CONTAINER-DRIVEN REPRODUCIBLE RESEARCH MADE SIMPLE

Ronaldas Paulius Lencevicius, Sang-Yun Oh
Statistics & Applied Probability
University of California
Santa Barbara, CA 93106-3110
{ronaldas, sang}@ucsb.edu

Scholarship in data science requires a complete software development environment along with instructions for all the results and figures [1]. However, fully specifying and reproducing an arbitrary data science workflow can often be a challenging task, especially with the increasing complexity of software configurations and computational infrastructure. Reproducibility that depends on documentation can involve a lot of specialized adjustments and tweaking that not all researchers may have the background or especially the time for. In contrast, we propose a computational research framework that can specify complex computational environments by using an OS-level virtualization technology called containers. Our container-driven reproducibility approach balances flexibility and ease of use through Visual Studio Code, a popular code editor.

Containerization is a form of software packaging that contains operating system level finetuning while requiring only the containerization software to run. This is different from virtual machines in the sense that containers are much more lightweight and do not require emulation of system hardware. Containers are built and run locally on a laptop or remotely in a high performance computing environment. Usually, containers are accessed through the command line which can be a barrier to entry to users. To alleviate the complexity of using container software through the command line, we make use of Visual Studio Code which has extensions to help navigate the container running process as well as managing remote servers through a graphical user interface.

With regards to reproducibility of containers, the entire project configuration lives on 2 files: Dockerfile and devcontainer.json. The Dockerfile defines the container image including the operating system and any necessary software configuration while the devcontainer.json defines the development tooling and extensions on Visual Studio Code. These containers can be deployed on a shared user system where users may be running code concurrently using a container management software called Podman, a drop-in replacement for the widely used Docker platform.

In order to facilitate this process, we provide a templated way of defining the container files that only requires the user to answer a set of questions on the requirements of their project. The users can then build their entire project with all the necessary software components and run it on top of any system (locally, remote server, friend's computer, etc.). This project is portable and easily transferable to other users as well. It is further customizable by adding to the container files which maintains the reproducibility of the overall setup. We have found that this framework provides a generic way to set up, reproduce, and distribute research with minimal interaction at the end of the researcher all while alleviating the need for hands-on reproduction of the research environment.

Acknowledgments

This work used Jetstream2 at Indiana University through allocation MTH230010 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

[1] Jonathan B Buckheit and David L Donoho. *Wavelab and reproducible research*. Springer, 1995.