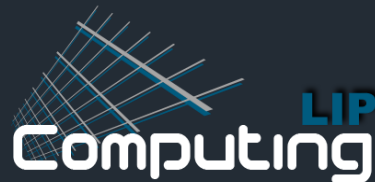


UDOCKER

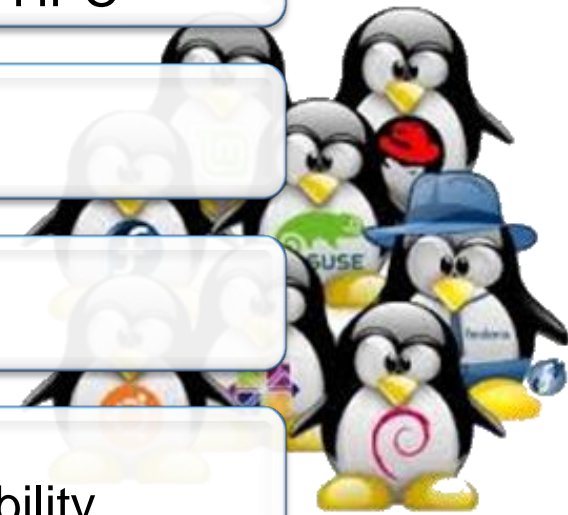
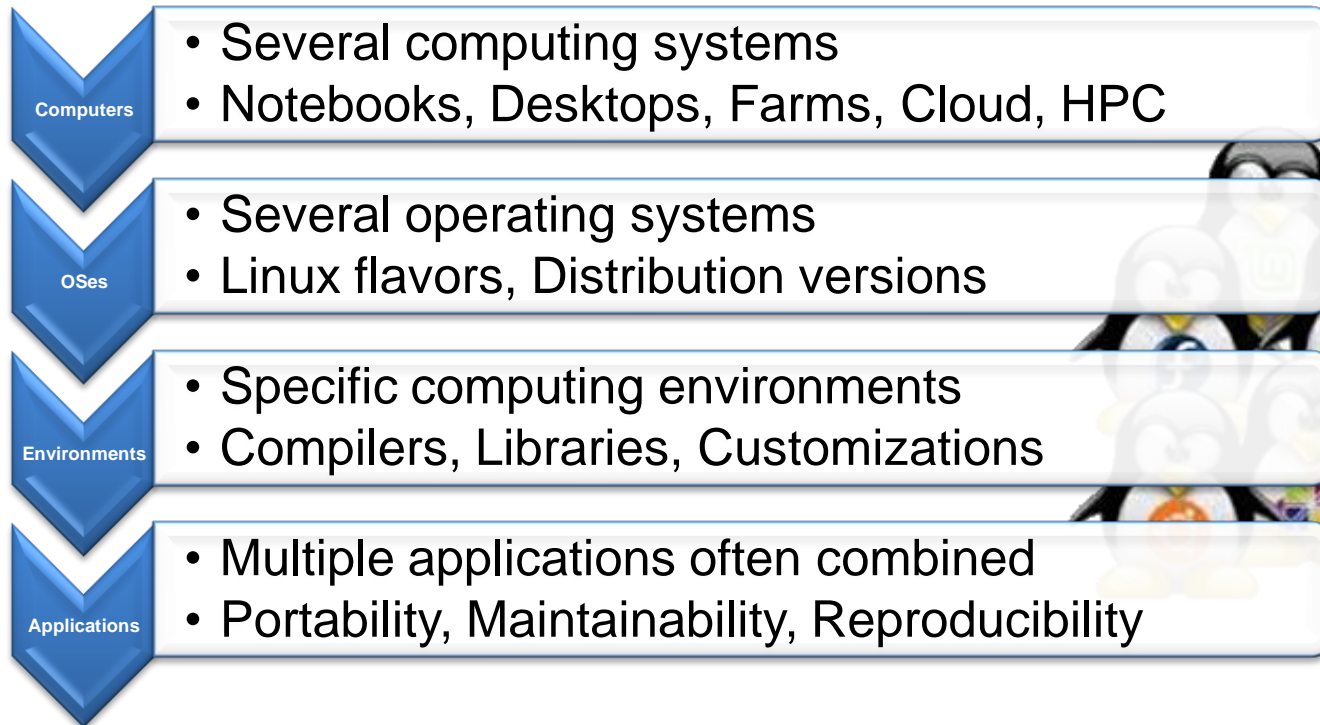
BE ANYWHERE



