Container workload and orchestration in HPC

Kees de Jong & Maxim Masterov

June 15, 2021

Table of contents

Introduction

- In e.g. federated HPC infrastructures it is a challenge to maintain predictable software environments.
- With container technology there is also the question of orchestration (SLURM versus Kubernetes).
- The conclusion in this presentation in part based on related work[1][2][3].
- Left out of scope in this presentation: Docker, udocker, Podman and Shifter.

[1] *HPC workloads in containers: Comparison of container run-times*. Justin W. Flory's blog. URL: https://blog.justinwflory.com/2019/08/hpc-workloads-containers/.

[2] Pankaj Saha et al. "Evaluation of docker containers for scientific workloads in the cloud". In: *Proceedings of the Practice and Experience on Advanced Research Computing*. 2018, pp. 1–8.

[3] *The State of HPC Containers*. StackHPC Ltd. URL: https://www.stackhpc.com/the-state-of-hpc-containers.html.

Research questions

1. How do container technologies compare in terms of usability, security, features, and performance on single and multi node compute jobs?
2. How to orchestrate/schedule compute jobs with containers? How do Kubernetes and SLURM compare with each other in terms of usability, scheduling features, and resource allocation?

Singularity

Secure and very popular container solution which allows to build and run containers without root.

Charliecloud

Secure container solution with a small attack surface due to its lightweight nature. Charliecloud also does not require root privileges to build and run a container.

Enroot

Mixes different isolation methods, while staying lightweight, with builtin GPU support. Cannot build containers without root (but can be done by Buildah).

SLURM

Mainly focused on scheduling distributed parallel compute jobs with a defined end time. Where it is specialized in customizable efficient partitioning, queuing, accounting, monitoring and prioritizing jobs while being topology-aware

Kubernetes

Mainly focused on keeping microservices up and running without interruption. Kubernetes includes high-availability and load-balancing features between containers to mitigate congestion and latency for the microservice. There are developments towards support for HPC workflows.

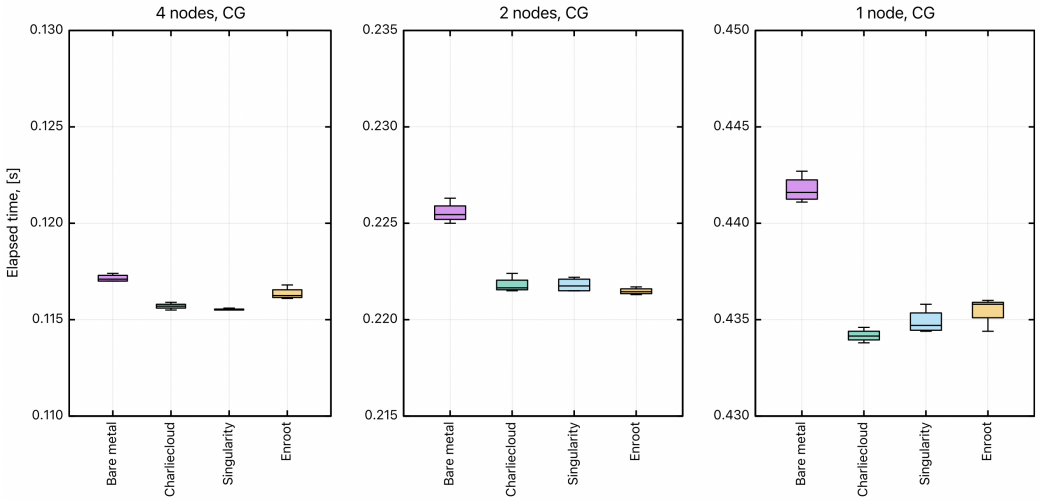
Method

- Goal: cover the majority of the possible use cases from fields like Computational Fluid Dynamics, Mechanical Engineering, Astrophysics, Machine Learning, and many others.
- We tested three linear solvers from PETSc library[4]: CG, BiCGSStab and AMG.
- The chosen linear solvers cover the most frequently used algebraic operations from the back-ends of most scientific codes.
- Every test was executed on 1, 2 and 4 nodes with 24 MPI tasks per node with hyperthreading switched off (repeated 4x)

