Arquitetura Resiliente MFog para Aplicações em névoa

Marcos Francisco da Silva[1], **Aleteia Araujo**[1]

[1]Departamento de Ciência da Computação, Universidade de Brasília – UnB

marcos.silva@aluno.unb.br, aleteia@unb.br

Abstract. *There are situations where Internet access is precarious, such as in places where natural disasters have occurred. In these situations, the use of software, which assist in aid and rescue operations, can be hampered due to this precariousness. A possible solution is the migration of these applications to a layer closer to the end user. In this context, an architecture for applications in fog (MFOG) was proposed with a focus on application resilience, tests carried out in a case study indicated the feasibility of the proposed architecture for these scenarios.*

Resumo. *Existem situações onde o acesso à Internet é precário, como por exemplo em lugares onde ocorreram desastres naturais. Nessas situações o uso de softwares, que auxiliam nas operações de ajuda e resgate, pode ser dificultado devido a essa precariedade. Uma possível solução é a migração dessas aplicações para uma camada mais próxima do usuário final. Nesse contexto, foi proposto uma arquitetura para aplicações em névoa (MFOG) com foco em resiliência de aplicações, testes realizados em um estudo de caso indicaram a viabilidade da arquitetura proposta para esses cenários.*

1. Introdução

Na ocorrência de eventos como furacões, enchentes e outros desastres naturais são necessárias ações de agências governamentais para socorro da população. Para coordenar essas operações, de modo geral, é necessário uso de sistemas de software táticos e operacionais que auxiliam as equipes na tomada de decisões. Esses softwares são utilizados por agentes ligados à Centros de Coordenação que possuem a função de integrar diversas agências governamentais com intuito de fornecer soluções para os problemas que surgem durante uma operação. No entanto, em boa parte dos casos, os servidores dos softwares táticos estão instalados em local distante da ocorrência, sendo o acesso possibilitado através de *link* com a Internet, isso gera uma grande dependência com a capacidade do *link* disponível no local da catástrofe, e devido a precariedade dessas conexões a atuação de forma coordenada pelos agentes que compõem os Centros de Operações pode ficar comprometida.

Uma possível solução para a perda de acesso aos serviços em locais deste tipo é a transferência de parte do processamento da aplicação para próximo do usuário final. Essa estratégia fornece maior resiliência durante os períodos de indisponibilidade de conexão. Esse conceito é uma das bases da computação em névoa (*fog computing*) [Habibi et al. 2020]. Essa transferência pode ser para a borda da rede ou no caminho entre o cliente e a nuvem, ou seja, em qualquer equipamento utilizado para fazer a interligação

entre o dispositivo do usuário final e a nuvem. Nesse contexto, cada dispositivo, ou um conjunto de dispositivos interligados, são considerados nós da névoa.

Uma das formas de se operacionalizar a computação em névoa é a migrações das aplicações para nós dessa camada [Prokhorenko and Babar 2020]. Diversos trabalhos na literatura propõe implementações arquiteturais para utilização de aplicações em *fog*. Uma técnica comum nessas propostas é a distribuição dos componentes da aplicação pelos nós da névoa de forma orquestrada, de tal forma que os recursos de hardware são utilizados de maneira mais eficiente [Santos et al. 2019, Rosário et al. 2018]. No entanto, nesses trabalhos não são analisados situações de perda de conexão entre os nós, em cenários no qual um nó de acesso ficará isolado de todos os demais. Dessa forma, a principal contribuição deste trabalho é apresentar uma arquitetura resiliente, chamada *MFog*, que adota mecanismos que possam garantir o fornecimento do serviço mesmo em situações em que um nó fique isolado.

2. Arquitetura Resiliente MFog para Aplicações em névoa

Uma pesquisa abrangente sobre diversos modelos arquiteturais para infraestrutura de computação em névoa foi realizada por Habibi *et al.* (2020) [Habibi et al. 2020], na qual os autores mostram que existem alguns modelos de referência desenvolvidos para computação em névoa a partir de várias perspectivas. A *MFog* foi concebida a partir de elementos comuns de implantações de arquiteturas analisadas da literatura. Para ganhar resiliência contra problemas de conexão foram acrescidos elementos e mecanismos que estão detalhados na Figura 1, onde o elemento principal é o acionador que com base em métricas de desempenho toma ações para garantia do funcionamento das aplicações.