Jorge Gomes <jorge@lip.pt>
Mário David <david@lip.pt>

Scientific computing and containers

Running applications across infrastructures may require considerable effort



Need a consistent portable way of running applications

INDIGO-DataCloud

Italy national museums digital library

ENES climate models

Molecular dynamics analysis

DisVis and Powerfit

SOCIAL SCIENCE

LBT data archive

Galaxy instances

PHYSICS & ASTROPHYSICS

DARIAH Zenodo based repositories

Algae RNA sequencing

EMSO seismic data analysis

LHC/CMS clusters on demand

Omnia big analytics

Molecular dynamics of proteins

INDIGO udocker

- Run applications encapsulated in docker containers:
 - without using docker
 - without using privileges
 - without system administrators intervention
 - without additional system software
- and run:
 - as a normal user
 - with the normal process controls and accounting
 - in interactive or batch systems

INDIGO udocker

udocker in open source

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

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

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

udocker: install from github

```
$ curl https://raw.githubusercontent.com/indigo-dc/udocker/master/udocker.py > udocker
```

```
$ chmod u+rx udocker
```

```
$ ./udocker install
```

or devel

**Does not require compilation or system installation
Tools are delivered statically compiled**

udocker: pull images from repository

```
$ udocker pull ubuntu:14.04
```

Search for names and tags at:
<https://hub.docker.com/>

```
Downloading layer: sha256:bae382666908fd87a3a3646d7eb7176fa42226027d3256cac38ee0b79bdb0491
Downloading layer: sha256:f1ddd5e846a849fff877e4d61dc1002ca5d51de8521cced522e9503312b4c4e7
Downloading layer: sha256:90d12f864ab9d4cfe6475fc7ba508327c26d3d624344d6584a1fd860c3f0fefa
Downloading layer: sha256:a57ea72e31769e58f0c36db12d25683eba8fa14aaab0518729f28b3766b01112
Downloading layer: sha256:783a14252520746e3f7fee937b5f14ac1a84ef248ea0b1343d8b58b96df3fa9f
Downloading layer: sha256:a3ed95caeb02ffe68cdd9fd84406680ae93d633cb16422d00e8a7c22955b46d4
```

udocker: create container from image

```
$ udocker create --name=ub14 ubuntu:14.04
```

←
container-alias

```
9fe2f9e7-ce37-3be5-b12d-829a3236d2a6 ← container-id
```




UDOCKER
BE ANYWHERE

udocker: run container

```
$ udocker run ub14
```

udocker respects container metadata, if the container has a default cmd to run it will be run otherwise starts a shell

```
*****
*
*          STARTING 9fe2f9e7-ce37-3be5-b12d-829a3236d2a6
*
*****
executing: bash
root@nbjorge:/# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=14.04
DISTRIB_CODENAME=trusty
DISTRIB_DESCRIPTION="Ubuntu 14.04.5 LTS"
root@nbjorge:/# apt-get install firefox
```

← **root emulation**

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 does it work ...

