Containerization

Performance Characterization

Shantenu Jha: Project Manager

Matteo Turilli: Technical Supervisor

What is a container?

- A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computer environment to another.

- It is lightweight since it shares the machine’s OS system kernel and therefore do not require an OS per application.

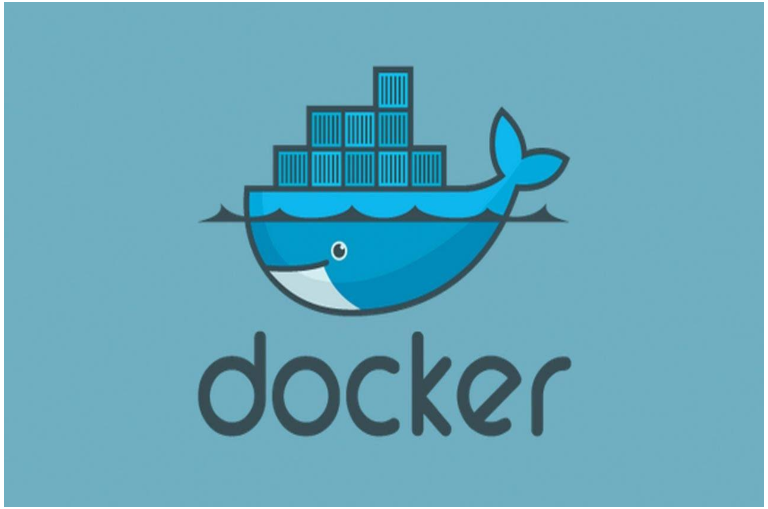
- A container image becomes a container at runtime, and said containerized software will always run the same, regardless of the infrastructure. Containers isolate software from its environment and ensure that it works uniformly

Docker

- It is the most well known and utilized container platform, designed primarily for network micro-service virtualization.

- Docker facilitates creating, maintaining and distributing container images. Containers are somewhat reproducible, easy to install, well documented and standardized.

- Docker works great for local and private resources, and you can develop and share your work with others through Docker-Hub. However, if you need to scale beyond your local resources, let’s say, HPCs, it won’t be efficient or even compatible.



Singularity

- Singularity is a container runtime, like Docker, but it starts from a very different place. It favors integration rather than isolation, while still preserving security restrictions on the container, and providing reproducible images.

- Singularity enables users to have full control of their environment. Singularity containers can be used to package entire scientific workflows, software and libraries, and even data. This means that you don’t have to ask your cluster admin to install anything for you, you can put it in a Singularity container and run.

Why Singularity over Docker on HPCs?

