

# Balanceamento de carga na borda da rede usando blockchain das coisas

Alexandre B. Costa<sup>1</sup>, Atslands R. da Rocha<sup>1</sup>, Flávia C. Delicato<sup>3</sup>, Jose N. de Souza<sup>2</sup>

<sup>1</sup>Departamento de Engenharia de Teleinformática, Campus do Pici

<sup>2</sup>Departamento de Computação, Campus do Pici  
Universidade Federal do Ceará, Fortaleza - CE, CEP 60020-181, Brasil

<sup>3</sup>Departamento de Ciência da Computação - Instituto de Computação  
Universidade Federal Fluminense

alexandrebc@alu.ufc.br, atslands@ufc.br, fdelicato@gmail.com, neuman@ufc.br

**Abstract.** *Fog computing has been used as a processing infrastructure for IoT applications, bringing benefits such as low latency for the applications. However, characteristics such as the high dynamism, distribution, and the heterogeneity of the fog environments bring up several challenges regarding the management of available resources. In this context, we present a decentralized and autonomous mechanism for resource management in fog computing. Our proposal is based on the associated use of blockchain and the autonomic computing principles to address the challenges of managing in the fog. The results show that the proposed mechanism distributes the load between the fog nodes in a balanced way, even in case of failures occurrence.*

**Resumo.** *A computação em névoa tem sido usada como infraestrutura de processamento para aplicações IoT, trazendo benefícios como baixa latência para a aplicação. Entretanto, características como o dinamismo, a distribuição e a heterogeneidade dos ambientes de névoa trazem à tona vários desafios no tocante ao gerenciamento dos recursos disponíveis. Nesse contexto, apresentamos um mecanismo para gerenciamento de recursos e balanceamento de carga de forma descentralizada e autônoma. O mecanismo baseia-se no uso associado de blockchain e dos princípios da computação autônoma para lidar com os desafios do gerenciamento na névoa. Os resultados mostram que o mecanismo proporciona uma distribuição da capacidade de processamento na borda, mesmo em caso de falhas.*

## 1. Introdução

A Internet das Coisas (*Internet of Things* - IoT) é um paradigma que prevê a conexão de objetos físicos heterogêneos via Internet. O avanço desta tecnologia possibilita que os mais diversos tipos de dispositivos sejam capazes de capturar e enviar dados do ambiente em que se encontram, os quais servem de insumos para detecção de eventos, tomadas de decisão e para a construção de serviços inteligentes e aplicações avançadas. Um grande fluxo de dados oriundos desses objetos inteligentes conectados em larga escala é esperado. Além disso, espera-se uma demanda crescente de processamento desses dados, o qual o próprio dispositivo, em geral, não é capaz de realizar, sendo necessário direcionar tal fluxo para outro(s) dispositivo(s).

A computação em névoa [Bellavista et al. 2019] tem sido amplamente utilizada como infraestrutura de processamento para IoT. Em [Cisco 2015] computação em névoa é descrita como uma extensão da nuvem até a borda da rede, trazendo para mais perto dos objetos IoT o poder de processamento e armazenamento dos dados produzidos pelos mesmos, permitindo assim, realizar análise de alguns dados nesse nível. Com a introdução de dispositivos na névoa, formam-se ambientes de computação em três níveis: os dispositivos IoT, os nós da névoa e a nuvem.

A grande quantidade e a diversidade de objetos em sistemas IoT, bem como sua característica de distribuição e dinamismo, dão origem a vários desafios no que tange ao gerenciamento dos recursos disponíveis em ambientes de computação em névoa. Observa-se, por exemplo, a necessidade de uma estrutura gerencial que realize de modo descentralizado a análise dos recursos do nível da névoa e a distribuição da carga de trabalho oriunda dos dispositivos IoT. Uma solução em potencial é a *blockchain* [Swan 2015]. Por possuir a capacidade de distribuição dos dados, a *blockchain* favorece a descentralização, e por ter seus dados registrados sequencialmente também é possível criar uma análise de uso dos recursos em uma série histórica. Tais características a tornam uma tecnologia promissora para o gerenciamento em ambientes de névoa.

O dinamismo, a larga escala e a dispersão geográfica entre os nós trazem a tona outro desafio de gerenciamento, que é a necessidade de os nós serem auto-gerenciáveis. O conceito de computação autônoma [IBM 2005] baseado no modelo MAPE-K (*Monitoring, Analysis, Planning, Execution and Knowledge*) auxilia no monitoramento de recursos de sistemas heterogêneos e distribuídos [IBM 2005]. Cada dispositivo que contém um *loop* de controle de *feedback* MAPE-K implementado torna-se auto-gerenciável, podendo realizar atividades de monitoramento, planejamento, análise e execução de acordo com regras definidas na base de conhecimento.