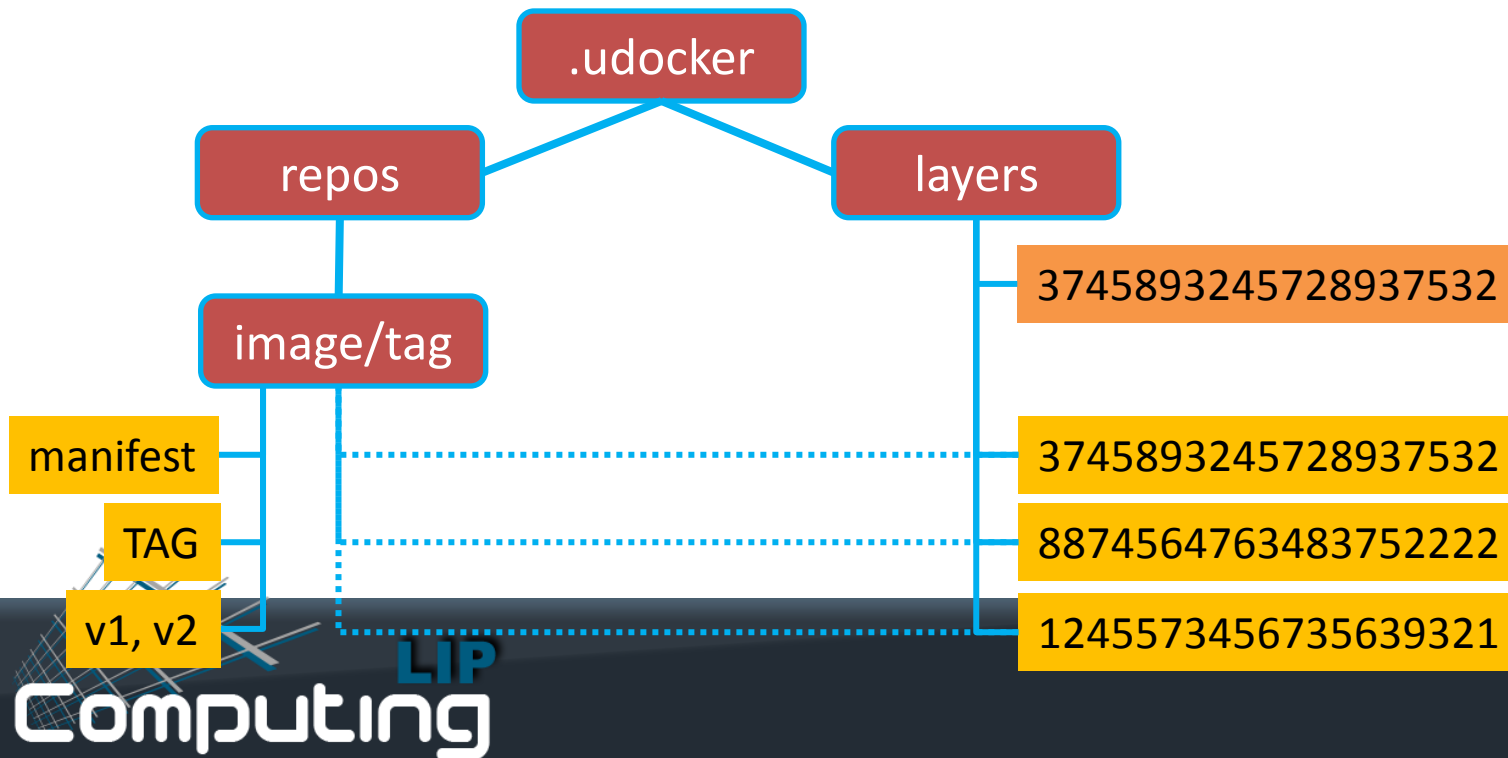
udocker

- 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 2.7
- 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: pull

- Images
 - Layers and metadata are pulled with DockerHub REST API
 - Image metadata is interpreted to identify the layers
 - Layers are stored in the use home directory under `~/.udocker/layers` so that can be shared by multiple images

