Reprodutibilidade de experimentos com uso de conteinerização

Gustavo Lopes Nomelini
Universidade Estadual do Oeste do Paraná
Cascavel, Brasil
gustavo.nomelini@unioeste.br

Guilherme Galante
Universidade Estadual do Oeste do Paraná
Cascavel, Brasil
guilherme.galante@unioeste.br

***Abstract*—Nowadays more areas of knowledge can take advantage of computational experiments to formulate and prove theories. Thus, the reproducibility of these experiments becomes a central requirement of the scientific process, as it allows validating or refuting statements made in a publication. In this context, this article discusses the issue of experimental reproducibility and proposes an approach based on Docker containers and the Spack tool, both open source projects, to ensure the replication of experiment execution environments.**

***Resumo*— Cada vez mais áreas do conhecimento podem tirar proveito de experimentos computacionais para formular e comprovar teorias. Assim, a reprodutibilidade destes experimentos passa a ser um requisito central do processo científico, uma vez que permite validar ou refutar as afirmações feitas em uma publicação. Neste contexto, no presente artigo discute-se a questão da reprodutibilidade de experimentos e propõe-se uma abordagem baseada em contêineres Docker e na ferramenta Spack, ambos projetos de código aberto, para garantir a replicação de ambientes de execução de experimentos.**

***Palavras-chave*—reprodutibilidade de experimentos; computação em nuvem; conteinerização.**

I. Introdução

A computação se estabeleceu como uma ciência fundamental para formular e comprovar teorias a partir de experimentos e simulações em ambientes computacionais, podendo aprimorar o desenvolvimento da ciência geral [1]. Com o rápido avanço da computação cada vez mais áreas do conhecimento podem tirar proveito de sistemas computacionais para formular e comprovar teorias com o uso de modelos computacionais e simulações [2]. Alguns dos problemas teóricos, que são analiticamente complexos de serem solucionados, podem ser resolvidos por meio de simulações e da utilização de recursos computacionais.

Nesse contexto, a reprodutibilidade passa a ser um requisito central do processo científico, uma vez que permite validar ou refutar as afirmações feitas em uma publicação por meio da replicação dos experimentos computacionais [3]. Em princípio, os experimentos computacionais deveriam ser facilmente reproduzidos, considerando que ao executar um programa usando as mesmas entradas em uma arquitetura equivalente, são esperados resultados equivalentes. Na prática, a complexidade do software e hardware de hoje torna difícil a reprodução dada a diversidade desses elementos.

É comum que aplicações computacionais possuam dependências em relação ao sistema operacional, binários e bibliotecas, exigindo uma compatibilidade específica entre desenvolvimento, teste e produção dos resultados. Esse problema foi ressaltado pela revista Nature já em 2016 [4] e poderá ser agravado ao longo do tempo com o surgimento de novas ferramentas, bibliotecas e sistemas operacionais (ou novas versões incompatíveis).

Nesse sentido, a tecnologia de contêineres pode ser empregada na reprodução de ambientes de execução, possibilitando o encapsulamento de um sistema computacional, contendo seu sistema operacional, ferramentas, bibliotecas e tudo mais que for necessário. Esse encapsulamento pode ser feito de várias formas gerando uma imagem capaz de ser replicada em outras máquinas físicas ou virtuais. Tais imagens podem ser disponibilizadas em repositórios públicos, em serviços de nuvem ou juntamente com a publicação do artigo, tornando os resultados passíveis de reprodução por terceiros a qualquer instante. Os contêineres podem ser combinados a ferramentas de gerenciamento de pacotes que auxiliam o cientista a gerenciar as dependências de software.

Assim, o objetivo deste trabalho é propor um método utilizando as ferramentas Docker e Spack que possibilite a reprodutibilidade de experimentos com as necessidades atuais. Além disso, propõe-se um fluxo para a construção do ambiente a ser reproduzido. Por fim, o método é comparado com plataformas já existentes que permitem o compartilhamento e reprodução de experimentos até certo nível, como o *DropBox* e o *Github*.

II. Reprodutibilidade de experimentos

Um experimento computacional pode ser classificado em diferentes níveis de reprodutibilidade. Tal classificação pode ser feita pelo modelo PRIMAD [5], que especifica a reprodutibilidade em vários componentes: Plataforma, Meta de pesquisa (*Research Goal*), Implementação, Método, Ator e Dados. Um exemplo é apresentado a seguir:

- Plataforma: gcc 5.4, Ubuntu 18.04.

- *Research Goal*: aplicar um algoritmo de busca eficiente.
- Implementação: utilizando C++.
- Método: Quicksort.
- Ator*:* o usuário que executar o algoritmo do experimento.
- Dados:
    - Entrada: dados a serem analisados na busca.
    - Parâmetro: posição do pivô.

