



UDOCKER

BE ANYWHERE

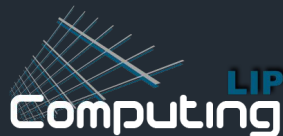
Developers:

Jorge Gomes <jorge@lip.pt>

Mário David <david@lip.pt>

Speaker:

Samuel Bernardo <samuel@lip.pt>



Scientific computing and containers



Running applications across infrastructures may require considerable effort

- ❖ **Computers**
 - Several computing systems
 - Laptops, Desktops, Farms, Cloud, HPC
- ❖ **OSes**
 - Several operating systems
 - Linux flavors, Distribution versions
- ❖ **Environments**
 - Specific computing environments
 - Compilers, Libraries, Customizations
- ❖ **Applications**
 - Multiple applications often combined
 - Portability, Maintainability, Reproducibility



Need a consistent portable way of running applications

Containers for batch processing

- Challenges of batch systems?
 - Integrate it with the batch system (how to start/stop etc) ?
 - Respect batch system policies (such as quotas/limits) ?
 - Respect batch system actions (job delete/kill) ?
 - Collect accounting ?

- Can we execute in a more basic way?
 - Can we download container images ?
 - Can we run without a layered filesystem ?
 - Can we run them as normal user ?
 - Can we still enforce container metadata ?

udocker

- Run applications encapsulated in docker containers:
 - without using docker
 - without using (root) privileges
 - without system administrators intervention
 - without additional system software
 - Does not require Linux namespaces
- Run:
 - as a normal user
 - with the normal process controls and accounting
 - in interactive or batch systems
 - does not run as a service (for that use containerd)

udocker

How does it work ...

udocker

udocker is open source

Developed under the **Indigo-Datacloud** and **DEEP Hybrid-Datacloud** projects



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

- <https://github.com/indigo-dc/udocker/tree/master>
- <https://github.com/indigo-dc/udocker/tree/devel>

Documentation:

<https://github.com/indigo-dc/udocker/tree/master/doc>

udocker

- Run time to execute docker containers:
 - search
 - pull
 - images
 - create
 - rmi
 - ps
 - rm
 - run
 - login
 - logout
 - load
 - save
 - import
 - export
 - setup
 - clone
 - verify
 - Inspect
 - mkrepo

udocker

How it was implemented ...

udocker

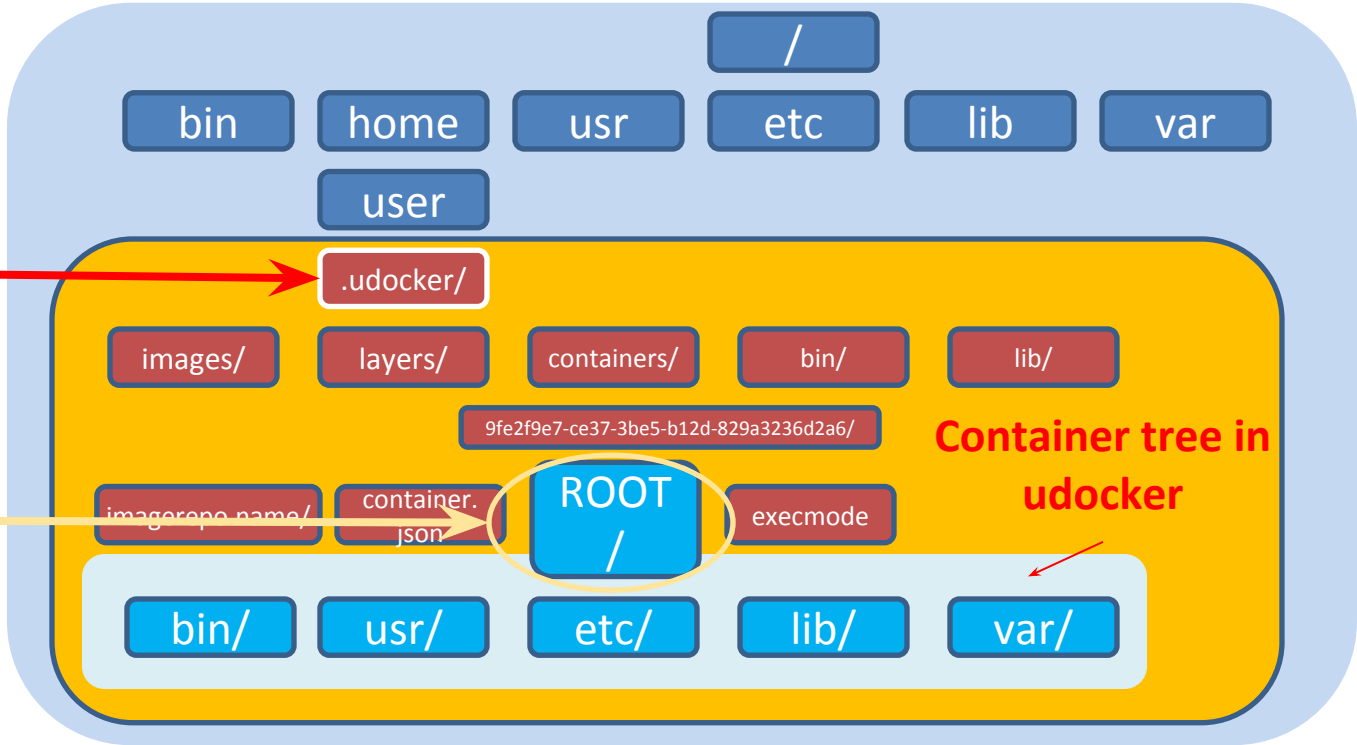
- **Implemented**
 - python, C, C++, go
- **Can run:**
 - CentOS 6, CentOS 7, Fedora \geq 23
 - Ubuntu 14.04, Ubuntu 16.04
 - Any distro that supports python 2.6 and 2.7
- **Components:**
 - Command line interface docker like
 - Pull of containers from Docker Hub
 - Local repository of images and containers
 - Execution of containers with modular engines