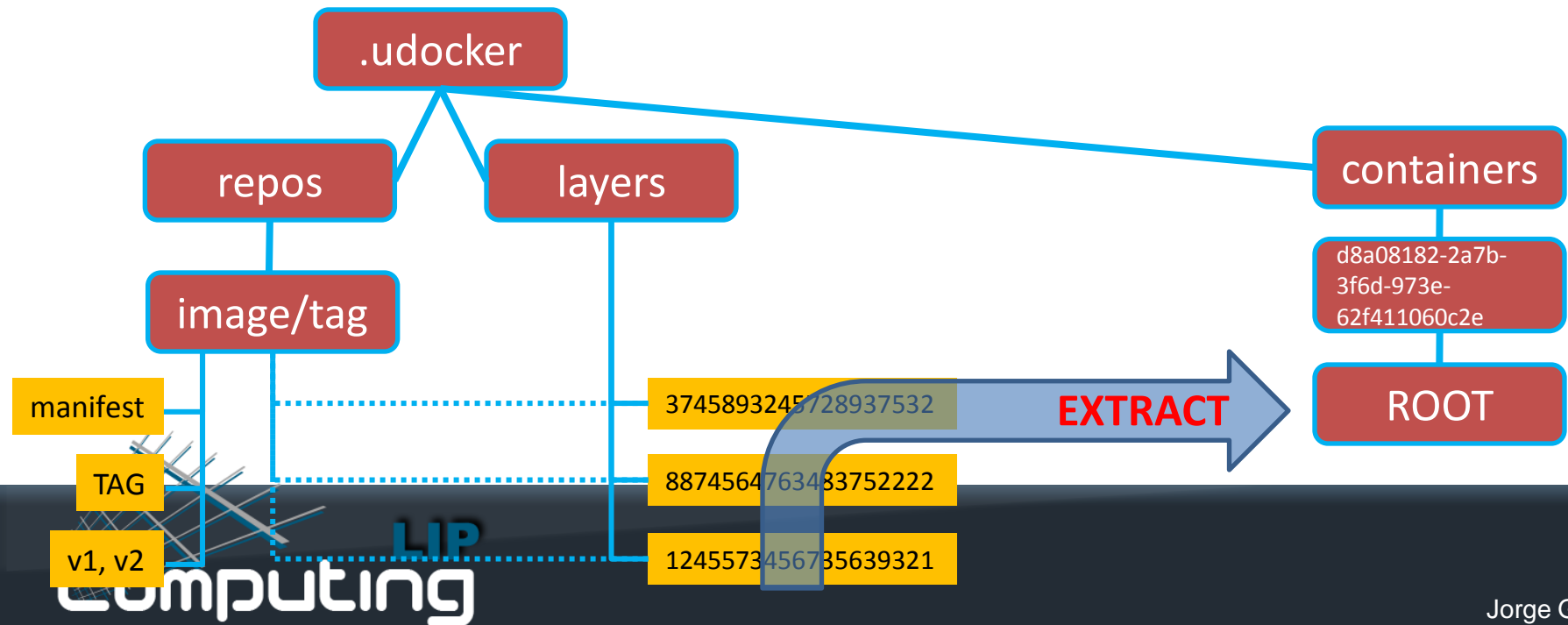




UDOCKER
BE ANYWHERE

Udocker: create

- Containers
 - Are produced from the layers by flattening them
 - Each layer is extracted on top of the previous
 - Whiteouts are respected, protections are changed
 - The obtained directory trees are stored under `~/.udocker/containers` in the user home directory