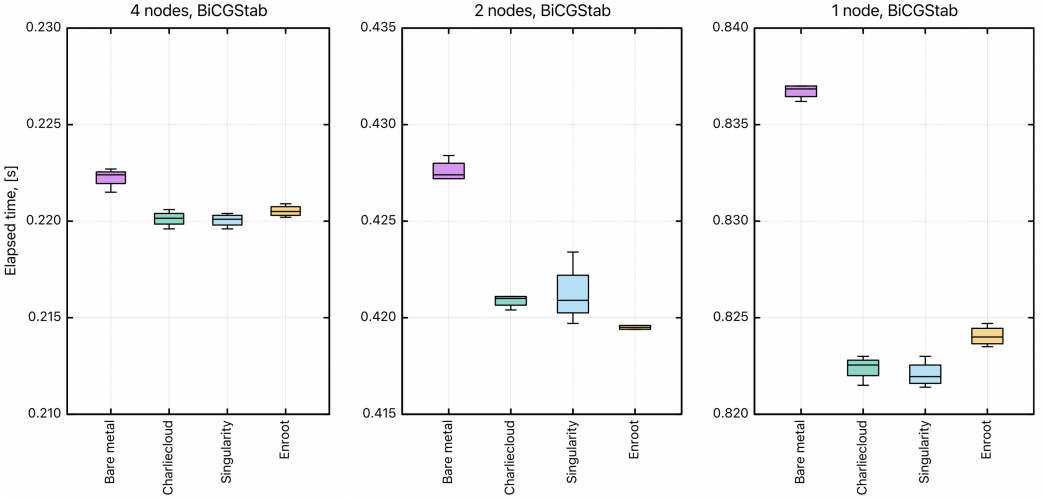
[4] S. Abhyankar et al. "PETSc DMNetwork: A Library for Scalable Network PDE-Based Multiphysics Simulations". In: *ACM Transactions on Mathematical Software* 46.1 (2020).

Results

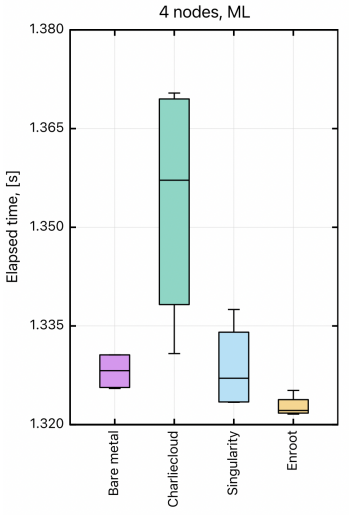
Benchmarks CG



Benchmarks BiCGSStab



Benchmarks ML



Introduction

Evaluation of the strengths and weaknesses of HPC-oriented container technologies. Based on an evaluation done by nVidia[a].

[a] *SLURM: Seamless Integration With Unprivileged Containers*. nVidia. URL: https://slurm.schedmd.com/SLUG19/NVIDIA_Containers.pdf.

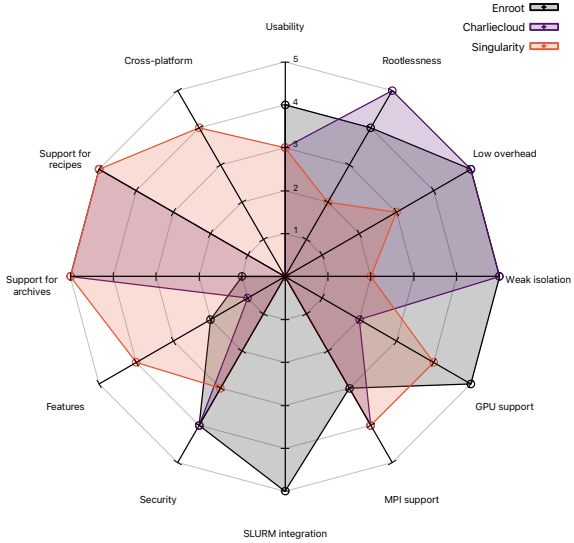
Usability, features, and support for archived images

- Singularity has 78 man pages. Charliecloud 20 manuals. Enroot had none.
- Online documentation was fine for all.
- Enroot had the least features, easier to start working with.
- Enroot lacked archived images (simplifies off-site utilization and execution of containers).
- Had to build and maintain our own RPMs for enroot[5] [6].