Segundo esse modelo, a reprodutibilidade pode vir de várias formas combinando as diferentes variáveis da sigla, e isto permite que conhecimentos diferentes sejam obtidos dependendo do tipo de reprodutibilidade. No estudo realizado, o objetivo foi fixar as variáveis *Plataforma*, *Implementação* e *Dados*, e segundo o modelo, esse estudo é classificado como PR'IM'A'D, onde as siglas com apóstrofe indicam que são variáveis independentes enquanto as outras são fixadas. Pelo modelo PRIMAD, mudar a meta da pesquisa permite verificar novas hipóteses e dar ressignificação para um experimento, enquanto mudar o método permite validar uma mesma hipótese utilizando outra metodologia e alterar o ator é uma mudança ortogonal com todas as outras.

Atualmente, existem alguns projetos em andamento, incluindo softwares de código aberto, que estudam a reprodutibilidade de experimentos como o ReproZip [6], o NextFlow [7] e a plataforma OCCAM [8]. Neste trabalho, busca-se garantir a reprodução de plataforma, implementação e dados por meio do uso de contêineres Docker aliados à ferramentas de gerenciamento de pacotes Spack. Detalhes são apresentados na Seção III.

III. Reprodução de experimentos utilizando os softwares Docker e Spack

Para garantir a reprodução da plataforma, implementação e dos dados, neste trabalho utilizou-se uma abordagem baseada em conteinêres Docker utilizando o gerenciador de pacotes Spack.

A conteinerização é um tipo de virtualização em nível de sistema operacional. Ela permite criar múltiplas instâncias dentro de um único sistema operacional, dependendo de um único *Kernel Space*, sem depender de uma camada complexa de virtualização com um *Hypervisor* e sistema operacional próprio, como ocorre nas máquinas virtuais [9], essa diferença está ilustrada na Figura 1.

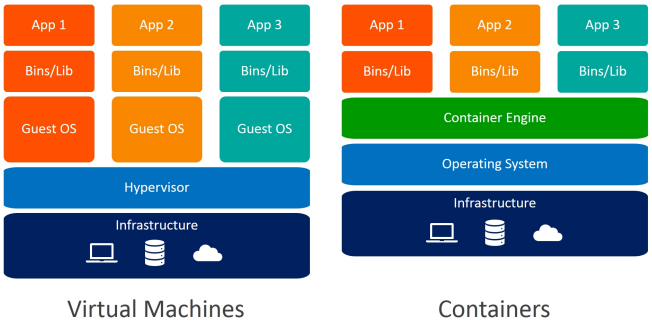

Fig 1. Diferença entre a organização de máquinas virtuais e conteinêres.

Assim sendo, os contêineres ocupam um espaço de memória bem reduzido. Eles são capazes de armazenar um microsserviço ou aplicação, e todas suas dependências como bibliotecas e programas secundários. Outros benefícios dos contêineres são a agilidade de desenvolvimento, a flexibilidade, a segurança, a portabilidade e a escalabilidade deles.

O Docker é um conjunto de produtos de plataforma como serviço (*PaaS*), capaz de fazer a conteinerização de aplicações a partir da virtualização em nível de sistema operacional. Apesar da possibilidade de se criar contêineres sem o Docker, esta ferramenta facilita muito o desenvolvimento, lançamento e manutenção dos contêineres, por isso foi utilizada no estudo realizado.

O software *Spack*, também é um projeto de código aberto, que consiste em um gerenciador de pacotes para supercomputadores e Linux [10]. Esse gerenciador facilita a reprodução científica pela possibilidade de se criar um Ambiente Spack no qual podem coexistir diversas bibliotecas, compiladores, configurações e versões. Esse ambiente pode ser replicado em várias plataformas de forma flexível e uma mesma máquina pode suportar vários ambientes. É uma ferramenta especialmente útil quando os testes a serem realizados servem justamente para comparar desempenho e funcionalidade entre diferentes bibliotecas e versões. Além disso, esses ambientes são reprodutíveis em espaço de usuário, o que é um fator interessante para ambientes onde os usuários possuem restrições na configuração do sistema (não root), como supercomputadores e clusters.

IV. Processo da criação de um container docker com uso da ferramenta Spack

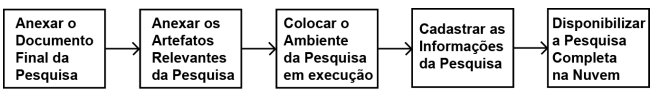

Fig. 2. Fluxo de construção de um ambiente reprodutível.

