

# ORQUESTRAÇÃO DE APLICAÇÕES DE COMPUTAÇÃO DE ALTA PERFORMANCE EM AMBIENTES CLOUD CONTEINERIZADOS

Lucas M. Varella<sup>1</sup>, Patricia D. Plentz<sup>1</sup>, Hugo M. Watanuki<sup>2</sup>, Artur Baruchi<sup>2</sup>

<sup>1</sup>Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)

Caixa Postal 5040 – 88.040-970 – Florianópolis – SC – Brazil

<sup>2</sup>LexisNexis Risk Solutions

lucas.varella@grad.ufsc.br, patricia.plentz@ufsc.br,  
hugo.watanuki@lexisnexisrisk.com, artur.baruchi@lexisnexisrisk.com

**Resumo.** *Este trabalho explora a orquestração da plataforma HPCC Systems (High Performance Computing Cluster) em ambientes cloud containerizados, com a ferramenta de orquestração Kubernetes. O objetivo do trabalho é avaliar as características, benefícios e desafios da implantação da plataforma HPCC Systems nesse paradigma através de diferentes provedores de cloud pública, especificamente Amazon Web Service (AWS) e Microsoft Azure. Os resultados preliminares sugerem que o paradigma de orquestração traz diversos benefícios para a plataforma em questão, mas suposições estritas sobre persistência de armazenamento de dados e recursos compartilhados específicos ao host, entre outras condições, geram desafios ao tentar levar a tecnologia a este ambiente.*

## 1. Introdução

A tecnologia de contêiner é uma metodologia de empacotamento de aplicações para que possam ser executadas/disponibilizadas com o seu conjunto de dependências de maneira isolada e eficiente. Contêineres têm emergido como uma escolha popular para desenvolvimento de aplicações na nuvem, recentemente mostrando-se uma melhor opção sobre ambientes convencionais, incluindo máquinas virtuais. Com gerenciamento de recursos e isolamento mais simples, maior eficiência, portabilidade, e uma boa disponibilidade em geral, a possibilidade de escolher containerização como paradigma de arquitetura torna-se mais atraente [Ramon-Cortes 2018]. Especificamente no escopo de computação na nuvem, o ambiente containerizado se destaca com seu gerenciamento de recursos eficiente, tendo em vista sua integração com orquestradores e escalonadores cloud [Beltre 2019].

No contexto de contêineres, o empacotamento de uma aplicação, com suas dependências, para uma determinada plataforma (hardware e software) realiza uma virtualização em nível de sistema operacional, contendo componentes e dependências da aplicação em si. Uma aplicação pode ter uma ou mais virtualizações disponíveis, a depender do número de plataformas alvo. As ferramentas de orquestração de contêineres possibilitam implantar suas virtualizações, com mecanismos de gerenciamento e lançamento de contêineres como *clusters*, ou *pods*. No caso deste trabalho, a ferramenta de orquestração escolhida foi Kubernetes [Kubernetes 2020], por ser uma das ferramentas mais conhecidas, com mais documentação e referências de uso dentre os provedores cloud

disponíveis. Estas ferramentas também trabalham para garantir condições de ambiente de desenvolvimento como escalabilidade horizontal, tolerância a falhas, entre outras, além de permitir escalonamento e automação flexíveis.

O paradigma de computação na nuvem em geral também tem crescido em uso, aplicações, e principalmente disponibilidade, com usuários finais podendo acessar recursos de armazenamento e até infraestrutura por preços acessíveis, ou até gratuitamente. Apesar desta grande adoção, porém, computação na nuvem ainda pode apresentar lacunas de funcionalidade que tornam difícil entregar o verdadeiro potencial delas para as aplicações alvo. Um dos principais problemas é a falta de compatibilidade e padronização entre as soluções nos diferentes provedores de cloud, e, portanto, uma avaliação de como uma mesma aplicação ou implementação se comporta através de provedores diferentes pode ser útil não só àquela aplicação específica, mas também para prover percepções da extensão do trabalho necessário para haver elasticidade dinâmica entre tecnologias cloud, para outras aplicações similares [Caballer 2018].