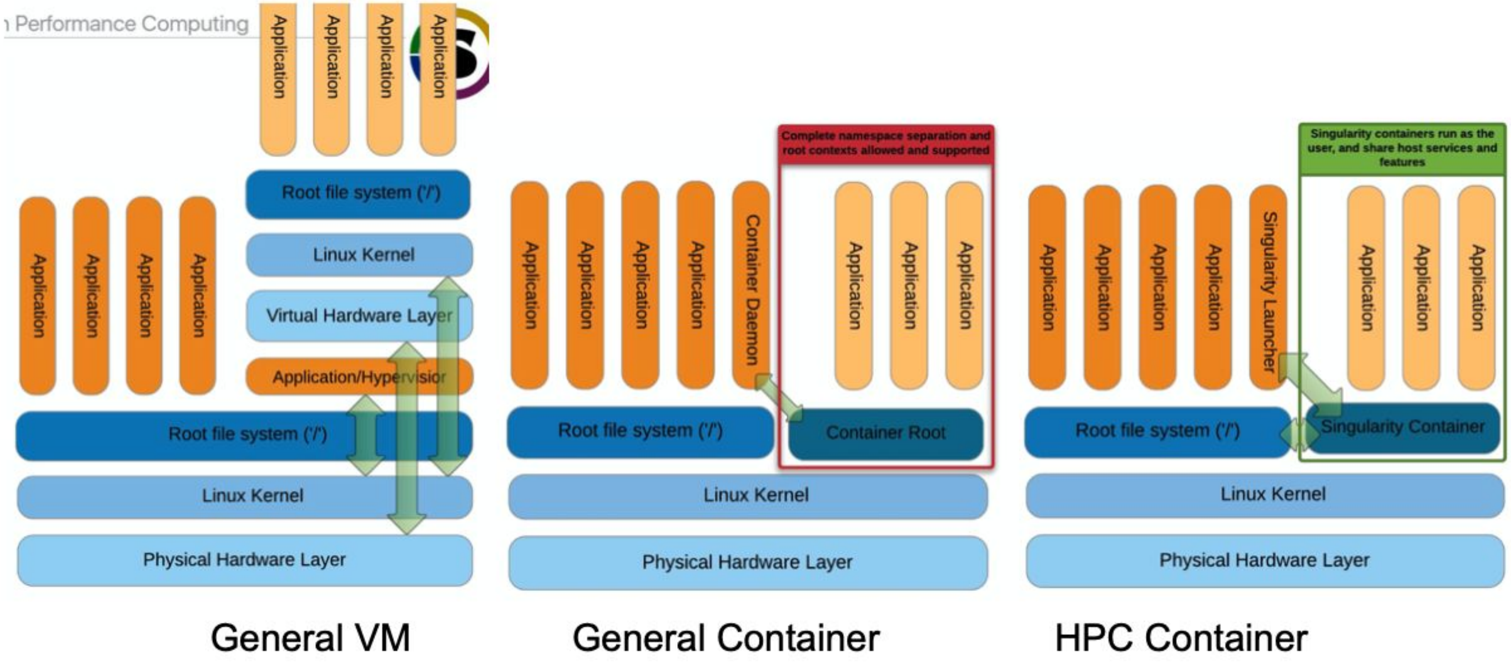
1. **Security:** Because of Docker daemon, a user inside the Docker container is able to obtain root access on the host. In contrast, Singularity solves this by running the container with the user's credentials.

2. **HPC Scheduler:** Users submit jobs with CPU/memory/time requirements. The Docker command is just an API client that talks to the docker daemon, so the resource requests and actual usages don't match. Singularity runs container processes without a daemon. They just run as child processes. In other words, Docker does not support any HPC job scheduler, but Singularity integrates seamlessly with all job schedulers including SLURM, Torque, SGE, etc.

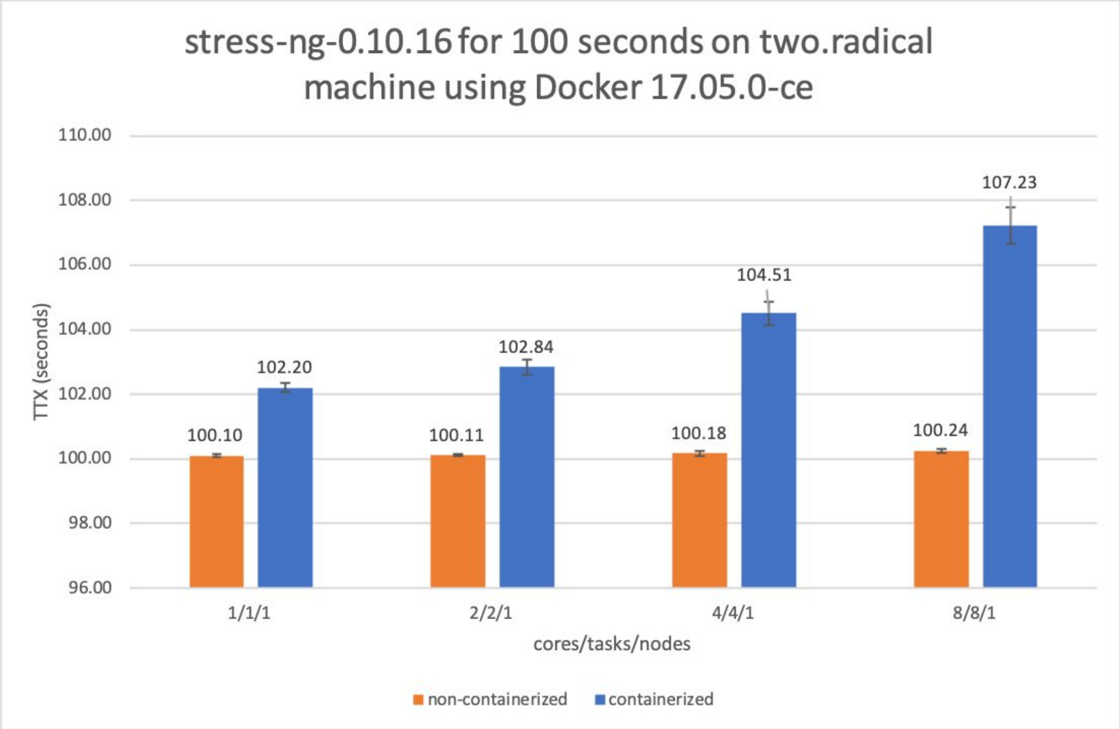
3. **GPU support:** Docker does not support GPU natively. Nvidia Docker is a GPU-enabled Docker container, but it pre-installs various software that a user may not need. Singularity is able to support GPUs natively. Users can install whatever CUDA version and software they want on the host which can be transparently passed to Singularity.