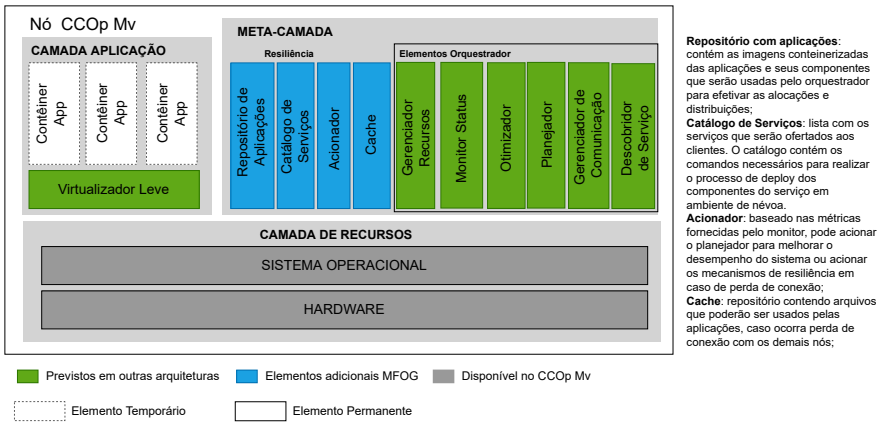

Figura 1. Modelo proposto para arquitetura *MFOG*.

Em relação a problemas de desconexão, o mecanismo de recuperação da arquitetura *MFog* atua para instanciar a aplicação no nó desconectado, e reestabelecer seu funcionamento para os clientes desse ponto de acesso. O mecanismo possibilita a ativação de componentes da aplicação conteinerizados. No entanto, componentes de armazenamento de arquivos persistentes não podem ser migrados (devido ao tamanho do repositório). Nesse caso, a *MFog* passa a utilizar o conteúdo do *cache* contendo parte do componente de armazenamento, formado por arquivos acessados mais recentemente. Caso a aplicação esteja distribuída pelos nós da névoa, e um desses nós tenha algum problema de falta de

recursos, o mecanismo de resiliência faz com que ocorra uma redistribuição nos outros nós não afetados da rede, de forma que o sistema também continue em funcionamento para os usuários.

Essa arquitetura pode ser aplicada, por exemplo, ao projeto CCOp Mv (Centro de Coordenação de Operações Móvel) é um projeto do Exército Brasileiro que integra o Programa Proteger[1]. Ele é composto por um conjunto de viaturas com equipamentos de comunicação que fornecem serviços de TIC para o comando e elementos da tropa, possibilitando a operação com recursos computacionais próximos a locais de eventos críticos. A Figura 2 apresenta um exemplo de equipamentos utilizadas neste contexto.

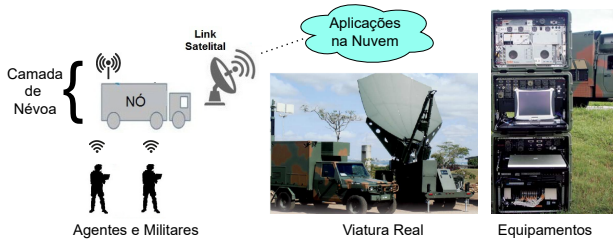

Figura 2. Viaturas do CCOp Mv.

3. Testes com a Arquitetura Proposta

Para verificar o funcionamento do mecanismo, um estudo de caso foi realizado em ambiente simulado. A aplicação em estudo foi a *Open Map Tile Server* que é um servidor de mapas e permite visualizar mapas e dados geográficos em computadores e celulares. Essa aplicação possui dois componentes, sendo o Componente 1 (C_1) o *front-end* da aplicação, e o Componente 2 (C_2) o repositório de dados geográficos (que exige capacidade de armazenamento persistente). A validação do mecanismo de resiliência foi feita com base em um experimento que objetivou verificar a disponibilidade da aplicação para o cliente. As métricas analisadas foram o tempo e o tipo de resposta de requisições feitas para a aplicação (que indicam o desempenho e a disponibilidade do sistema, respectivamente).