Este trabalho tem principal foco no comportamento da plataforma HPCC Systems (*High Performance Computing Cluster*) [HPCC Systems 2020], desenvolvida pela empresa LexisNexis Risk Solutions. É uma plataforma de sistema de computação que incorpora uma arquitetura de software implementada em clusters de computadores de baixo custo (*commodity*), para prover processamento paralelo de dados em alta performance para aplicações utilizando *Big Data*. No caso, foi investigado o comportamento da implementação da plataforma neste paradigma de orquestração de contêineres, não sendo necessário explorar especificamente o funcionamento interno de cada componente da plataforma, mas sim a comunicação entre eles em um escopo de contêineres.

Foi estudado o comportamento funcional e de desempenho em comparação entre diferentes provedores de cloud pública, e também a implementação em máquinas físicas (*bare metal*). Especificamente, foram explorados os serviços de computação em nuvem Microsoft Azure [Microsoft 2020] e Amazon Web Service (AWS) [Amazon 2020], ambos sendo operados com orquestração via Kubernetes, via os respectivos Azure Kubernetes Service (AKS) e o Elastic Kubernetes Service (EKS) da Amazon.

## **2. Objetivos**

### **2.1. Objetivo Geral**

O objetivo principal deste trabalho é avaliar as características, benefícios e desafios de implementações da plataforma HPCC Systems em ambientes cloud conteneirizados, através de diferentes provedores de cloud pública, especificamente Amazon Web Service (AWS) e Microsoft Azure e suas respectivas soluções para orquestração de contêineres via Kubernetes. Para isso, foram estudadas as configurações e tecnologias que buscam solucionar os problemas ou desafios de interação entre a plataforma e um ambiente conteneirizado, e também sendo aplicados diferentes testes funcionais e de desempenho para comparar as soluções. Além disso, durante o trabalho, algumas atividades foram realizadas para auxiliar o time de desenvolvimento da plataforma HPCC Systems. Como resultado destas atividades, é esperado que tanto os membros da equipe em questão quanto a comunidade acadêmica obtenham melhoria nos processos de implementação e configuração de ambientes de computação de alto desempenho em paradigmas de orquestração de contêineres na nuvem.

## 2.2 Objetivos Específicos

Os objetivos específicos necessários para alcançar o objetivo geral são os seguintes:

- Estudar todas as etapas necessárias para obter uma implementação em execução da plataforma HPCC Systems, em cada uma das diferentes infraestruturas de nuvem e *bare-metal*;
- Executar conjuntos de testes de desempenho sobre as diferentes opções de infraestrutura / provedor, além de sobre as diferentes configurações possíveis de cada opção;
- Identificar as diferentes soluções que cada ambiente (Azure, AWS, e *bare-metal*) traz para questões de armazenamento persistente, balanceamento de carga, comunicação de rede interna ou externa, entre outros fatores a serem explorados no trabalho.
- Realizar uma análise comparativa das diferentes funcionalidades previamente identificadas para cada ambiente de implementação;
- Executar e concluir as atividades propostas pelo time de desenvolvimento da plataforma HPCC, entregando os artefatos requisitados e buscando agregar o conhecimento adquirido a partir da execução destas, à este trabalho.