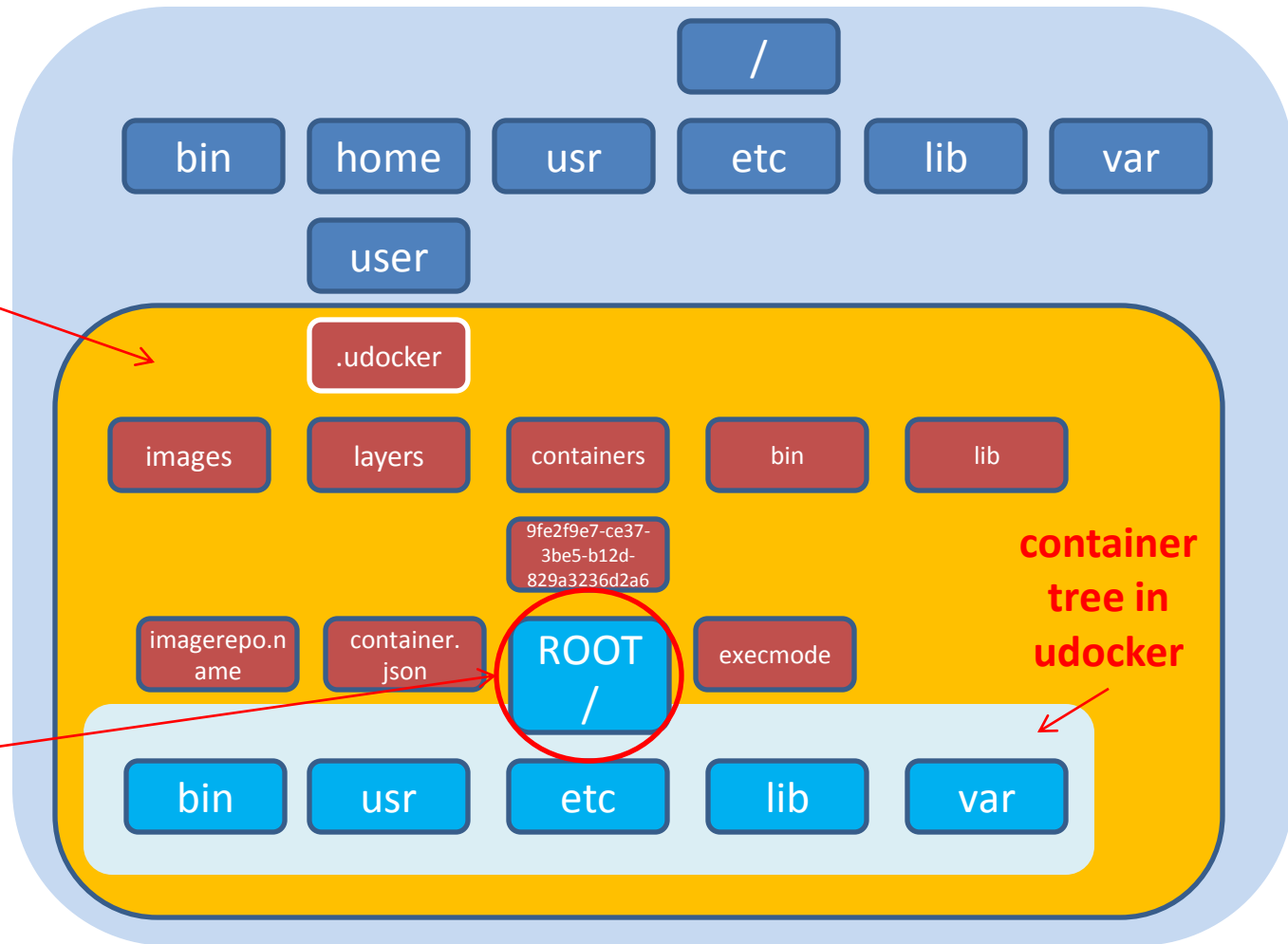
UDOCKER
BE ANYWHERE

udocker: run

- Execution
- chroot-like

**udocker
directory tree
\$HOME/.udocker**

**chroot to this
directory
becomes the
new root for
container