Nesse trabalho é proposto um mecanismo distribuído para gerenciamento dos nós da névoa, capaz de monitorar seus recursos e distribuir a carga de trabalho gerada pelos dispositivos IoT. Para tal, propomos o uso associado de uma *blockchain* como parte integrante da base de conhecimento distribuída nos nós da névoa, os quais implementam um *loop* MAPE-K. Dessa forma, em nossa proposta, são definidas tarefas autônomas de (i) autoconhecimento, em que os nós da névoa possuem o conhecimento do seu estado atual (referente ao uso de recursos) e dos demais nós deste nível; e (ii) autoconfiguração, em que a quantidade e os tipos de dispositivos IoT que devem ser atendidos são configurados, em tempo de execução, de acordo com os recursos monitorados do nó da névoa, havendo assim, um balanceamento de carga.

## 2. Trabalhos Relacionados

Um gerenciador de recursos eficiente envolve o processo de descoberta e identificação de recursos disponíveis, bem como a divisão de forma a maximizar sua utilização, seja em termos de custo, energia ou performance [Buyya and Dastjerdi 2016]. O balanceamento de carga é um problema importante para o gerenciamento de recursos, especialmente para aplicações sensíveis à latência. Para equilibrar a carga entre os nós da névoa com diferentes recursos, as tarefas são distribuídas entre eles de acordo com uma estratégia para evitar sub-carga ou sobrecarga.

Há diversos trabalhos que realizam o balanceamento de carga em ambientes de

névoa para aplicações IoT usando um dos três tipos de abordagens a seguir. A abordagem centralizada ([Manasrah et al. 2019] e [Pinto Neto et al. 2017]) em que um nó central realiza a estratégia para o balanceamento da carga. A abordagem descentralizada ([SONG Ningning 2016] e [Verma et al. 2016]) utiliza *clusters* para distribuir a capacidade de realização da estratégia para o balanceamento. E, ainda existem abordagens híbridas ([Li et al. 2018] e [Xu et al. 2018]) que buscam um *trade-off* entre as vantagens e desvantagens das outras soluções. Porém, nenhum dos trabalhos anteriores buscam gerenciar a arquitetura da computação em névoa em pelo menos dois níveis, o balanceamento ocorre nos nós da névoa sem gerenciar os dispositivos IoT.

Em [Pesic et al. 2018] os autores propõe uma solução capaz de atuar no gerenciamento da própria névoa e validam-na através de um *testbed* da topologia completa de uma arquitetura de névoa, mostrando o desempenho da solução ao melhorar a resiliência e a capacidade de resposta do sistema de computação em névoa no contexto de alterações na topologia. O autor usa a nuvem como ponto central para realizar o gerenciamento, de modo que ela possui informação sobre todos os nós nos três níveis da infraestrutura.

[Prazeres and Serrano 2016] propõe uma implementação do paradigma FoT (*Fog of Things*), chamada SOFT-IoT, que permite que parte do processamento de dados e operações de entrega de serviços sejam processados em nós na borda da rede. Em [Sousa 2018] é desenvolvido o M2-FoT, um sistema de gerenciamento que usa o SOFT-IoT como base, o qual se propõe a gerenciar os componentes da infraestrutura IoT permitindo o monitoramento, manutenção e configuração utilizando a arquitetura SNMP.

