



Rootless containers with udocker

Jorge Gomes (LIP), Mário David (LIP), João Martins (LIP), João Pina (LIP), Isabel Campos (CSIC), Alvaro López García (CSIC), Pablo Orviz (CSIC), Valentin Kozlov (KIT)

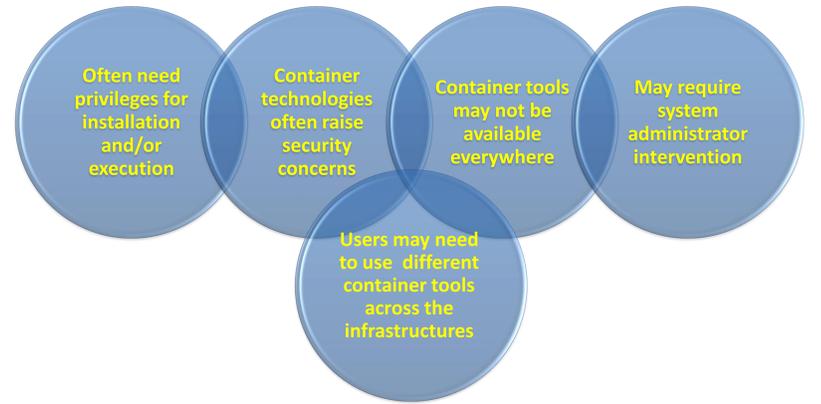
Running scientific applications across infrastructures can require considerable effort from the end-users

- Several computing systems
- Notebooks, Desktops, Farms, Cloud, HPC
- Several operating systems
- Linux flavors, Distribution versions
- Specific computing environments
- Compilers, Libraries, Customizations
- Multiple applications often combined
- Portability, Maintainability, Reproducibility



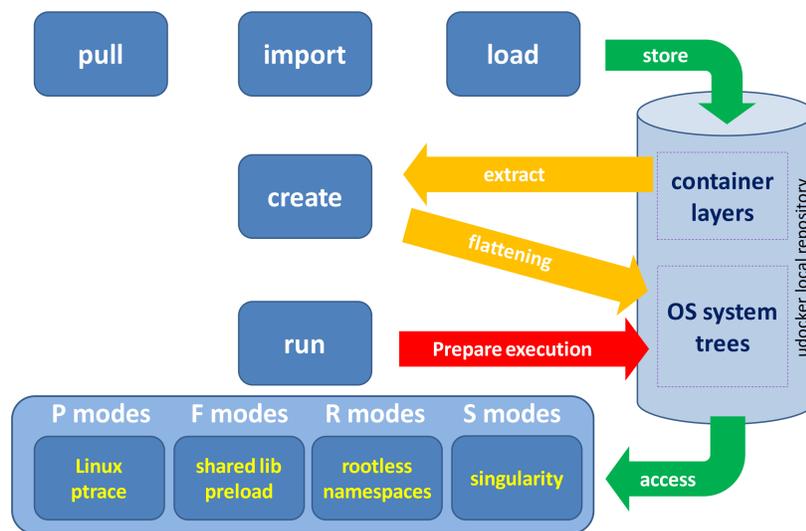
Need a consistent portable way of running applications across the infrastructures eco-system

Containers are a great way to encapsulate applications and run across different systems but also have limitations



udocker enables the execution of application containers in user space without requiring privileges or sysadmin intervention

- Empowers end-users
- Pulling from docker repositories
- Import of directory tarballs
- Load of docker images
- Multiple execution modes supported
- Most execution modes without privileges
- Works across multiple Linux distributions
- Adequate for HTC, HPC and GPGPU computing in batch and interactive clusters