## 3. Resultados preliminares

Foi possível implementar a plataforma HPCC nos ambientes cloud providos por Amazon Web Services, e Microsoft Azure. Foram identificadas peculiaridades funcionais entre as aplicações. Com o ambiente Azure, o gerenciamento e alocação de armazenamento persistente é mais intuitivo, podendo ser atualizado em tempo de execução da aplicação, permitindo maior flexibilidade para carregamento e utilização dos dados da aplicação. Já com AWS, a parte de armazenamento foi um desafio maior, enquanto que a configuração de balanceamento de carga foi mais facilmente alcançada. Também no quesito de configurações, a utilização de Helm Charts [Helm 2020] foi bem vantajosa para alcançar uma implementação inicial fácil e intuitiva, para ambos os cenários citados. Um dos resultados esperados do trabalho será arquivos de configuração *helm* que resolvam os problemas encontrados no seu desenvolvimento, provendo à trabalhos futuros ou usuários da plataforma maior facilidade de implementação no ambiente cloud.

No caso de provisionamento dinâmico de armazenamento, com Azure existe o *azurefile*, uma classe de armazenamento nativa do Azure que é selecionada na configuração do *helm chart* da implementação, permitindo modificação de tamanho máximo alocado em tempo de execução, já resolvendo, de forma automática, questões de persistência de dados. Com Amazon Web Services, não há uma solução nativa como *azurefile*. Foi utilizado o serviço EFS (*Elastic File System*), um sistema de arquivos NFS elástico, escalável, e completamente gerenciável provido pela AWS. A interface de armazenamento de contêiner EFS (EFS CSI) permite clusters Kubernetes em execução em AWS gerenciar o ciclo de vida do sistema de arquivos EFS, utilizando também o *efs-provisioner*, um contêiner que tem acesso aos recursos AWS EFS, permitindo alocar armazenamento EFS em *PersistentVolumes* em Kubernetes. Ainda é necessária a criação do serviço e servidor EFS durante a implementação da plataforma, com esse gerenciamento adicional demonstrando maior facilidade de solucionar a questão de armazenamento dinâmico na implementação Azure.

Sobre acesso externo dinâmico, como balanceamento de carga e visibilidade externa dos serviços do cluster, existem variações de uso do *ingress controller* do Kubernetes, sendo este um serviço que gerencia acesso externo assim como balanceamento de carga. Para Azure, pode ser usado o *NGINX ingress controller*, instalado via Helm, que através de configuração em arquivo YAML, permite expor as portas de acesso de determinados IPs do cluster à um IP externo. Com AWS, o usuário a implementar cria um *Ingress* Kubernetes, um balanceador de carga de aplicativo AWS (*AWS Application Load Balancer*, ALB) é provisionado, o que significa que o gerenciamento de acesso externo é feito em Kubernetes, e somente o balanceamento de carga será gerenciado com AWS, com o *AWS Load Balancer Controller*, controlador de balanceamento de carga, que fornece um AWS ALB, como citado anteriormente, ou AWS NLB (Balanceador de carga de Rede, *Network Load Balancer*), quando é criado um serviço Kubernetes do tipo *LoadBalancer*. No caso, estas questões mostram as diferenças entre os provedores Azure e AWS para a mesma solução, e não foi identificada maior facilidade de uso ou implementação para determinado caso.

#### 4. Referências

HPCC Systems (2020). HPCC Systems, disponível em: <https://hpccsystems.com/>

Microsoft (2020). Microsoft Azure, disponível em: <https://azure.microsoft.com/>

Amazon (2020). Amazon Web Service, disponível em: <https://aws.amazon.com/>

Kubernetes (2020). Kubernetes, disponível em: <https://kubernetes.io/>

Beltre, A.(2019). “Enabling HPC workloads on Cloud Infrastructure using Kubernetes Container Orchestration Mechanisms”. In: 2019 IEEE/ACM Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIEHPC), Binghamton, NY.

Caballer, M. (2018). “Orchestrating Complex Application Architectures in Heterogeneous Clouds”. *Journal of Grid Computing*, 16, 3-18.V.

Ramon-Cortes, C.(2018) “Transparent Orchestration of Task-based Parallel Applications in Containers Platforms”. Springer Science+Business Media B. V., part of Springer Nature

Helm (2020). Helm Charts, disponível em: <https://helm.sh/docs/topics/charts/>