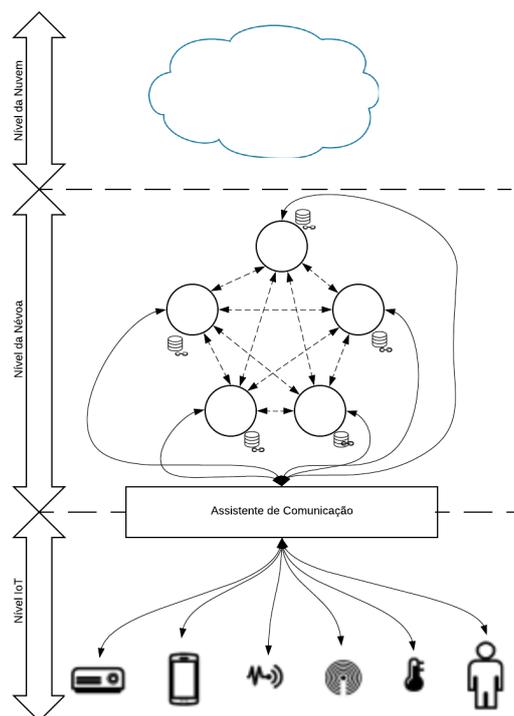
Em [Batista et al. 2018] o autor propõe um balanceamento de carga usando o paradigma FoTO FoT-Gateway é um nó gateway dentro do paradigma que realiza a comunicação com os dispositivos IoT (FoT-Devices) em diversos protocolos de comunicação e permite que o dado seja consumido por HTTP pelos nós da névoa. O trabalho propõe solucionar o problema de balanceamento com algoritmos que são executados dentro dos FoT-Gateways.

Esse trabalho possui uma contribuição na descentralização da gerência, permitindo que os nós do nível da névoa possuam um conhecimento de seus recursos (autoconhecimento) e dos demais nós e consiga balancear a carga de entrada (autoconfiguração).

### **3. Mecanismo de Gerenciamento de Recursos para Ambientes de Névoa**

Os componentes do mecanismo proposto são ilustrados na Figura 1. Em nossa proposta, quando um dispositivo IoT se conecta à rede, ele envia uma mensagem *broadcast* de anúncio e assim, recebe do *assistente de comunicação* a informação à qual nó da névoa deve se conectar. Para fins de balanceamento de carga, o nó selecionado deve ser o nó com menor gasto atual de recursos. Em tempo de execução, esse dispositivo pode ter a conexão refeita em função do consumo de recursos devido à autogerência de cada nó. O *assistente de comunicação* também é responsável pela comunicação bilateral entre o nó da névoa e os dispositivos IoT conectados.

Para a realização do balanceamento de carga, propomos uma abordagem de sistema autônomo baseado no modelo MAPE-K para ambientes de névoa. O ciclo MAPE-K engloba as atividades de monitoramento, análise, planejamento e execução, todas suportadas pela base de conhecimento. Uma *blockchain* é associada e distribuída como parte integrante da base de conhecimento (contendo os dados sobre os recursos dos nós da névoa



**Figura 1. Mecanismo de Gerenciamento de Recursos para Ambientes de Névoa**

conectados à rede) dos elementos autônômicos, i.e, nós do nível da névoa que implementam o modelo MAPE-K. No nível da névoa foi utilizada uma rede privada *Ethereum* [Buterin 2013], em que cada nó possui uma cópia da *blockchain* como parte integrante de sua base de conhecimento, permitindo que os nós sejam capazes de enviar e visualizar os dados publicados pelos outros nós por meio de um contrato inteligente. Um contrato inteligente é um conjunto de regras definido para reforçar a negociação entre duas entidades, proporcionando confiabilidades nas transações. Assim, fazendo uso de eventos criados no contrato inteligente, os nós recebem os dados assim que estes são publicados na rede.

O contrato inteligente tem como objetivo a criação de métodos para que os nós da névoa possam tanto inserir dados sobre os seus recursos na *blockchain*, bem como obter os dados provenientes dos outros nós. Para colocar o contrato em produção, é necessário conectar um nó com a rede pretendida, também chamado de fornecedor, para que as transações do contrato que será enviado possam acontecer. Como fornecedor foi utilizado um nó da névoa como nó minerador. Para que isso seja possível foi utilizado o *Geth*, um cliente responsável pela interação com as redes *Ethereum*, o qual torna o nó local em um nó da rede. Para o desenvolvimento do contrato inteligente foi utilizado um framework que possibilita a criação de Dapps para *blockchains* compatíveis com Ethereum Virtual Machine. O Truffle oferece um ciclo completo de desenvolvimento em *Solidity*, desde compilação, *linking*, *deployment* e testes.

O serviço de monitoramento do ciclo MAPE-K ocupa-se em obter os dados necessários para o autoconhecimento. Desta forma, o nó da névoa monitora seus próprios recursos (e.g.: CPU, memória RAM e nível de bateria) e insere esses dados na *blockchain*. Após a inserção, é disparado um evento definido no contrato inteligente para alertar a todos os nós sobre este novo dado que foi adicionado à base de conhecimento.