processes**



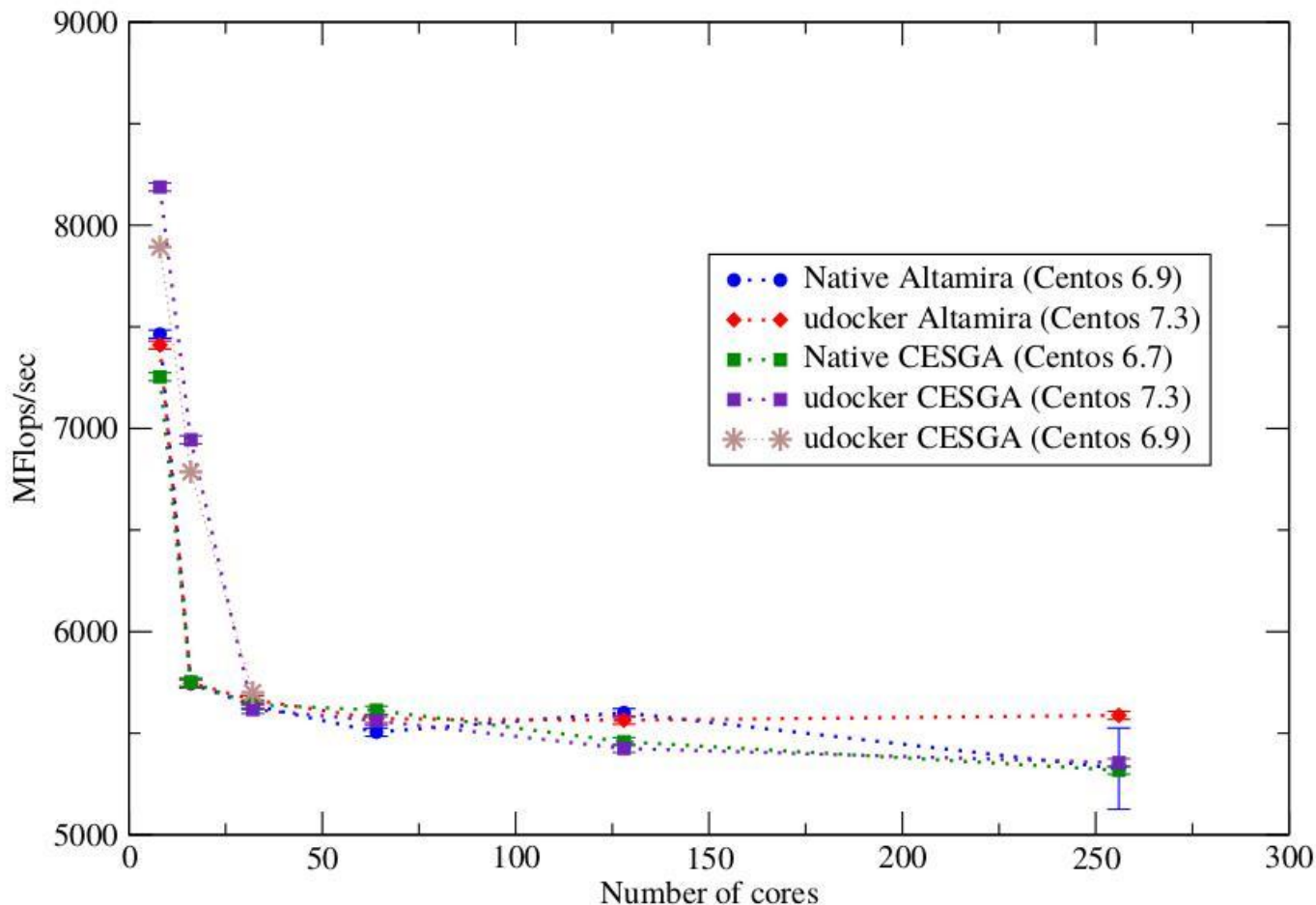


udocker: Execution methods

- 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) ← 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 & Lattice QCD