4. **MPI support:** Docker does not support MPI natively. So if a user wants to use MPI with Docker, a MPI-enabled Docker needs to be developed. If a MPI-enabled Docker is available, the network stacks such as TCP and those needed by MPI are private to the container which makes Docker containers not suitable for more complicated networks like Infiniband. In Singularity, the user's environment is shared to the container seamlessly.

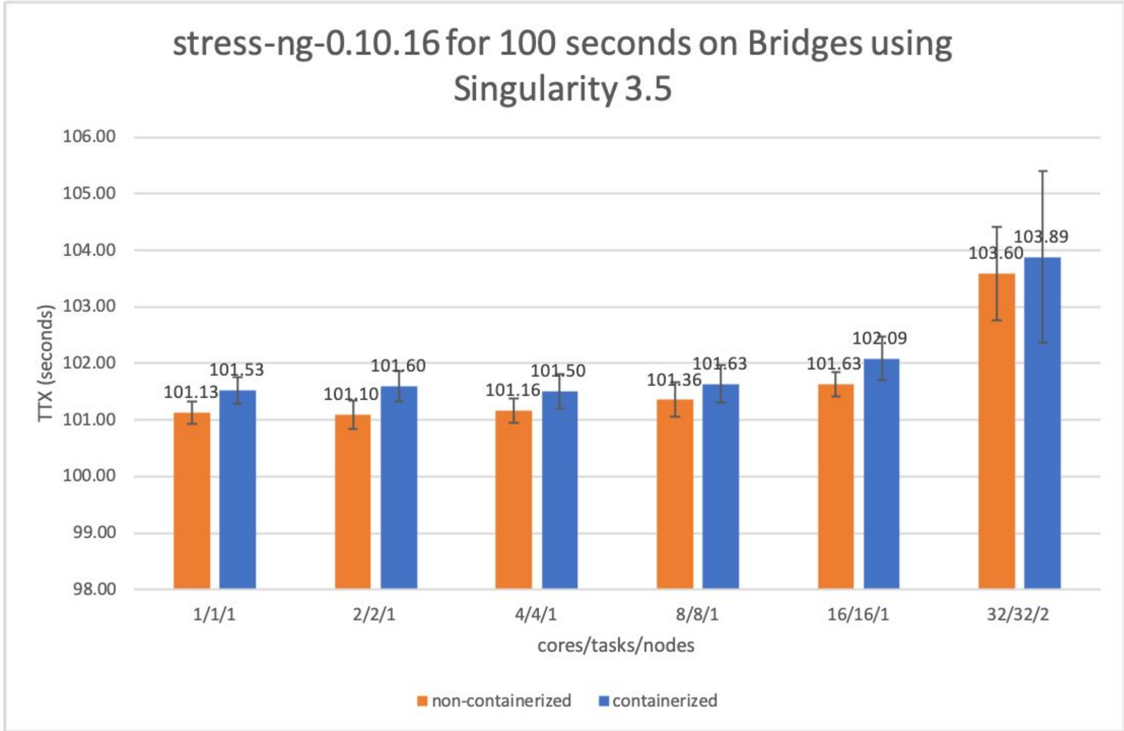
Virtual Machine vs. Docker vs. Singularity