O serviço de análise é responsável pelo autodiagnóstico, tratando os dados recebidos pelo serviço anterior a fim de obter informações importantes para a tomada de decisão. Os dados dos recursos do nível da névoa são analisados em tempo de execução e transformados em eventos. Os eventos utilizam regras do tipo (SE-ENTÃO) para avaliar se o valor do recurso encontra-se abaixo ou acima em relação a um critério definido. Para este trabalho, foi usado como critério a média do uso do recurso nos nós de névoa ativos.

No serviço de planejamento, as ações necessárias são estruturadas para que o balanceamento de carga seja realizado. Esta etapa seleciona um procedimento para realizar uma alteração desejada no elemento gerenciado após o diagnóstico dado pelo serviço de análise. Após a análise comparativa é elaborado um plano de ações que se restringe à verificar se um novo dispositivo IoT conectou à rede ou se algum dos nós de névoa está consumindo mais recursos que os outros nós. Este plano de ação emite um alerta para que o sistema seja capaz de entender onde o dispositivo deve realizar a conexão para realizar o balanceamento da carga, dando início à etapa de execução no elemento gerenciado. Ao detectar que o comportamento atual deve ser alterado utilizando como base as ações recomendadas pela etapa de planejamento, o mecanismo proposto verifica qual nó da névoa possui o menor uso de recursos no momento e dá início ao processo de reconexão dos dispositivos conectados ao nó de maior consumo para o nó de menor consumo.

O serviço de execução é responsável por realizar a autoconfiguração sob a ótica do nó da névoa, realizando as modificações necessárias para que as configurações satisfaçam as necessidades do elemento gerenciado. Após receber o plano de ações, são executadas as ações estabelecidas no serviço de planejamento. Para isso, o nó da névoa comunica-se com o *assistente de comunicação*. As configurações necessárias são enviadas por meio de mensagens para o dispositivo IoT conectado ao nó que possui um consumo de recurso acima da média. As mensagens contêm instruções necessárias para que o dispositivo realize uma nova conexão, desta vez, com o nó que possui o menor consumo de recurso.

#### **4. Design experimental**

Foram realizados três experimentos (I, II, III), com os seguintes objetivos: (i) avaliar o custo de *deploy* da *blockchain* como parte integrante do mecanismo de gerenciamento; (ii) avaliar a proposta em relação ao balanceamento de carga em cenário adaptativo; (iii) avaliar o comportamento da proposta em caso de falha. Diferentes cenários (A, B, C e D) foram criados a fim de avaliar o mecanismo proposto nos experimentos II e III. Para cada cenário, dois parâmetros são definidos: (i) o tempo de criação entre os dispositivos IoT (em segundos); e (ii) período da pausa (em minutos) que ocorre a cada 25 dispositivos criados. A pausa é adicionada para verificar como o mecanismo reage se houver algum tempo sem perturbação, afim de que as modificações sejam feitas em um período com menos mudanças. Para o Experimento II, os valores dos parâmetros tempo de criação e o período de pausa foram 1s e 0m (Cenário A), 10s e 0m (Cenário B), 1s e 5m (Cenário C) e 1s e 10m (Cenário D). Para o Experimento III foi usado o Cenário C com os valores de 1s e 5m. Esses valores foram encontrados empiricamente em simulações realizadas (Experimento II). Além das variações dos dois parâmetros, foi observada uma limitação para a criação de até 110 dispositivos por cenário devido às configurações da máquina virtual responsável pela criação dos dispositivos. Após atingir esse valor, ocorria *starvation*.

Para os experimentos realizados, uma máquina física (Windows 10, CPU Intel(R)

Core(TM) i7-4750HQ @ 2.0 GHz e 8GB RAM) foi usada para emular a arquitetura da computação em névoa de três níveis. Quatro máquinas virtuais foram criadas para atuarem como nós da névoa. Três nós da névoa, com configuração similar (Tipo B - Ubuntu Mini 18.04, 1 core, 512MB RAM), se conectam à rede e submetem as transações (dados sobre seus recursos) para a *blockchain*. A configuração do Tipo B foi baseada na configuração de um *Raspberry Pi Zero W*, com o intuito de avaliar uma solução de baixo custo. Devido à necessidade computacional maior para os nós mineradores realizarem o trabalho de inserir os dados dos nós na *blockchain*, foi criada uma outra máquina virtual com maior poder computacional (definido empiricamente durante os experimentos) que as demais. Desta forma, a máquina principal (Tipo A - Ubuntu Mini 18.04, 1 core, 2GB) é responsável por minerar sozinha usando o mecanismo de consenso *Proof of Authority* e não participa dos experimentos de balanceamento de carga. Uma quinta máquina virtual (Tipo B) foi criada exclusivamente para simulação da criação dos dispositivos IoT.