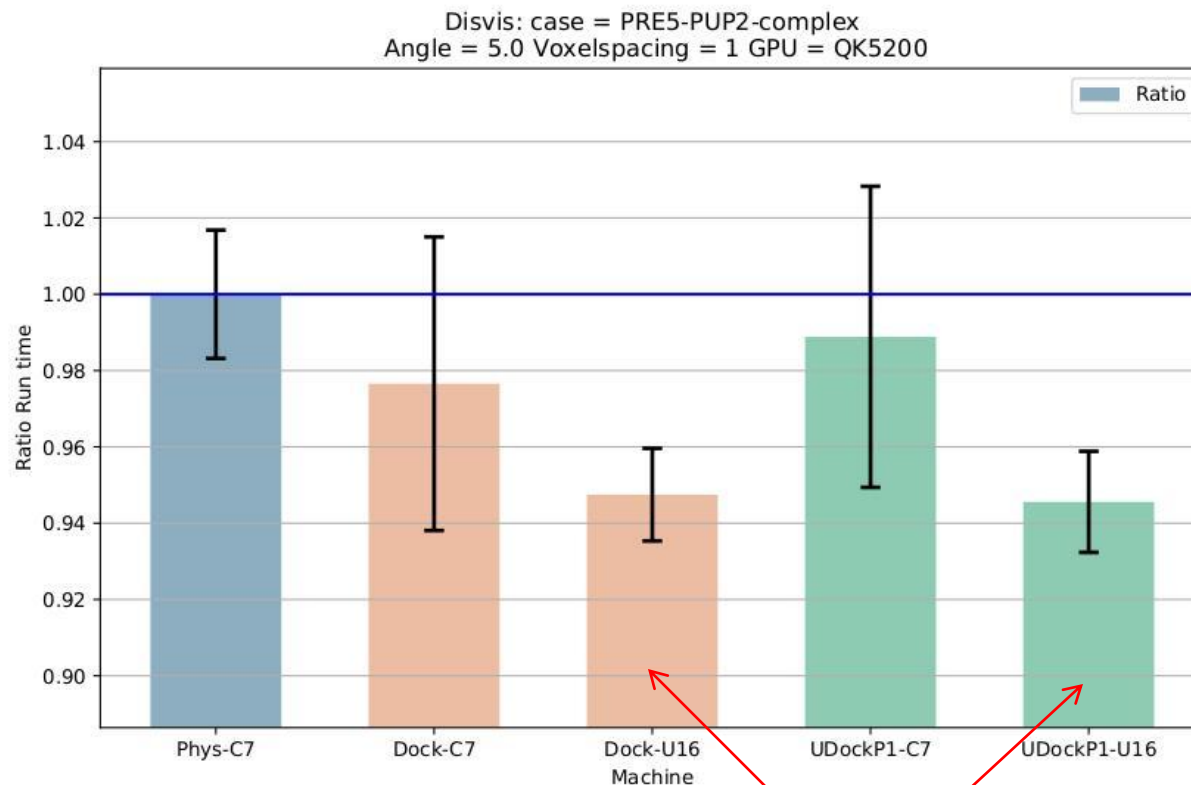
OpenQCD is a very advanced code to run lattice simulations

Scaling performance as a function of the cores for the computation of application of the Dirac operator to a spinor field.

Using OpenMPI

udocker in P1 mode

udocker & Biomolecular complexes



DisVis is being used in production with udocker

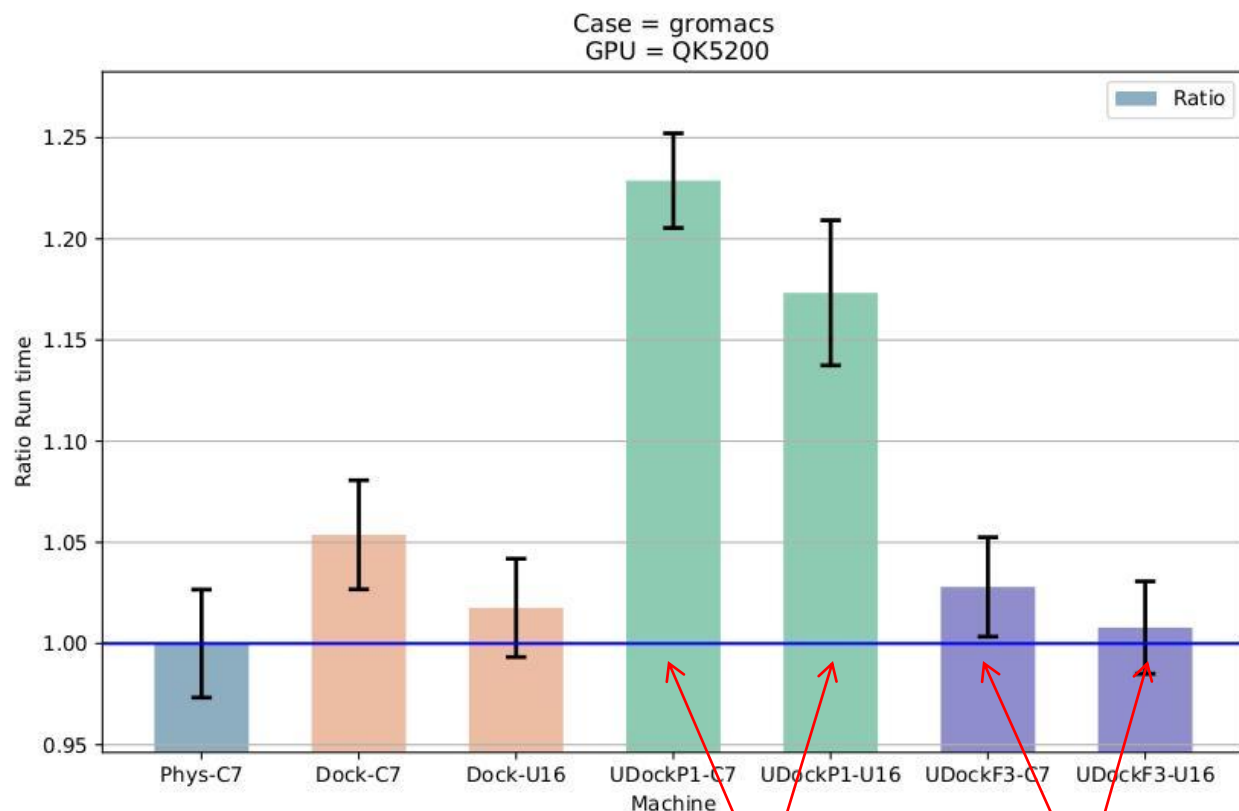
Performance with docker and udocker are the same and very similar to the host.

