados ou não. A partir do momento que um *gateway* ultrapassa a sua capacidade máxima de dispositivos conectados, é enviada para a rede uma mensagem solicitando o balanceamento de carga. Outros *gateways* que possuem disponibilidade podem responder a essa solicitação, e iniciar a transferência do(s) dispositivo(s) excedente(s). Esse é o fluxo normal caso em uma camada, sempre exista pelo menos um *gateway* com capacidade de aceitar novos dispositivos sem ficar sobrecarregado. Todavia, em um cenário em que todos os demais *gateways* não têm disponibilidade, é quando ocorre o balanceamento de carga multicamada.

Foram feitos experimentos em um ambiente virtual com máquinas virtuais (VM) possuindo processador Intel (R) Core (TM) i7 1.30GHz, 12GB de memória RAM e sistema operacional *Linux*.

¹ <https://www.docker.com/resources/what-container/>

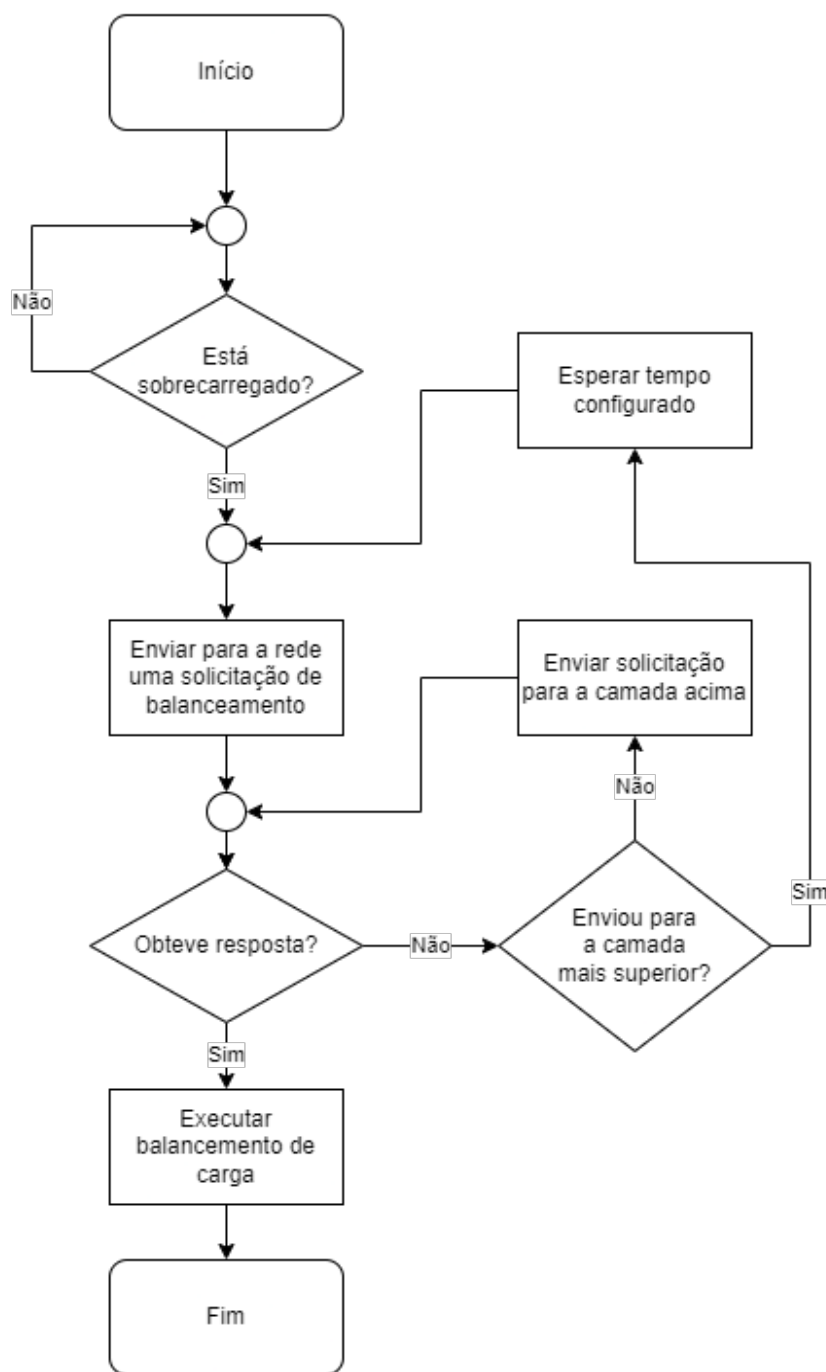


Figura 1: Fluxograma do algoritmo de balanceamento de carga multicamada.