A comunicação bilateral presente no *assistente de comunicação* ocorre através do protocolo MQTT. Desta forma, por meio de tópicos, os dispositivos IoT podem enviar seus dados brutos, bem como receber informações sobre como refazer a conexão para a realização do balanceamento da carga de trabalho.

O Experimento I tem como produto final a publicação do contrato inteligente na rede privada e busca avaliar a solução utilizando a *blockchain*. Após a compilação do contrato foi realizada uma simulação de publicação, em que é possível ver dados (como bloco e *gas* usado, dentre outros) que são utilizadas para avaliar o custo da solução. Após esta simulação, tem-se o início do processo de publicação. O processo demora alguns segundos, pois ocorre o processo de mineração do bloco, onde o contrato será armazenado. Na *Ethereum* o tempo médio de mineração de um bloco é de 15 segundos. Para validação da proposta o tempo de mineração não acarretou em problemas, mas as regras de dificuldade da rede privada podem ser ajustadas para diminuir o tempo de mineração, permitindo que a solução seja utilizada, por exemplo, em aplicações de tempo real. Ao fim do processo de publicação, o *log* mostra o endereço do contrato na rede, com o qual é possível se comunicar publicamente e escutar os eventos emitidos pelo contrato na rede.

No Experimento II, para validar a habilidade de distribuição de carga da proposta, foram analisados os recursos CPU e memória RAM durante a alocação dos dispositivos IoT. A RAM foi escolhida como parâmetro para o balanceamento de carga devido a baixa variação na CPU que permaneceu entre 2% e 10% durante todo o experimento, incluindo o processo de conexão dos dispositivos ao nó da névoa. Em caso de mudança de recurso ou técnica, o serviço de análise do MAPE-K deve ser modificado. No início do Experimento II, cada nó da névoa monitora o consumo de seus recursos para ser utilizado no serviço de análise do MAPE-K. Dado início às fases do MAPE-K, é possível indicar aos dispositivos IoT a qual nó se conectar baseado no uso de recurso. Em seguida, são criados virtualmente os dispositivos para gerar fluxo de informação. Conforme os dispositivos são adicionados e as pausas são realizadas é feita uma verificação do comportamento, averiguando se a carga encontra-se distribuída igualmente entre os nós.

No Experimento III, uma falha é forçada para simular o desligamento por escassez de recurso para demonstrar a atuação do mecanismo proposto realocando a outros nós os dispositivos que antes estavam conectados ao nó que foi desligado. Dessa forma, pode-se avaliar como o mecanismo reage em caso de falha de um dos nós da névoa.

## 5. Resultados

No Experimento I, após a criação do contrato inteligente foi realizada a publicação na rede privada. Pelos valores obtidos (Tabela 1), observa-se que esta é uma solução de baixo custo para a rede. As primeiras métricas indicam a quantidade de blocos usados e o tempo para publicação, os quais impactam diretamente no espaço alocado em disco e custo computacional. O tempo necessário para a rede minerar o contrato foi de 12s, ocupando menos de um bloco, assim, há um baixo custo computacional para publicar a solução. Uma quantidade de *gas* é cobrada em toda transação que ocorre na *Ethereum* e toda vez que algum contrato é executado. As taxas cobradas são calculadas através da multiplicação do preço do *gas* por sua quantidade. A unidade de medida do *gas* é o Wei que representa a menor divisão possível do Ether (ETH), ou seja, o *gas* está relacionado diretamente ao impacto financeiro na publicação do contrato. O custo total para a publicação foi de R\$ 5,35 na cotação de 01 de abril de 2020 (1 ETH = R\$ 714,79).