[5] https://fedorapeople.org/cgit/keesdejong/public_git/rpmbuild.git/tree/SPECS/enroot.spec
[6] https://fedorapeople.org/cgit/keesdejong/public_git/rpmbuild.git/tree/SPECS/pyxis.spec

SLURM and MPI support

- Enroot provided a SLURM plugin (significantly simplifies HPC workflows).
- Enroot allows users to rely on system-defined rules for core affinity and core binding.
- All containers allowed to compile and execute MPI applications in the container.


Enroot
Charliecloud
Singularity
Usability
Rootlessness
Low overhead
Weak isolation
GPU support
MPI support
SLURM integration
Security
Features
Support for archives
Support for recipes
Cross-platform
1
2
3
4
5

Kubernetes versus SLURM

- Kubernetes is not performing optimally compared to SLURM.
- Typical HPC characteristics such as support for NUMA management, GPUs, InfiniBand, and FPGU is developing in Kubernetes.
- MPI support for Kubernetes, namely for large scale Machine Learning by the use of e.g. Kubeflow.
- Kubernetes lacks topology-aware scheduling.

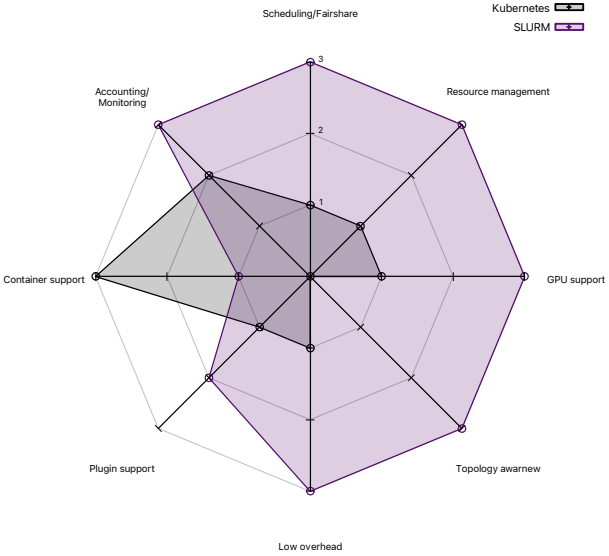
Deployment suggestions (1/2)

1. Maintain separate infrastructures: setup different environment for Kubernetes workloads, comes with increased cost.
2. Switch to hybrid workload managers: such as Univa Grid Engine Container Edition and IBM Spectrum LSF are adding native support for containers (Docker, Singularity and Shifter).

Deployment suggestions (2/2)

3. Use native job scheduling features in Kubernetes: known limitations in the context of HPC.
   - Slower time-to-spool due to determining network plugins[7].
   - Worker pods have to be first provisioned before a controller pod could be provisioned via the batch job.
   - Suffers latency from utilizing SSH for its communication protocol (SLURM uses MUNGE).
   - Node availability determination slower (added layers of virtual network and custom dynamic DNS).
   - And much more...

[7] Jeremy Stephen Futral. "A method of evaluation of high-performance computing batch schedulers". In: *University of North Florida* (2019).


Scheduling/Fairshare
Kubernetes
SLURM
Accounting/
Monitoring
Resource management
3
2
1
Container support
GPU support
Plugin support
Topology awarnew
Low overhead

Conclusion

- Kubernetes excels at orchestrating containers, containerized HPC applications can be tricky to deploy on Kubernetes[8].
- Significant developments under way for HPC features in Kubernetes.
- SLURM is designed to be used in HPC and excels for that reason.
- Singularity and Enroot score best overall. Enroot excels with (nVidia) GPU workloads. Charliecloud looks very promising as well.
- Traditional HPC and Kubernetes co-existence is possible. However, there should be a valid use case for such a more complex and expensive setup[9]

[8] *Kubernetes Meets High-Performance Computing*. The Kubernetes Authors. URL: https://kubernetes.io/blog/2017/08/kubernetes-meets-high-performance/.

[9] *Cloudy Topics from the Hutch*. Fred Hutch. URL: https://www.fredhutch.org/en/events/partly-cloudy/_jcr_content/root/responsivegrid_1/panelcontainer_206790373/contents/downloadpdf_1916703528/file.

Full report can be downloaded by scanning the QR code or visit the short URL[10]

[10] https://bit.ly/3pTNfIZ