Reproducibility Framework for Scientific Application in HPC

Amarjeet Sharma
Center for Development of Advance Computing, Pune, India
amarjeets@cdac.in

Anil Kumar Gupta
Center for Development of Advance Computing, Pune, India
anilg@cdac.in

***Abstract*— The poster proposes a framework/tool which will bundle the underline application in such a format that will provide all the compatible hardware/software. The framework enables to create reproducible application package which will be immune to host execution environment (software and hardware) upgrades.**

Keywords-reproducibility; cwl; reprzip;sciunit

I. INTRODUCTION

It is often difficult to reproduce the most straightforward computational steps [1] in an existing research work due to issues in the application package, installation, execution steps and continuously changing features of underline software/hardware. It is very problem of almost all computational domains because of having common characteristics of huge input data sets and complex computational requirements. This problem leads to many obstacles to ever-evolving computational research and analytical outputs. The research results are dependent on the execution environment and input data if both can be preserved then the output can be reproduced at a later point of time. A massive effort in the researcher's community is being made to ensure reproducibility [2] of the scientist work. A robust mechanism/framework needs to deployed for reproducible application development. The framework services will enable application reproducibility along with all its dependencies.

II. TECHNIQUES FOR REPRODUCIBILITY

Many methods and techniques had been proposed for reproducibility [3]. These techniques are described below in brief:

A. Automation Scripts:

Scientific software can be automated via text-based scripts. Common Workflow Language (CWL), Python, Perl are the most commonly used script for reproducibility. The scripts provide a trivial solution because it has no enough documentation explaining the commands that were used. The other major problem of the script is names of libraries, packages versions and its repositories address (URL) often changes which often fail to reproduce the application execution result.

B. Virtual Machines:

The virtual machine is another viable solution for computational reproducibility. It contains all the software/execution environment along with Operating Systems. There are frameworks build upon virtual machine like Reprzip [4]. The problem with the virtual machine is its startup time, scalability, resource utilization, performance and vast size incorporating excess packaging of the entire Operating System while using a small portion of it.

C. Containers:

The container is a recent advancement in virtualization technology. It makes ease in configuring and packaging the software/execution environment. It packages only necessary dependencies close to the software. It is independent of underlying host software and hardware changes. It is lighter than Virtual Machines. The problem with the container is its security, host operating system dependency and resource management.

There are frameworks (like SciUnit [5]) developed using all of the above techniques as hybrid approach.

III. PROPOSED FRAMEWORK

The framework consists of four parts namely Input Scripts (Input Script, Resource Script), Application Containerizer, Application Container Bundle, Launcher (Figure-1). A brief description of each of the component is given as:

A. ***Container Scripts:*** *The input scripts are standard json like script consisting of key-value pairs.*

B. ***Application Containerizer:*** *This module parses the user input-script to build the app with all the dependencies.*

C. ***Application Container Bundle:*** *It will create a compressed portable form of an image that will be used for reproducibility at compatible machines.*

D. ***Launcher:*** *It will load the image into the system/machine and launches the container to execute the application.*

IV. Working Mechanism

The proposed framework will create a portable bundle/image of the containerized application based on user input. Once the bundle is created, then it can be loaded on any compatible machine where it will produce a similar execution environment so that reproducibility of computational application result should be ensured. For clarity, please see Figure-2.

V. Conclusion

SciComRep would ease to develop reproducible applications. It will Speed-up development and debugging process and increase the usability/reproducibility of researchers work. The framework is still in the research/development phase. Its aim to provide an easy interface to package an application in a common standard format which is easy to reproduce the earlier work. The framework also intends to provide compiler directed checkpointing and restore features in the form of pragmas. The application developer would insert the pragmas inside the application to mark checkpoint location. Later it will be used to restore or debug the application. The framework can also be used for resource management, power optimization in the heterogeneous cluster using node consolidation technique.

References

[1] Ivie, P. and Thain, D., 2018. Reproducibility in Scientific Computing. ACM Computing Surveys (CSUR), 51(3), p.63.