udocker is available on github:
<https://github.com/indigo-dc/udocker>

Basic usage

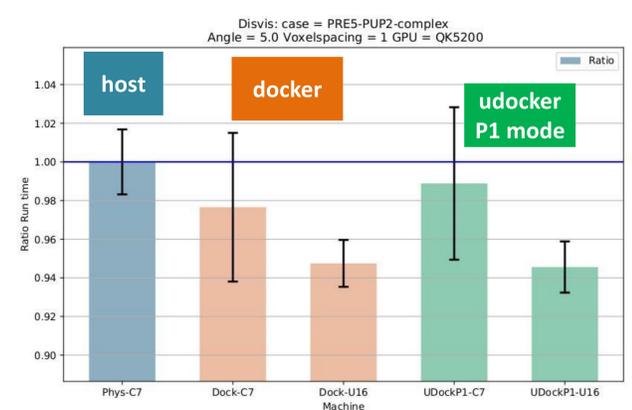
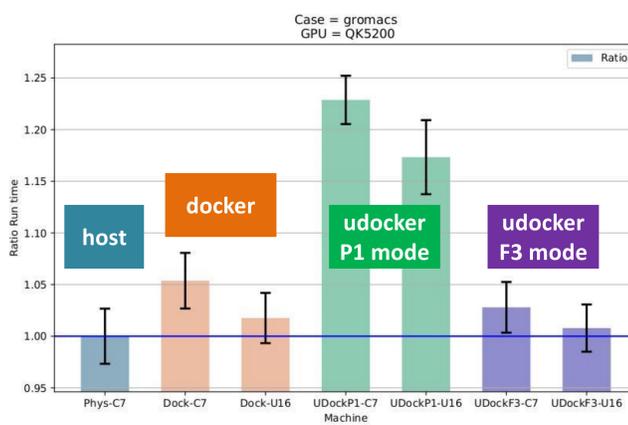
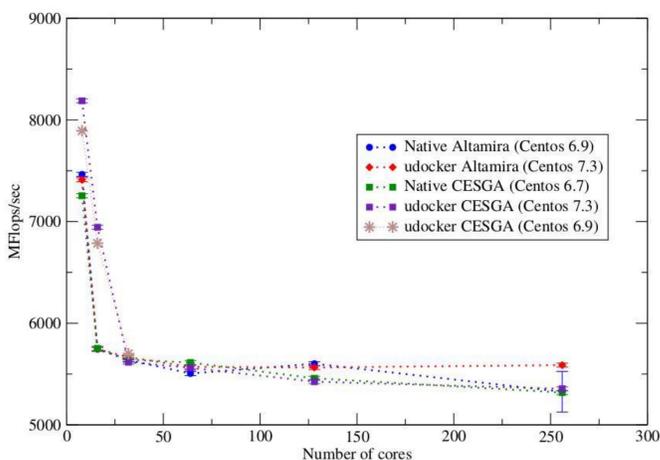
```
$ curl https://raw.githubusercontent.com/indigo-dc/udocker/master/udocker.py > udocker
$ chmod u+rx ./udocker
$ ./udocker install
$ ./udocker run ubuntu:18.04
```

← Pull and run the container from docker hub

Roadmap

- Installation with Python pip
- Porting to Python 3
- Improved translation of volume pathnames
- Improved support for MPI libraries
- Improved root emulation
- Support for OCI containers
- Network port remapping in Fn modes
- Expand the set of execution engines

Application examples and execution performance



- OpenQCD code for lattice simulations
- Using OpenMPI
- Scaling performance as a function of the cores for the computation of application of the Dirac operator to a spinor field.
- udocker in default execution mode P1

- Gromacs code for molecular dynamics
- Using OpenCL with Nvidia GPUs in combination with OpenMP
- udocker P1 mode has lower performance than docker
- udocker F3 mode has better performance than docker

- DISVIS code to visualize and quantify the information content of distance restraints between macromolecular complexes
- Using OpenCL and Nvidia GPUs
- Performance with docker and udocker are similar and very close to the physical host. Better performance with Ubuntu container.
- udocker in default execution mode P1

udocker is an open source product initially developed by the INDIGO-datacloud H2020 project.

udocker is being further developed in the DEEP-HybridDataCloud H2020 project to support deep learning and data analysis with accelerated computing. udocker maintenance and support is performed by LIP (www.lip.pt) and INCD (www.incd.pt) with support from the EOSC-hub H2020 project.