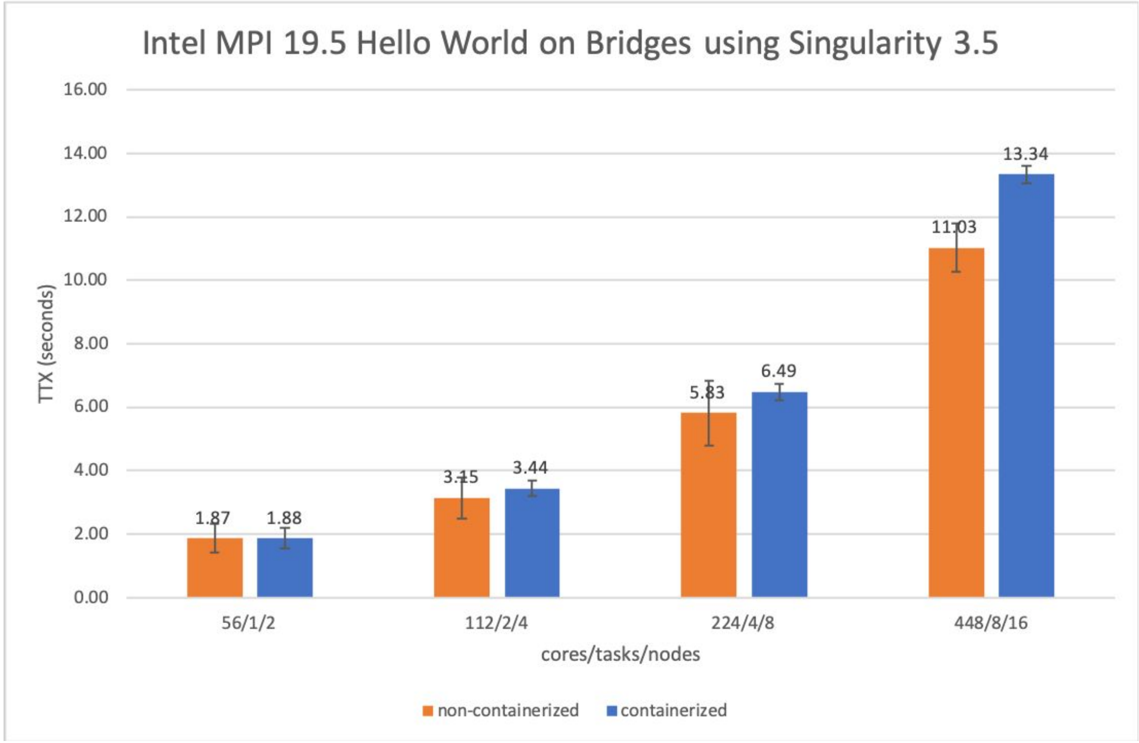
Performance Characterization - Localhost