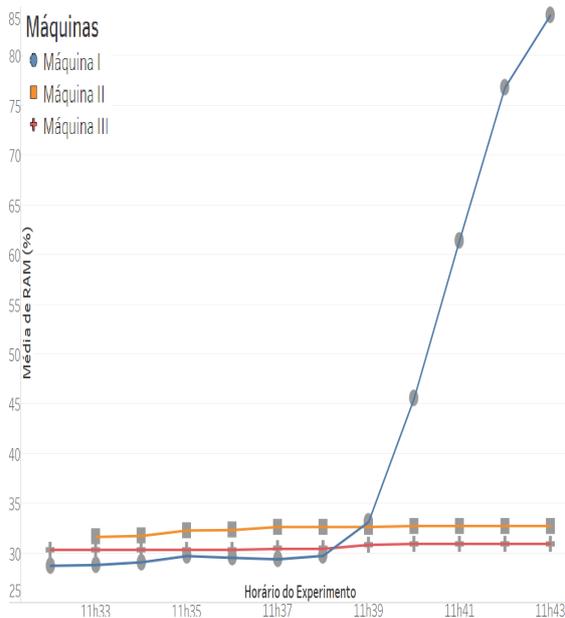
**Tabela 1. Resultados do Experimento I**

Métricas	Valores
Blocos usados	0
Tempo para publicação (s)	12
Gas usado	374309
Preço do gas (gwei)	20
Valor enviado (ETH)	0
Custo total (ETH)	0.00748618

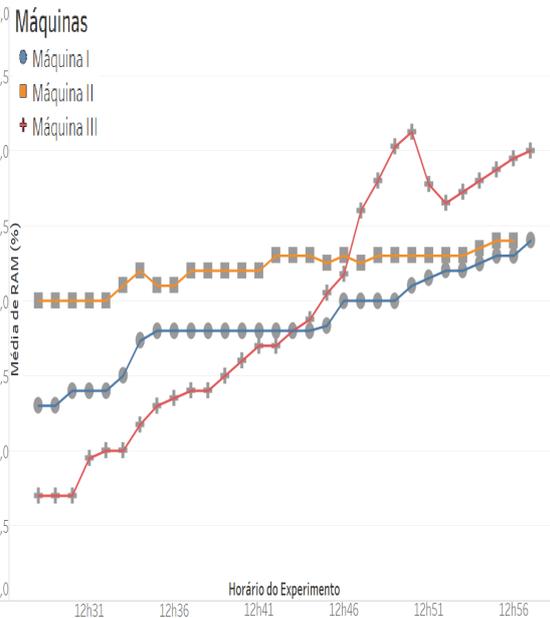
Além da validação dos custos para a publicação do contrato inteligente é possível verificar o custo da solução como um todo. A partir das informações sobre a rede Ethereum é possível prever o quanto essa solução custa em disco para manter o balanceamento de forma descentralizada. Considerando que um bloco contém aproximadamente 70 transações e que os blocos possuem menos de 2 KB, logo, cada transação possui cerca de 27 bytes. Se cada nó só realiza uma transação por tempo de mineração (15 segundos), em um ambiente com cinco nós na névoa em um mês haveria um consumo de 23MB.

No cenário A do Experimento II (Fig. 2), a Máquina I é identificada como o nó da névoa com mais recursos disponíveis, portanto as novas conexões de dispositivos IoT foram alocadas a este nó. Os dispositivos IoT foram criados rapidamente e como é possível observar em  $t = 11h38$  houve um crescimento repentino no uso do recurso. No período de 5 minutos, o consumo do nó saltou de 29.4% para 84.1% e o nó foi incapaz de refazer as conexões dos dispositivos devido ao estresse repentino.

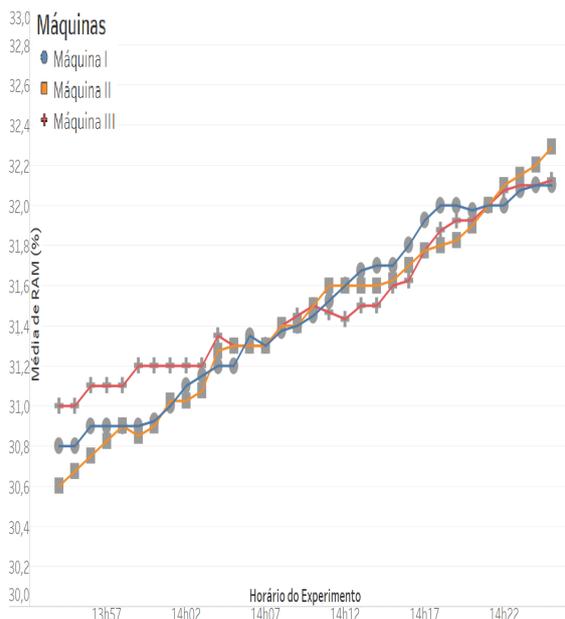
No cenário B do Experimento II (Fig. 3) há uma melhora na resposta dos nós quanto a alocação dos dispositivos devido ao aumento do tempo de espera. Em  $t = 12h43$ , a Máquina III (nó com mais recurso disponível) passa a ter um consumo de recurso maior que a Máquina I, fazendo com que novos dispositivos sejam alocados agora à Máquina I em vez da Máquina III. Assim, o crescimento do consumo dos recursos de cada nó da névoa é observado por todos os nós, tendendo à convergir para um valor médio de consumo de memória RAM, como se pode ver ao final do experimento. Entretanto, devido à grande quantidade de dispositivos alocados inicialmente à Máquina III, é observado tardiamente o processamento das mensagens que é exigido do nó pelos dispositivos. Desta