A estrutura foi construída usando aplicações *open-source*. Assim, para o virtualizador foi utilizado o *Docker Engine* e os componentes da aplicação foram obtidos através do repositório *Docker Hub*. Para o DNS foi utilizada a aplicação *Bind DNS* e criado um *script* em *python* que faz o papel de Monitor de Recursos / Acionador e também alimenta o *Cache* com arquivos consumidos pela aplicação. O *Proxy* foi implementado utilizado a aplicação *Squid*.

4. Resultados

O resultado dos testes em relação ao funcionamento do mecanismo proposto, que faz a migração da aplicação da nuvem para a névoa, no caso de problemas com a conexão, está apresentado na Figura 3. No gráfico é possível observar que a partir do momento que a capacidade do *link* é reduzida de 8.192 Kbps para 4.096 Kbps, o tempo de resposta médio (média para 10 requisições) aumenta e fica acima da latência limite (LL), que para

[1] `http://www.epex.eb.mil.br/index.php/proteger`

este estudo de caso foi fixada em 3.000 ms. Após 10 segundos na situação de tempo de resposta maior do que LL, o mecanismo de migração é acionado de tal forma que ele cria uma instância da aplicação no nó da névoa, muda a URL da aplicação para que aponte para o nó e, por fim, reinicia o *Proxy* para que as configurações sejam atualizadas. Durante este processo, os usuários ficam cerca de 10s sem acesso a aplicação.

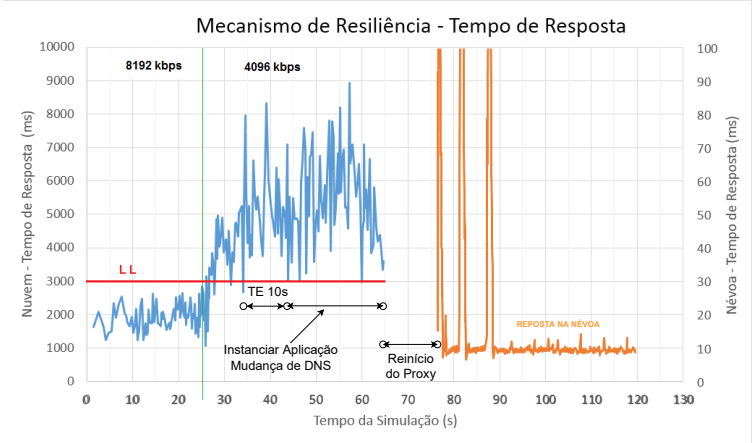

Figura 3. Funcionamento do mecanismo proposto.

5. Conclusão

Este artigo apresentou a arquitetura *MFog*, que é uma proposta de arquitetura resiliente para a implantação de aplicações em estruturas computacionais localizadas próximas ao usuário final, mas com limitação em recursos de conexão com a Internet e recursos de hardware. Os nós destas estruturas além de ser um elemento de um sistema distribuído, também são, por vezes, os únicos pontos de acesso de diversos usuários. Assim sendo, a arquitetura *MFog* pretende ser uma solução plenamente resiliente, com foco em resiliência orientado a eficiência e capacidade, mantendo desta forma a disponibilidade das aplicações para o usuário final. Os resultados também indicaram que um mecanismo de resiliência contra problemas de conexão, que migra a aplicação para o nó isolado, pode fornecer uma garantia de funcionamento da aplicação mesmo diante de uma perda total de conexão por este nó. Em trabalhos futuros serão realizados estudos em relação à migração e ao sincronismo de arquivos.

Referências

Habibi, P., Farhoudi, M., Kazemian, S., Khorsandi, S., and Leon-Garcia, A. (2020). Fog computing: a comprehensive architectural survey. *IEEE Access*, 8:69105–69133.

Prokhorenko, V. and Babar, M. A. (2020). Architectural resilience in cloud, fog and edge systems: A survey. *IEEE Access*, 8:28078–28095.

Rosário, D., Schimuneck, M., Camargo, J., Nobre, J., Both, C., Rochol, J., and Gerla, M. (2018). Service migration from cloud to multi-tier fog nodes for multimedia dissemination with qoe support. *Sensors*, 18(2):329.

Santos, J., Wauters, T., Volckaert, B., and De Turck, F. (2019). Towards network-aware resource provisioning in kubernetes for fog computing applications. In *2019 IEEE Conference on Network Softwarization (NetSoft)*, pages 351–359. IEEE.