Using OpenCL and NVIDIA GPGPUs

Better performance with Ubuntu 16 container

udocker in P1 mode

udocker & Molecular dynamics



PTRACE

SHARED LIB CALL

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

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

Using OpenCL and OpenMP

udocker in P1 mode
udocker in F3 mode

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,
legacy code

udocker in P1 mode

udocker & Phenomenology

```
export MASTERDIR=/gpfs/csic_users/userabc/mastercode  
export UDOCKER_DIR=$MASTERDIR/.udocker
```

```
udocker.py run --hostauth \  
    -v /home/csic/cdi/ica/mcpp-master \  
    -v /home/csic/cdi/ica \  
    -user=user001 \  
    -w /home/csic/cdi/ica/mcpp-master mastercode \  
    /bin/bash -c "pwd; ./udocker-mastercode.sh"
```



UDOCKER
BE ANYWHERE

udocker

- Docker compatibility
- User oriented
- Independence from host system
- Does not require additional software
- Does not require recompilation
- Does not require administrator intervention
- Does not require privileges
- Does not require Linux namespaces

Thank you

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



udocker: list local images

```
$ udocker images
```

```
REPOSITORY  
msoffice:lastest .  
iscampos/openqcd:latest .  
fedora:25 .  
docker.io/susymastercode/mastercode:latest .  
ubuntu:14.04 .  
ubuntu:16.10 .  
ubuntu:latest .  
indigodatacloud/disvis:latest .  
jorge/private:latest .  
busybox:latest .  
jorge_fedora22_32bit:latest .  
debian:oldstable .
```


udocker: list containers

```
$ udocker ps
```

container-id	alias	image
CONTAINER ID	P M NAMES	IMAGE
9fe2f9e7-ce37-3be5-b12d-829a3236d2a6	. W ['ub14']	ubuntu:14.04
5c7bd29b-7ab3-3d73-95f9-4438443aa6d6	. W ['myoffice']	msoffice:lastest
676eb77d-335e-3e9a-bf62-54ad08330b99	. W ['fedora_25']	fedora:25
c64afe05-adfa-39de-bf15-dcd45f284249	. W ['debianold']	debian:oldstable
7e76a4d7-d27e-3f09-a836-abb4ded0df34	. W ['ubuntu16', 'S']	ubuntu:16.10
9d12f52d-f0eb-34ae-9f0e-412b1f8f2639	. W ['f25']	fedora:25