O modelo de arquitetura proposto deve seguir o fluxo de processo representado na Figura 2 [11], de forma a contemplar as siglas P, I e D do modelo PRIMAD. Primeiramente, o método deve ser capaz de anexar os documentos e a seguir anexar os artefatos, bibliotecas e

ferramentas, relevantes para a reprodução da pesquisa. Após criar um ambiente computacional com o documento final e os artefatos da pesquisa, é necessário colocá-lo em execução. Posteriormente, o método deve possibilitar o cadastro de informações na pesquisa, permitindo que o pesquisador complete as informações de pesquisa. Como última etapa, a arquitetura deve lidar com a disponibilização da pesquisa completa na nuvem de forma prática.

É possível instalar todos os artefatos via Spack, os quais ficam registrados no arquivo packages.yaml. Caso seja necessário uma biblioteca que não esteja no banco de dados padrão é possível compilar localmente para o Spack novos pacotes com o uso de um script Python. Após instalar os pacotes é possível criar um ambiente Spack, que ao ser ativado, possibilita carregar todas as bibliotecas que serão necessárias para executar uma aplicação. De maneira prática pode-se gerar vários ambientes com diferentes configurações, compiladores e versões de bibliotecas. Esses ambientes podem ser replicados em outras máquinas garantindo a reprodutibilidade de testes, isso pode ser feito a partir dos arquivos .yaml e .lock do diretório do ambiente.

Além disso, também é possível utilizar a virtualização em nível sistema operacional a partir de um comando do próprio Spack. Isto se torna útil no caso da reprodutibilidade de experimentos por reproduzir o ambiente para experimentação de forma completa, incluindo um sistema operacional. A ferramenta permite, a partir de comando dentro do diretório do ambiente, gerar um "*Dockerfile*" que é um arquivo texto com todas as instruções, binários e dependências da aplicação que será inserida em um container. Neste caso, é possível escolher o sistema operacional que servirá de ambiente pro experimento com todas as bibliotecas necessárias obtidas via Spack. Por enquanto apenas duas versões do Ubuntu e duas versões do CentOS têm suporte e compatibilidade com a ferramenta [12].

A partir desse arquivo é possível gerar uma imagem Docker que pode ser utilizada pelos desenvolvedores para replicar uma aplicação. Uma imagem para ser ativada deve ser inserida dentro de um Container Docker. Essas imagens Docker podem ser então compartilhadas entre os pesquisadores, inclusive de forma nativa por comando utilizando o registro padrão chamado "*Docker Hub*" onde ficam registradas essas imagens de forma privada ou pública.

V. Resultados e Discussões

Nesta seção realiza-se uma análise do método proposto e sua comparação com plataformas para publicação existentes. Para estudo de caso foi comparada a efetividade da reprodutibilidade de experimentos entre o uso do Docker mais o Spack com métodos diferentes utilizando duas plataformas: o *Dropbox* que é um serviço de armazenamento pessoal em nuvem, e o *Github* que é uma plataforma de hospedagem de código-fonte para projetos.

Para isto, foi aplicado o fluxograma representado na Figura 2 e foi analisado se as plataformas, bem como o método proposto, são capazes de fornecer um ambiente replicável. Primeiramente, em relação ao anexo do documento final da pesquisa, todas as plataformas analisadas forneceram suporte. A segunda questão que analisou a possibilidade de anexar as ferramentas necessárias para o experimento também foi contemplada pelos três métodos. Como terceira análise, foi mostrado que as três alternativas são capazes de colocar o ambiente de pesquisa em execução. O quarto passo que analisou a possibilidade de cadastrar as informações da pesquisa também foi apreciado por todos os métodos. A seguir, como último passo do fluxo proposto, foi analisada a possibilidade de disponibilizar a pesquisa completa e seus resultados na nuvem, esta etapa também foi contemplada pelas três plataformas.

No entanto, apesar dos outros métodos contemplarem o fluxograma para construção de um ambiente reprodutível, a solução proposta vai além e é capaz de fornecer um ambiente compatível e pronto para o uso do usuário que deseja replicar o experimento. Enquanto que no Dropbox ou Github o usuário que deseja executar o experimento necessita baixar o conteúdo, garantir as dependências, compilar e configurar a aplicação, ao se utilizar contêineres, a plataforma, implementação e dados já estão contemplados e prontos para uso. Então, a execução do experimento pode ser feita diretamente na máquina do usuário ou até mesmo utilizando os recursos computacionais da nuvem. Assim sendo, o método adotado oferece um ambiente de experiência completo e replicável.