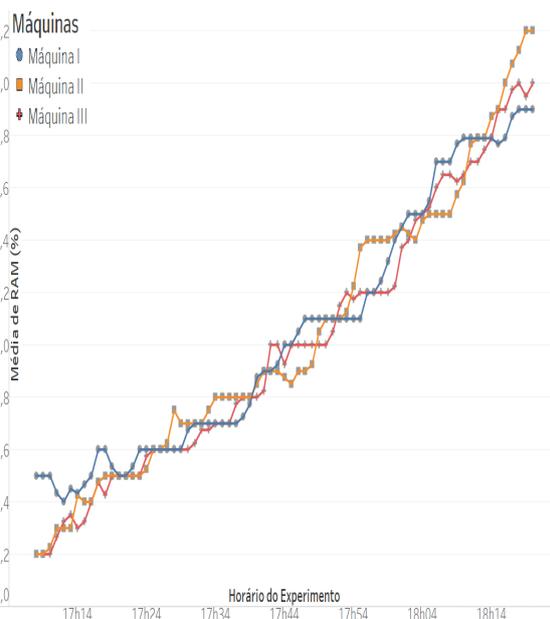
**Figura 2. Experimento II - Cenário A**



**Figura 3. Experimento II - Cenário B**



**Figura 4. Experimento II - Cenário C**



**Figura 5. Experimento II - Cenário D**

forma, observa-se um crescimento maior neste nó e, então, tem início o processo de reconexão dos dispositivos. Nota-se em  $t = 12h50$  o decaimento do consumo na Máquina III enquanto há o aumento do consumo de recursos nas outras máquinas da rede.

Nos cenários seguintes (C e D), apesar da rápida criação de dispositivos, observa-se que devido ao tempo de espera adicionado, houve uma melhora no comportamento dos nós, pois o uso dos recursos dos nós, nos cenários A e B, não convergiu para um valor médio em momento algum. Ao longo do experimento no Cenário C (Fig. 4), por exemplo, os nós convergiram para um mesmo valor médio de consumo de recursos, sendo possível visualizar em  $t = 14h07$  que todos os nós possuem o mesmo valor (31,3%). Esse

comportamento foi possível devido ao tempo de espera inserido, assim, o mecanismo teve tempo para processar as mensagens recebidas pelos dispositivos sem ser congestionado com muitas mensagens. Ao dobrar o tempo de espera no cenário D (Fig. 5) observa-se a tendência do sistema a convergir desde o início, além de possuir períodos de tempo maiores em que todos os nós possuem um mesmo valor médio de consumo de recursos.

No Experimento III (Fig. 6) é usado o cenário C devido ao seu comportamento de convergência para um valor médio de uso de recursos em menor tempo. Neste experimento, a Máquina II foi o nó da névoa que iniciou com a maior quantidade de recursos disponível e, em  $t = 19h53$ , observa-se que ela possui a maior quantidade de dispositivos IoT conectados. Então, a Máquina II foi escolhida para ser desligada abruptamente, simulando um desligamento por escassez de recurso. É visto que após o desligamento, os nós disponíveis realizam o balanceamento dos dispositivos antes conectados ao nó desligado.

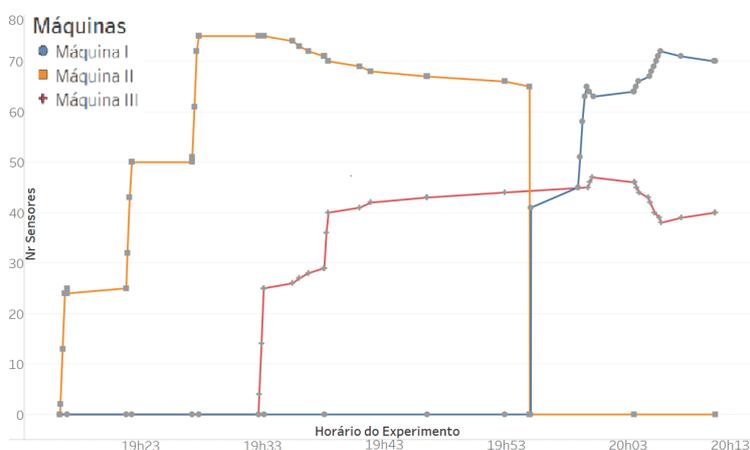


Figura 6. Experimento III.

