Running Workloads on the Prometheus Cluster with Singularity

Jan Kapała, Jan Meizner, Piotr Nowakowski, Patryk Wójtowicz, Marian Bubak

ACC Cyfronet AGH, AGH University of Science and Technology, Kraków, Poland

e-mails: ymkapala@cyf-kr.edu.pl, j.meizner@cyfronet.pl, p.nowakowski@cyfronet.pl, ymwojtow@cyf-kr.edu.pl, bubak@agh.edu.pl

Keywords: exascale, HPC, containers, Singularity

1. Introduction

In this paper we aim to describe a manageable yet efficient solution developed by us in the scope of the PROCESS [1][2] project, which aims to pave the way towards exascale systems. To this end, the project combines the resources of multiple HPC centers, including the Prometheus cluster at Cyfronet as well as CoolMUC and SuperMUC-NG clusters at Leibniz-Rechenzentrum (LRZ) augmented with other infrastructures, such as clouds.

In this paper we focus on the use of Singularity containers available on Prometheus in combination with the Interactive Execution Environment which helps schedule containerized applications as pipelines on HPC controlled via a web interface and REST API. We also briefly discuss integration with the LOBCDER component responsible for storing and moving data between infrastructures. The platform is depicted in Fig. 1.

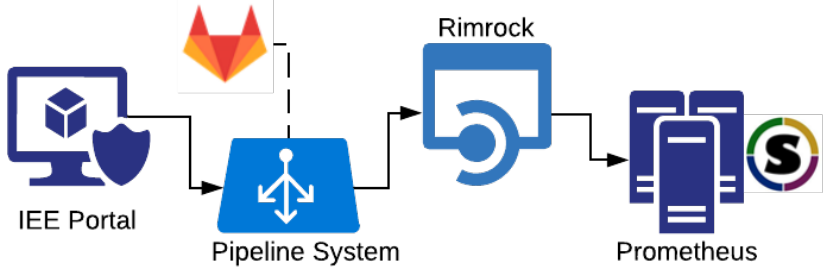

Fig. 1. Running PROCESS workloads on the Prometheus Cluster.

2. Description of the problem

The goal of PROCESS is to provide a platform for scientific domains experts and end users in a form which is ready for extreme scalability [3]. Given that no public exascale systems currently exist, reaching this goal cannot be directly verified; however, our aim is to validate it indirectly by ensuring good scalability than can be extrapolated to larger installations. The representative applications used in the PROCESS project are highly diverse and cover many use cases: medical, astronomical, financial and agricultural. Except for the last one, which needs to be run at LRZ (due to legal constraints), our goal is to run them on a wide range of sites, including Prometheus. However, due to the aforementioned heterogeneity, preparation of a runtime environment is non-trivial. An additional challenge is introduced by the need to support very specific software dependencies and hardware requirements (e.g. GPUs).

3. Proposed solution

HPC system administrators, including those responsible for Prometheus, provide solutions for most typical dependency issues through mechanisms such as modules which enable selection of required versions of tools or libraries from a wide range of pre-installed software. However, sometimes we need to go beyond this mechanism, especially in the case of legacy applications that have been developed under different OS distributions (e.g. Ubuntu rather than CentOS, which is installed on Prometheus). In this case preparation of the appropriate runtime environment for current and future versions of the application may not be feasible especially when the original authors cannot be directly reached. The simplest solution is therefore to allow the authors of an application to package it as an appliance.

One possibility involves the use of clouds, where the code may run as an independent VM composed of the kernel and the userland of our choosing. However, clouds introduce significant overhead and are not optimized for running large HPC tasks. A remedy is available in the form of the Singularity platform [4] enabling Docker-like containers to be run in a multi-tenant HPC environment. In the course of our work we were able to successfully prepare a platform based on the Interactive Execution Environment (web application implemented in Ruby on Rails) that enables scheduling such containers on Prometheus. It involves the standard mechanism used by the IEE to run jobs, as described in [5], with the addition of a so-called "Container Step" which deploys a container to HPC and invokes its embedded application services. It also enables using the GPU partition from inside the container.

4. Summary

The solution has been successfully integrated with the PLGrid Infrastructure [6] and deployed at Cyfronet.

For the needs of the PROCESS project we have secured the required resources on Prometheus through computational grants (for CPU/GPU): process1/process1gpu and process2/process2gpu respectively.

Acknowledgments. This work is supported by the "PROviding Computing solutions for ExaScale ChallengeS" (PROCESS) project that received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777533. This research was supported in part by PL-Grid Infrastructure. The authors also want to thank Maciej Czuchry, Łukasz Flis, Patryk Lasoń and Marek Magryś for help with making use of the HPC resources at Cyfronet.

References

1. PROCESS Project, https://www.process-project.eu/,
2. PROCES Project (DICE Team), http://dice.cyfronet.pl/projects/details/Process,
3. M. Bobak, A. S. Z. Belloum, P. Nowakowski, J. Meizner, M. Bubak, M. Heikkurinen, O. Habala, and L. Hluchý, "Exascale computing and data architectures for brownfield applications," in Fuzzy Systems and Knowledge Discovery (FSKD), 2018 14th International Conference on. IEEE, 2018, pp. 461-468,
4. Singularity containers: https://sylabs.io/singularity/,
5. M. Bubak, J. Meizner, P. Nowakowski, T. Gubała, M. Malawski, Towards a universal platform for simulations on Prometheus (submitted to KU KDM 2020),
6. PLGrid, http://www.plgrid.pl/.