O primeiro experimento teve como objetivo determinar quantos *gateways* uma máquina com as configurações fornecidas poderia suportar sem ultrapassar 80% da capacidade de memória RAM. Foi observado que a máquina conseguia suportar 17 *gateways*, resultando em um consumo de aproximadamente 9,7 GB de memória RAM.

Verificou-se também que a execução de um único nó da *blockchain Tangle* aumentava o consumo de memória RAM em 1,2 GB. É importante destacar que a máquina, quando iniciada e com apenas os processos padrões em execução, apresentava um consumo médio de 550 MB de memória RAM.

Um outro experimento foi conduzido para avaliar a relação entre a quantidade de *gateways* e dispositivos que uma máquina consegue suportar antes de ficar sem recursos disponíveis. Os resultados desse experimento são apresentados na Tabela 1.

Tabela 1: Quantidade de *gateways* e quantidade total de dispositivos.

Quantidade de <i>gateways</i>	Quantidade total de dispositivos	Quantidade de <i>gateways</i>	Quantidade total de dispositivos	Quantidade de <i>gateways</i>	Quantidade total de dispositivos
2	153	8	97	14	48
3	146	9	91	15	32
4	130	10	75	16	22
5	120	11	70	17	5
6	113	12	65	-	-
7	105	13	56	-	-

Também foi observado que a partir de 18 *gateways* em uma única máquina, o sistema começa a utilizar a memória de *swap*². Portanto, o número máximo de *gateways* listado na Tabela 1 é de dezessete (17).

CONSIDERAÇÕES FINAIS

Este estudo teve como objetivo validar e criar um novo modelo para equilibrar a carga de dispositivos virtuais de Internet das Coisas em um ambiente de névoa usando tecnologias de DLT/*blockchain*. Quando não é possível equilibrar a carga em uma camada, uma solicitação é feita à camada superior para realocar os dispositivos excedentes em um *gateway*, melhorando o desempenho da rede. Integrado a um projeto mais amplo de avaliação da escalabilidade na emulação de ambientes IoT, este estudo ampliou a arquitetura SOFT-IoT na rede do LARSID, desenvolvendo módulos e ferramentas para facilitar experimentos futuros. Embora alguns experimentos planejados não tenham sido realizados devido a problemas de conexão e manutenções, o modelo de balanceamento multicamada mostrou contribuição positiva para pesquisas futuras do Laboratório de Rede e Sistemas Distribuídos. Experimentos futuros podem ser realizados para avaliar o impacto do algoritmo de balanceamento em névoa proposto sobre métricas de desempenho da rede como o tempo de resposta entre os dispositivos da rede.

REFERÊNCIAS

- AAZAM, Mohammad et al. Cloud of Things: Integrating Internet of Things and cloud computing and the issues involved. In: **Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th-18th January, 2014. IEEE, 2014.** p. 414-419.
- CASTILHO, Gustavo Uruguay; KAMIENSKI, Carlos Alberto. Aplicação de Computação em Névoa na Internet das Coisas para Cidades Inteligentes: da Teoria à Prática. In: **Anais do XVI Workshop em Clouds e Aplicações.** SBC, 2018.
- COUTINHO, A. A. T. R.; CARNEIRO, Elisângela Oliveira; GREVE, Fabíola Gonçalves Pereira. Computação em névoa: Conceitos, aplicações e desafios. **Minicursos do XXXIV SBRC**, p. 266-315, 2016.
- GREVE, Fabíola et al. Blockchain e a Revolução do Consenso sob Demanda. **Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)-Minicursos**, 2018.
- MOURA, Jose; HUTCHISON, David. Review and analysis of networking challenges in cloud computing. **Journal of Network and Computer Applications**, v. 60, p. 113-129, 2016.

² <https://www.enterprisestorageforum.com/hardware/what-is-memory-swapping/>