udocker: run - Container

- Execution
- chroot-like

udocker directory tree
\$HOME/.udocker

chroot to this
directory becomes
the new root for
container processes



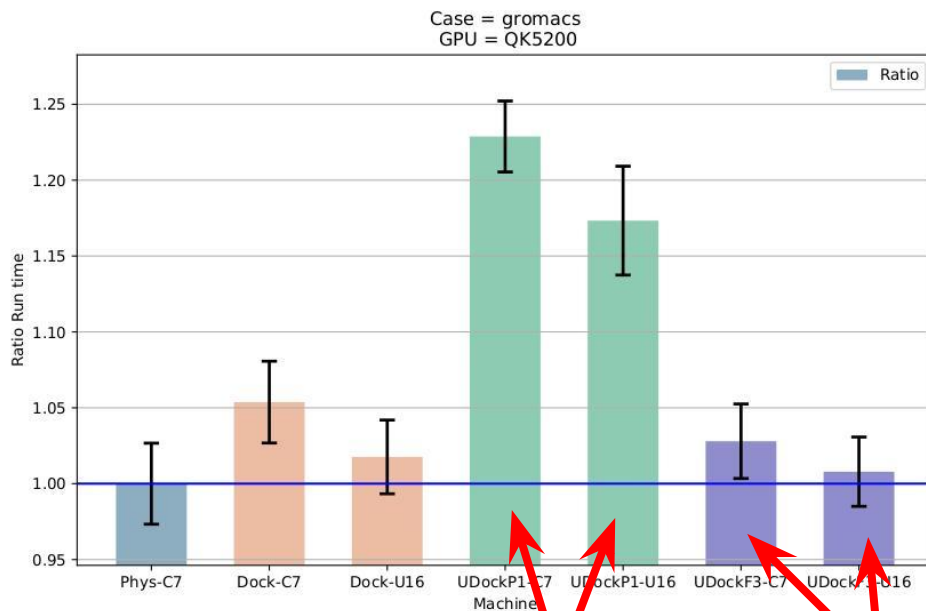
udocker: Execution engines

- udocker supports several techniques to achieve the equivalent to a chroot without using privileges
 - They are selected per container id via execution modes

Mode	Base	Description
P1	PRoot	PTRACE accelerated (with SECCOMP filtering) <input type="checkbox"/> DEFAULT
P2	PRoot	PTRACE non-accelerated (without SECCOMP filtering)
R1	runC	rootless unprivileged using user namespaces
F1	Fakechroot	with loader as argument and LD_LIBRARY_PATH
F2	Fakechroot	with modified loader, loader as argument and LD_LIBRARY_PATH
F3	Fakechroot	modified loader and ELF headers of binaries + libs changed
F4	Fakechroot	modified loader and ELF headers dynamically changed
S1	Singularity	where locally installed using chroot or user namespaces

udocker benchmarking

udocker & Molecular dynamics



Gromacs is widely used both in biochemical and non-biochemical systems.

udocker P mode have lower performance
udocker F mode same as Docker.

Using CUDA and OpenMP

udocker in P1 mode
udocker in F3 mode

PTRACE

SHARED LIB CALL

udocker & Phenomenology

Performance Degradation

	Compiling	Running
HOST	0%	0%
DOCKER	10%	1.0%
udocker	7%	1.3%
VirtualBox	15%	1.6%
KVM	5%	2.6%

MasterCode connects several complex codes. Hard to deploy.

Scanning through large parameter spaces. High Throughput Computing

C++, Fortran, many authors,

udocker in P1 mode

udocker

Next ...

udocker: What's next

- Ongoing:
 - <https://github.com/indigo-dc/udocker/tree/devel3>
 - Modularization of udocker (and unit tests)
 - Porting to Python3
 - pip install - <https://pypi.org>

udocker: What's next

- Next
 - Increase automation for MPI/infiniband applications
 - OpenMPI and MPICH
 - Better translation of “volume” directories
 - Command line interface enhancements
 - Improve root emulation

Thank you

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