Na última etapa, avaliou-se a possibilidade de executar o ambiente de pesquisa diretamente na nuvem, isto é, realizar os experimentos utilizando os serviços na nuvem para processar os dados. Neste quesito, apenas a abordagem da arquitetura proposta utilizando o Docker e o Spack foi capaz de executar o experimento de forma nativa utilizando a nuvem. Dessa forma, a arquitetura permite que o pesquisador transfira seu ambiente de pesquisa físico para um ambiente na nuvem. Portanto, o método possibilita a replicação de experimentos que exigem grande poder computacional sem a necessidade do pesquisador possuir a infraestrutura física necessária, podendo alocar os recursos necessários de provedores de nuvem, tais como, AWS, Google e Azure, que oferecem serviços relacionados de contêineres. A Tabela I sintetiza os resultados da comparação.

TABELA I

COMPARAÇÃO ENTRE A SOLUÇÃO PROPOSTA E OUTRAS SOLUÇÕES UTILIZADAS PARA A REPRODUÇÃO DE EXPERIMENTOS

Análise realizada	**Dropbox**	**Github**	**Docker+ Spack**

Permite anexo do documento final da pesquisa	Sim	Sim	Sim
Permite anexo das ferramentas necessárias para o experimento	Sim	Sim	Sim
Permite colocar o ambiente de pesquisa em execução	Sim	Sim	Sim
Permite cadastrar as informações da pesquisa	Sim	Sim	Sim
Permite a disponibilização completa na nuvem	Sim	Sim	Sim
Permite executar o ambiente da pesquisa direto na nuvem	Não	Não	Sim

VI. Conclusão

O objetivo deste trabalho foi apresentar uma abordagem baseada em contêineres Docker e na ferramenta Spack para garantir a replicação de ambientes para execução de experimentos científicos computacionais.

Foi possível constatar a praticidade e efetividade da solução proposta, uma vez que é capaz de fornecer suporte para a construção de um ambiente reprodutível. Todo o processo é suportado de maneira nativa pelas ferramentas, contemplando o fluxo proposto pelo método e garantindo a replicação da plataforma, implementação e os dados, como descrito pelo modelo PRIMAD.

Isso possibilita que experimentos possam ser reproduzidos por pesquisadores terceiros, atestando os resultados apresentados em uma publicação e auxiliando pesquisas relacionadas que podem se beneficiar de resultados já obtidos anteriormente por outros pesquisadores.

Ainda, foi realizado um estudo de caso comparando o método proposto com alternativas presentes, demonstrando que o uso do Docker com o Spack possui um diferencial por ser capaz de reproduzir um ambiente de experimento completo que pode ser executado direto na nuvem de forma nativa, considerando que os principais provedores de nuvem já oferecem serviços desta natureza.

Agradecimentos

Agradecimentos ao Programa de Ensino Tutorial (PET) do Ministério da Educação (MEC/SESU) pelo auxílio concedido na forma de bolsa.

Referências

[1] L. Badger, R. Patt-Corner e J. Voas, “Draft Cloud Computing Synopsis and Recommendations of the National Institute of Standards and Technology”, *Special Publication 800-146 NIST, May 2011*.

[2] H. Bossel, “Modeling and Simulation”. *CRC Press,* 1994. DOI:10.1201/9781315275574.

[3] P. Ivie e D. Thain. “Reproducibility in Scientific Computing”. ACM Comput. Surv. 51, 3, Article 63, 2018. DOI:10.1145/3186266.

[4] Nature, “Reality Check on Reproducibility”. Nature. 2016 May 26; 533(7604):437. DOI: 10.1038/533437a.

[5] J. Freire, N. Fuhr e A. Rauber, “Reproducibility of Data-Oriented Experiments in E-Science”, Dagstuhl Reports, Vol. 6, Issue 1, pp. 108–159, *Schloss Dagstuhlleibniz-Zentrum Fuer Informatik*, 2016.

[6] F. Chirigati, D. Shasha e J. Freire, “Reprozip: Using provenance to support computational reproducibility”, in *5th USENIX Workshop on the Theory and Practice of Provenance (TaPP 13)*, 2013.

[7] P. Di Tommaso, M. Chatzou e E. Floden, “Nextflow enables reproducible computational workflows”. Nature Biotechnology 35, 316–319, 2017. DOI: 10.1038/nbt.3820.

[8] OCCAM. Disponível em: https://occam.cs.pitt.edu/.

[9] Red Hat. “Containers x Máquinas Virtuais”. Disponível em: https://www.redhat.com/pt-br/ topics/containers/ containers-vs-vms. Acesso em: 20 de set. de 2021.

[10] Spack. “About Spack”. Disponível em: https://spack.io/about/. Acesso em: 20 de set. de 2021.

[11] Carvalho, Souza, “Um Modelo para Disponibilização de Pesquisas Computacionais e seus Artefatos em Contêineres de Software em Nuvem”. Dissertação de Mestrado. UFPE, 2017.

[12] Spack. “Container images”. Disponível em: https://spack.readthedocs.io/en/latest/containers.html . Acesso em: 20 de set.. de 2021.