

Towards a Universal Platform for Large Scale Simulations on Prometheus

M. Bubak^{1,2}, J. Meizner¹, P. Nowakowski^{1,2}, T. Gubała^{1,2}, M. Malawski^{1,2}

¹ACC Cyfronet AGH, AGH University of Science and Technology, Kraków, Poland

²Sano, Centre for Computational Personalised Medicine, Kraków, Poland

e-mails: bubak@agh.edu.pl, j.meizner@cyfronet.pl, p.nowakowski@cyfronet.pl, t.gubala@cyfronet.pl, malawski@agh.edu.pl

Keywords: HPC, execution environment, SLURM

1. Motivation

Our investigations concern methods, techniques and environments for programming and execution of large-scale complex scientific applications on different e-infrastructures. Recently, we have been and are still working on such environments for computational medicine, supporting – among others:

- Valvular Heart Disease simulations in the scope of the EurValve Project [1],
- Cancer diagnosis based on AI-driven medical image analysis in the scope of the PROCESS project [2],
- Pediatric Oncology to support decision making in detection of tumors based on biomarkers in the scope of the PRIMAGE project [3],
- Wide range of simulations of fluid dynamics and Finite Element Analysis envisioned in the scope of the Sano project [4].

2. The platform and its components

Traditionally scientists need to prepare data and sets of batch scripts to run any computations on the Slurm queuing system. As this was not always straightforward for domain specialists, we have prepared (in the scope of the EurValve project) a streamlined solution called the Model Execution Environment (MEE). Since then we have augmented and adapted this solution to serve the rest of the aforementioned large-scale simulations.

However, for every project the workflow is unified, providing a universal look-and-feel for the end users, and including the following steps:

- Preparation of the directory structure used by applications,
- Staging (in) to move data from a remote location (source location or another HPC infrastructure) to Prometheus,
- Automatic preparation of the relevant batch script (with all required parameters, properly interpolated based on the user's choice entered in the web portal),
- Running the job on Prometheus (CPU or GPU partition) via Rimrock [5],
- Staging (out) to move data to the designated output location and presenting results.

Additionally, the platform has been integrated with PLGrid OpenID to enable authentication and authorization services, as well as smooth credential delegation via proxy certificates. This ensures that the end user is not required to use any manual tools, such as grid-proxy-init, while the security (including accountability) of the solution is protected.

3. Conclusions

We have elaborated a scalable platform, that was already validated with a set of large-scale applications in the scope of multiple EU projects. Our goal, which – in our opinion – has been met, was to help scientists to run their code in a more streamlined fashion than before.

In the process of reaching this goal we have negotiated and utilized numerous computational grants, including eurvalve [2-6], process1/1gpu/2/2gpu and primage1.

Acknowledgments. This work is supported by the European Union’s Horizon 2020 research and innovation programmes under grant agreements No689617 (EurValve), 777533 (PROCESS), 826494 (PRIMAGE) and IRAP FNP programme (Sano, Grant MAB PLUS/2019/13).

This research was supported in part by PLGrid 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. EurValve, <http://www.eurvalve.eu/>,
2. 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.
3. PRIMAGE Project, <https://www.primageproject.eu/>,
4. Sano Centre, <https://sano.science/>,
5. Rimrock, <https://rimrock.plgrid.pl/>.