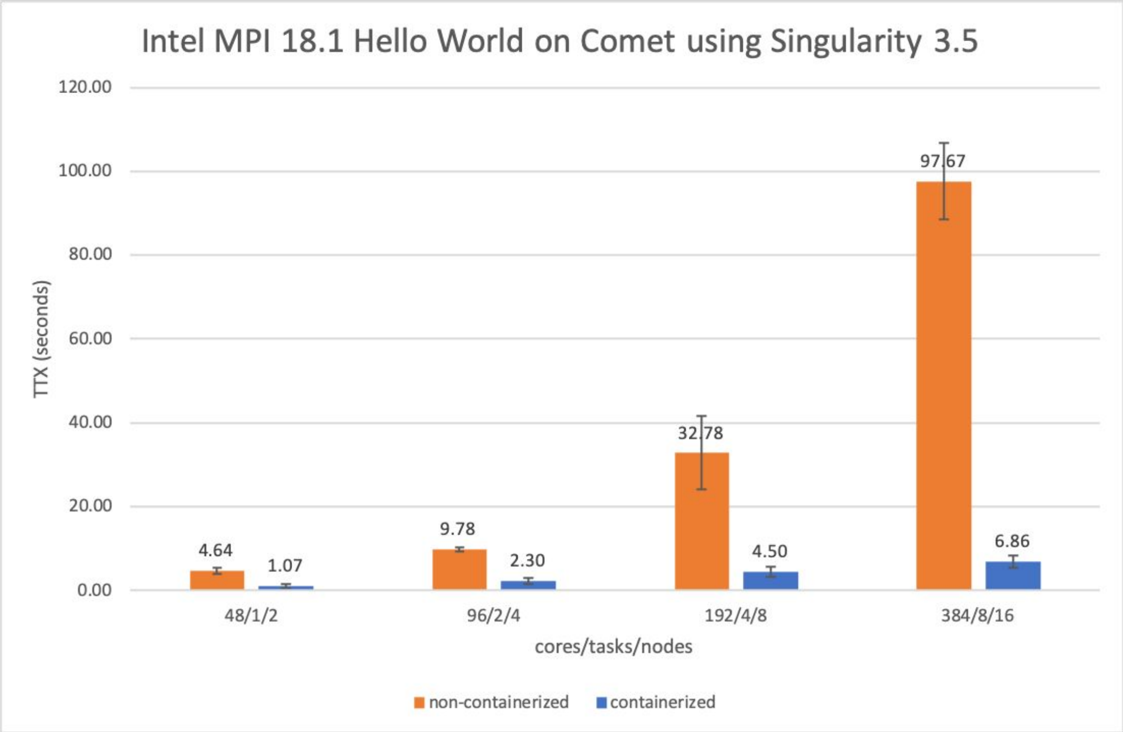
Performance Characterization - XSEDE Bridges



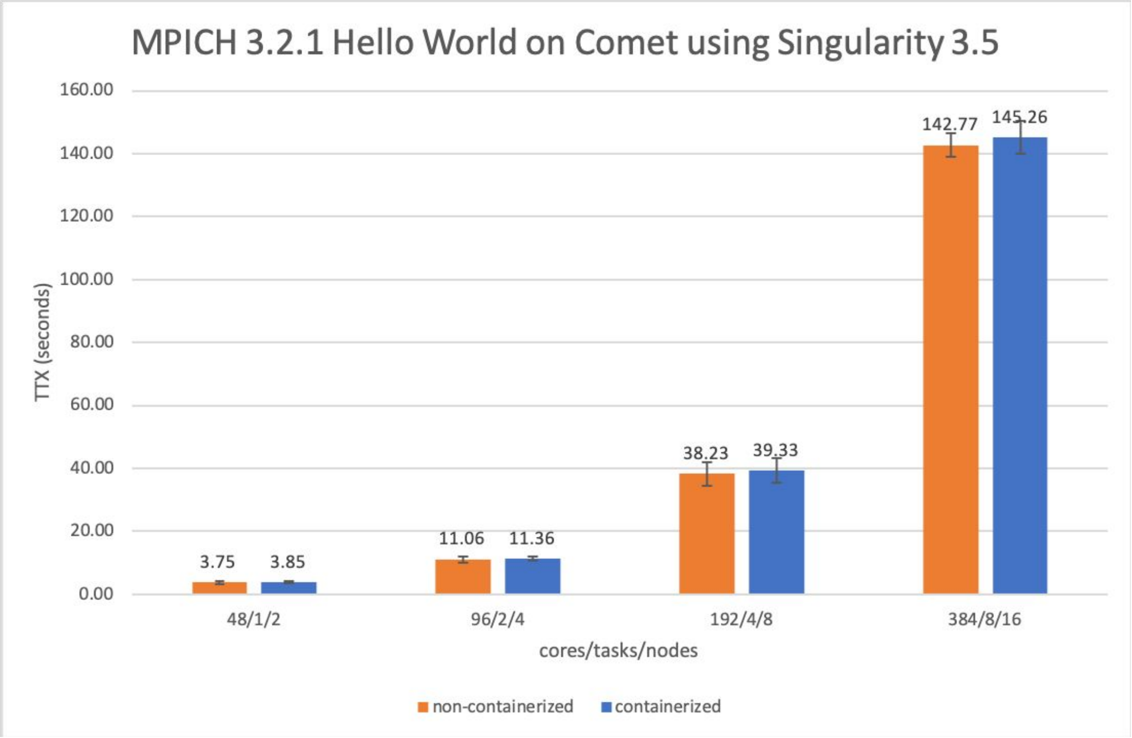
Performance Characterization - XSEDE Bridges



