A hybrid HPC and Cloud platform for multidisciplinary scientific application

Marian Bubak[1,2], Jan Meizner[1], Piotr Nowakowski[1,2], Martin Bobák[3], Ondrej Habala[3], Ladislav Hluchý[3], Viet Tran[3], Adam S. Z. Belloum[4], Reginald Cushing[4], Maximilian Höb[5], Dieter Kranzlmüller[5,6], Jan Schmidt[5]

[1] *ACC Cyfronet AGH, AGH University of Science and Technology, Krakow, Poland*
[2] *Sano, Centre for Computational Personalised Medicine, Krakow, Poland*
[3] *Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia*
[4] *Informatics Institute, University of Amsterdam, Amsterdam, Netherlands*
[5] *MNM-Team, Ludwig-Maximilians-Universität, Munich, Germany*
[6] *Leibniz Supercomputing Centre, Garching, Germany*

Introduction

In our previous publications on the PROCESS project we defined the major challenges and the basic architecture of the scalable storage and compute platform [1] for processing extremely large data sets. This architecture was later extended in our subsequent paper [2] that also presents details of five use cases, providing the means to validate the platform – particularly in its critical aspects such as scalability. These applications represent a wide range of scientific disciplines, including medicine, astronomy, disaster prevention, revenue analysis and agriculture, providing a solid basis for a comprehensive analysis of the proposed platform. In this paper we present work performed in the scope of the PROCESS project based on the abovementioned architecture and aiming to fulfill application requirements by providing a heterogenous, highly scalable platform composed of both High Performance Computing (HPC) and cloud resources. To this end, we focus on implementation aspects of the platform as well as the current results.

Implementation

The entry point to the platform is provided by the Interactive Execution Environment (IEE), shown in Figure 1. This component was implemented (using the well-established Ruby on Rails technology) as a convenient Web User Interface (UI) for the domain specialists. IEE is connected to various services enabling smooth integration with the relevant e-infrastructures such as the HPC layer via the Rimrock component[3] and computing clouds via the Cloudify extension[4]. It enables management of a wide range of infrastructures: those provisioned by scientific organizations – such as the Prometheus cluster at Cyfronet, or CoolMUC and SuperMUC-NG at Leibniz-Rechenzentrum (LRZ) – as well as private clouds at the Institute of Informatics Slovak Academy of Sciences (II SAS) and other compatible private and community clouds, including services provided in the scope of European Open Science Cloud (EOSC).

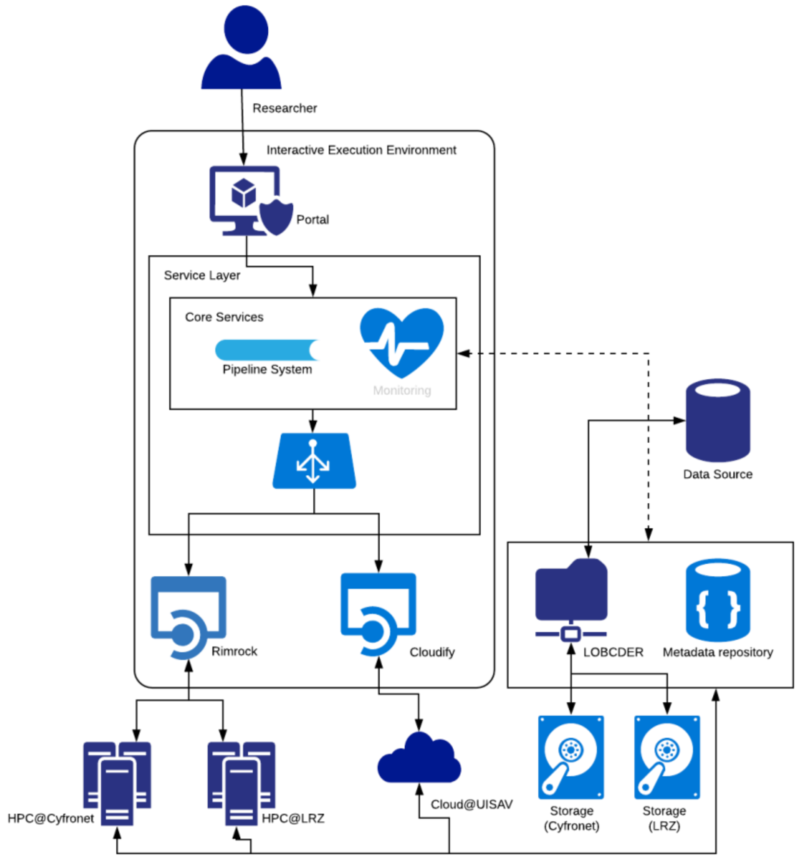

Figure 1: The PROCESS Interactive Execution Environment and the related components.

The platform furthermore enables orchestration of data transfers with the help of the Large OBject Cloud Data storagE fedeRation (LOBCDER) component[2] and the relevant scalable data micro-infrastructure, enabling data to be efficiently shifted between different geographical locations and technologies. This mechanism is required to address the challenges mentioned in our previous papers, such as the need to aggregate input data from diverse locations (e.g. radiotelescope arrays and temporary data repositories) and utilize the available geographically distributed resources in an efficient manner.

In addition to the core functionality described above, IEE interacts with a wide range of supplementary services, providing streamlined user experience for the domain specialist. This includes support for containers (enabling the whole application to be packaged as an appliance, ready to run on any e-Infrastructure without the need to install dedicated software tools or libraries). Currently, the platform provides support for three technologies, namely the Docker engine [5] for use in single-tenant environments such as the Cloud, Singularity [6], which is a major and well-established container solution properly adapted (security- and functionality-wise) for multi-tenant environments such as an HPC clusters, and Charliecloud [7], which offers lightweight user-defined software stacks for HPC environments based on standard Linux mechanisms such as namespaces. Additional features include a versioning mechanism for supplementary scripts via Git repositories and the ability to call external services when direct access to resource code is not possible (e.g. due to legal reasons) yet a public Application Programming Interface (API) is available.