## 6. Conclusão

Nesse artigo é apresentado um mecanismo distribuído para autogerenciamento de nós em ambientes de computação em névoa. Tal mecanismo monitora os recursos dos nós e distribui a carga de trabalho gerada pelos dispositivos IoT. Para tal, é utilizada uma *blockchain* associada ao conceito de computação autônoma. A principal contribuição do trabalho é a capacidade distribuída dos nós da névoa de autoconhecimento e autoconfiguração. Os resultados mostram que o mecanismo funciona como previsto, proporcionando a distribuição da capacidade de processamento na borda, evitando que poucos nós exerçam a maior parte do trabalho, mesmo em caso de falhas. A *blockchain* também provou ser uma solução de baixo custo em relação ao *deploy*.

## 7. Agradecimentos

Os autores agradecem ao apoio da CAPES - Código de Financiamento 001, FAPESP, (processo 2015/24144-7) e do CNPq (processo 432820/2016-7).

## Referências

Batista, E., Figueiredo, G., Peixoto, M., Serrano, M., and Prazeres, C. (2018). Load balancing in the fog of things platforms through software-defined networking. In *IEEE*

- Int. Conf. on Internet of Things and IEEE Green Computing and Communications and IEEE Cyber, Physical and Social Computing and IEEE Smart Data*, pages 1785–1791.
- Bellavista, P., Berrocal, J., Corradi, A., Das, S. K., Foschini, L., and Zanni, A. (2019). A survey on fog computing for the internet of things. *Pervasive and Mobile Computing*, 52:71–99.
- Buterin, V. (2013). Ethereum whitepaper. Technical report, Ethereum.
- Buyya, R. and Dastjerdi, A. V. (2016). *Internet of Things: Principles and paradigms*. Elsevier.
- Cisco (2015). Fog computing and the internet of things: Extend the cloud to where the things are. Technical report, Cisco Internet Business Solutions Group (IBSG).
- IBM (2005). White paper: An architectural blueprint for autonomic computing. Technical report, IBM Corporation.
- Li, C., Zhuang, H., Wang, Q., and Zhou, X. (2018). SSLB: Self-similarity-based load balancing for large-scale fog computing. *Arabian Journal for Science and Engineering*, 43(12):7487–7498.
- Manasrah, A., Aldomi, A., and Gupta, B. B. (2019). An optimized service broker routing policy based on differential evolution algorithm in fog/cloud environment. *Cluster Computing*, 22.
- Pesic, S., Tomic, M., Iković, O., Ivanovic, M., Radovanovic, M., and Boscovic, D. (2018). *Context Aware Resource and Service Provisioning Management in Fog Computing Systems*, volume 737, pages 213–223. .
- Pinto Neto, E. C., Callou, G., and Aires, F. (2017). An algorithm to optimise the load distribution of fog environments. In *IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC)*, pages 1292–1297.
- Prazeres, C. and Serrano, M. (2016). Soft-iot: Self-organizing fog of things. In *2016 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 803–808.
- SONG Ningning, GONG Chao, A. X. Z. Q. (2016). Fog computing dynamic load balancing mechanism based on graph repartitioning. *China Communications*, 13(3):156.
- Sousa, N. R. e Prazeres, C. (2018). M2-fot: a proposal for monitoring and management of fog of things platforms. *IEEE Symposium on Computers and Communications*.
- Swan, M. (2015). *Blockchain: Blueprint for a new economy*. ”O’Reilly Media, Inc.”.
- Verma, S. K., Yadav, A. K., Motwani, D., Raw, R. S., and Singh, H. K. (2016). An efficient data replication and load balancing technique for fog computing environment. *Int. Conf. on Computing for Sustainable Global Development*, pages 2888–2895.
- Xu, X., Fu, S., Cai, Q., Tian, W., Liu, W., Dou, W., Sun, X., and Liu, A. (2018). Dynamic resource allocation for load balancing in fog environment. *Wireless Communications and Mobile Computing*, 2018:1–15.