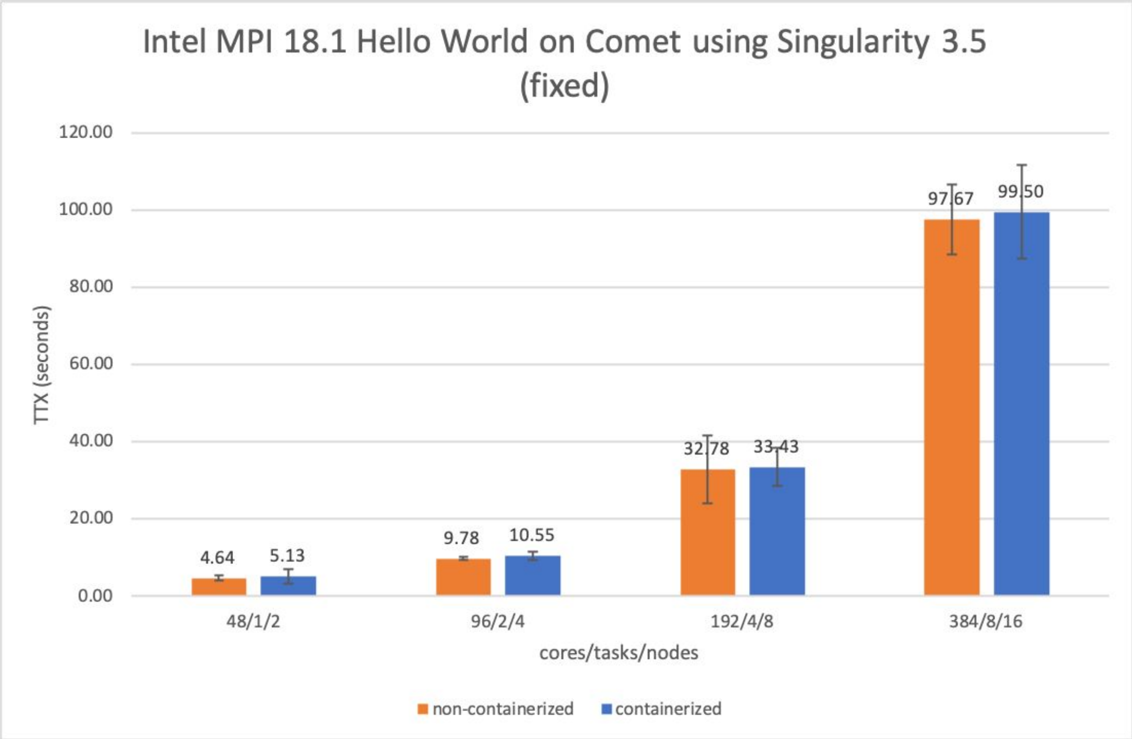
Performance Characterization - XSEDE Comet



Performance Characterization - XSEDE Comet



Performance Characterization - XSEDE Comet



References

1. https://www.docker.com/resources/what-container
2. https://docs.docker.com
3. https://sylabs.io/docs/
4. https://downloads.dell.com/manuals/all-products/esuprt_software/esuprt_it_ops_datcentr_mgmt/high-computing-solution-resources_white-papers10_en-us.pdf
5. https://djangostars.com/blog/what-is-docker-and-how-to-use-it-with-python/
6. https://likegeeks.com/docker-tutorial/#Run-your-container
7. https://www.thegeekdiary.com/how-to-update-add-a-file-in-the-docker-image/
8. https://github.com/TACC/TACC-Singularity/blob/master/docs/from-docker-to-singularity.md
9. https://singularity-tutorial.github.io
10. https://www.hpc.iastate.edu/guides/containers
11. https://www.sdsc.edu/education_and_training/tutorials1/singularity.html