Finally, IEE is also able to collect metrics which may be used to evaluate performance and scalability of the whole platform.

Results

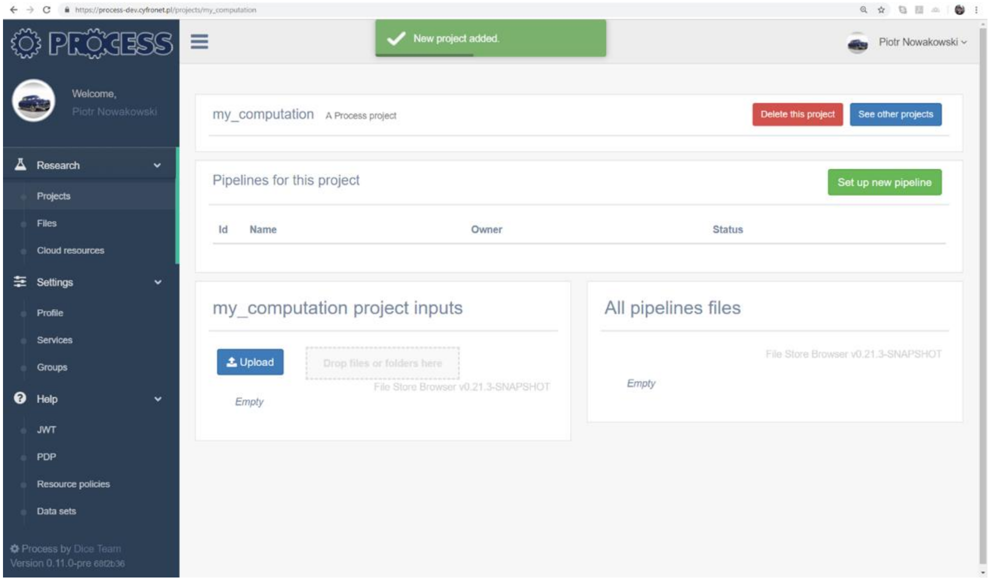

Figure 2 Setting up a new project in the IEE execution environment

Using the mechanisms described above we were able to deploy an IEE instance in Krakow and connect it to the PLGrid infrastructure [8] including an OpenID Single Sign-On (SSO) solution allowing smooth integration with HPC resources via the credential delegation mechanism. The IEE user interface shown in Figure 2 is based on the Model Execution Environment platform developed in the scope of the EurValve [9] project.

This integration was expanded to include cloud infrastructures (currently the one hosted in Bratislava at II SAS, with further extensions supported via the aforementioned Cloudify component) as well as other HPC resources such as the cluster in Amsterdam used for data storage, along with the CoolMUC computing cluster at LRZ. With help from LOBCDER, data can be freely moved between these infrastructures to facilitate complex hybrid topologies where computations can be executed on more than one type of infrastructure at any given time. Currently, the medical use case successfully runs on the Prometheus Cluster in Krakow using the GPU-enabled HPC nodes, the Low-Frequency Array (LOFAR) use case runs on the same cluster yet on traditional CPU resources, the airline revenue use case runs on cloud resources hosted in Bratislava while the agricultural use case runs on the CoolMUC cluster in Munich.

Based on the above runs we were able to confirm that the platform is able to execute appropriate workloads as required. We were also able to collect measurements confirming that the platform's operation parameters are acceptable and would not hinder our goal of maximizing its user friendliness while retaining a sufficient level of scalability. Finally, we performed scalability testing to confirm that the performance overheads of the platform scales linearly with load.

Conclusions and future work

In this paper, we describe the current status of the PROCESS platform. So far, we integrated a solution that enables running various use cases on heterogeneous e-Infrastructures supported by the PROECESS platform across a range of core technologies (HPC/cloud) and geographical locations.

In the future we will work to further stabilize the solution and make it as user friendly as possible. The most important tasks for the PROCESS platform include:

- Further streamlining of data transfers across technologies and sites to enable more complex scenarios such as concurrent execution of a single application on multiple sites
- Support for updated versions of use case applications, such as a multi-node MPI version of the Medical Use Case backed by Horovod at Cyfronet, as well as a scaled-up version of the LOFAR pipeline running on SuperMUC-NG (currently number 9 on the TOP500 list)
- Improving the user interface to provide better integration with Science Gateways such as the LOFAR portal, including delivery of results to end users, taking into account physical restrictions imposed by the size of input and output data

Acknowledgements

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 Mara Graziani, Balázs Somosköi, Jason Maassen and Henning Müller for their valuable insight on the presented subject.

References

1. 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.
2. M. Bobak, A. S. Z. Belloum, R. Cushing, J. Meizner, P. Nowakowski, V. Tran, O. Habala, J. Maassen, B. Somosköi, M. Graziani, M. Heikkurinen, M. Höb, J. Schmidt, L. Hluchý, "Reference exascale architecture", 15th International Conference on eScience, San Diego, USA
3. Rimrock, https://rimrock.plgrid.pl/
4. Cloudify, https://cloudify.co/
5. Docker, https://www.docker.com/
6. Singularity, https://sylabs.io/singularity/
7. Charliecloud, https://hpc.github.io/charliecloud/
8. The PL-Grid National Computing Infrastructure, www.plgrid.pl.
9. Bubak M., Bartyński T., Gubała T., Harężlak D., Kasztelnik M., Malawski M., Meizner J., Nowakowski P., EurValve Model Execution Environment in Operation. Editors: Marian Bubak Michał Turała.