[2] Wilson, G., Aruliah, D.A., Brown, C.T., Hong, N.P.C., Davis, M., Guy, R.T., Haddock, S.H., Huff, K.D., Mitchell, I.M., Plumbley, M.D. and Waugh, B., 2014. Best practices for scientific computing. PLoS biology, 12(1), p.e1001745.

[3] Begley, C.G. and Ioannidis, J.P., 2015. Reproducibility in science: improving the standard for basic and preclinical research. Circulation research, 116(1), pp.116-126.

[4] F. Chirigati, R. Rampin, D. Shasha, and J. Freire,2016. ReproZip: Computational Reproducibility With Ease, In Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data (SIGMOD), pp. 2085-2088, 2016

[5] D. That, G. Fils, Z. Yuan, T. Malik. Sciunits: Reusable Research Objects. In IEEE eScience, 2017

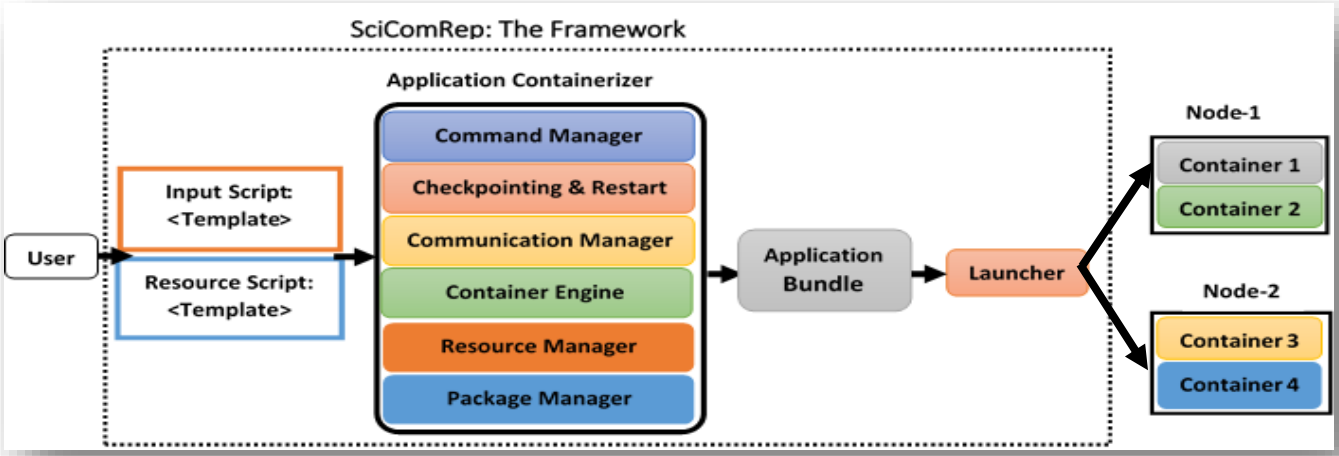

Figure-1

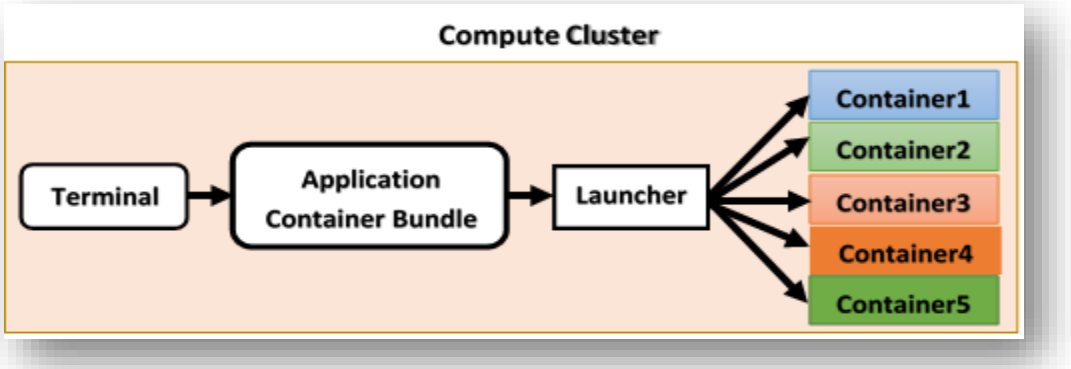

Figure-2