easybuild

3rd EasyBuild User Meeting
31 Jan 2018 – Amsterdam, NL

udocker

Running containers everywhere

Luís Alves
lalves @ lip.pt

udocker main developer:
Jorge Gomes - jorge @ lip.pt

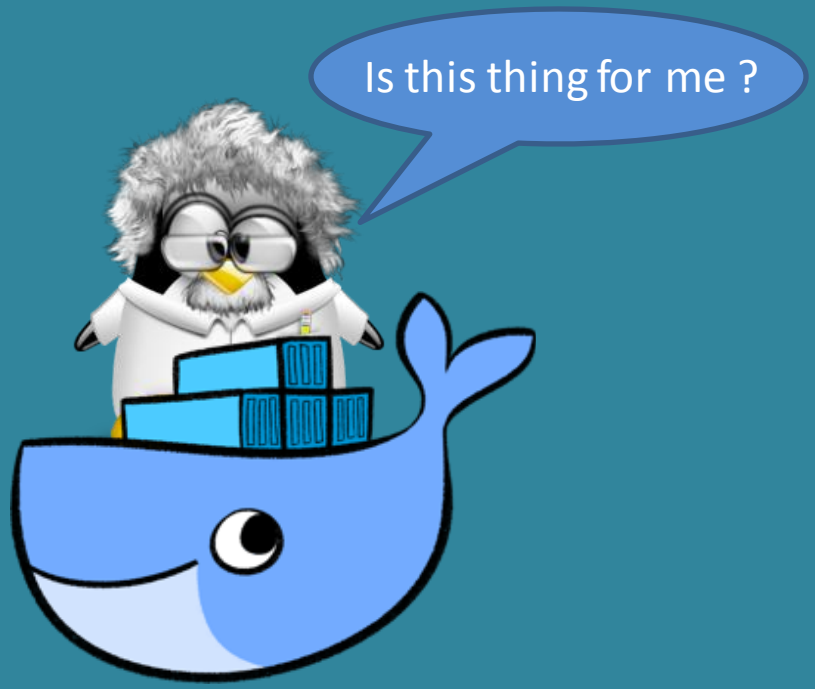
udocker is a tool to run containers
in user space

without Docker
without privileges
without sysadmin assistance

udocker empowers users
to run applications
encapsulated in Docker containers

but can be used to run any container that does not require root privileges

Scientific Computing and Containers



Scientific Computing

Running applications still requires considerable effort

Computers
- Several computing systems
- Notebooks, 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

Scienti mputing

Need a consistent portable way of running applications

udocker

INDIGO - DataCloud
Better Software for Better Science

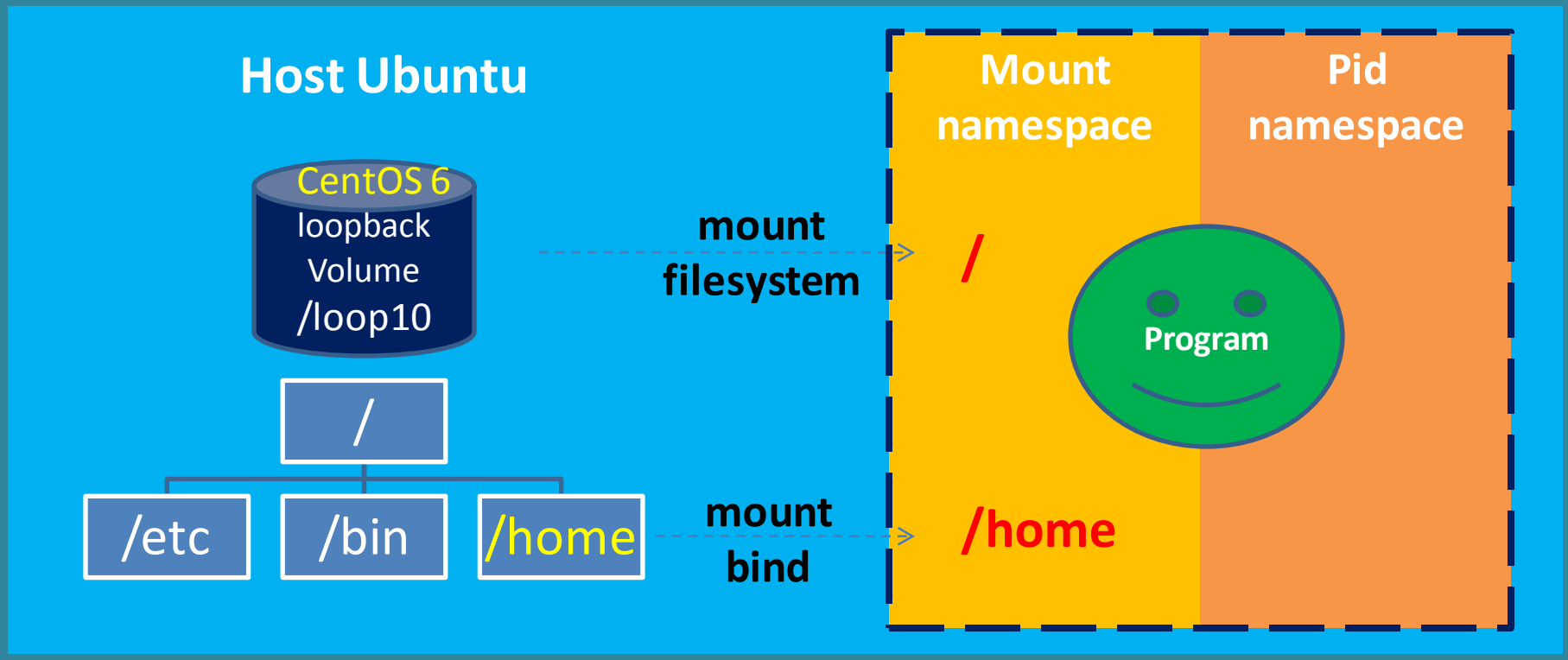
Containers take advantage of several kernel features

- **Kernel namespaces:** isolate system resources from process perspective
  - **Mount namespaces:** isolate mount points
  - **UTS namespaces:** hostname and domain isolation
  - **IPC namespaces:** inter process communications isolation
  - **PID namespaces:** isolate and remap process identifiers
  - **Network namespaces:** isolate network resources
  - **User namespaces:** isolate and remap user/group identifiers
  - **Cgroup namespaces:** isolate Cgroup directories
- **Seccomp:** system call filtering
- **Cgroups:** process grouping and resource consumption limits
- **POSIX capabilities:** split/enable/disable root privileges
- **chroot:** isolated directory trees
- **AppArmor and SELinux:** kernel access control

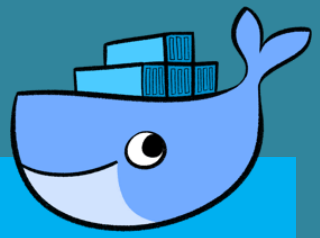
Container simply isolates processes

- Cannot see other processes running in the same host
- Restricted to their own file-system tree
- Can only see a subset of the host devices
- Cannot see others' shared memory, semaphores, queues
- Can be subject to strong resource usage limits
- Cannot invoke certain system calls
- Do not have access to all root capabilities/privileges

Container putting it together

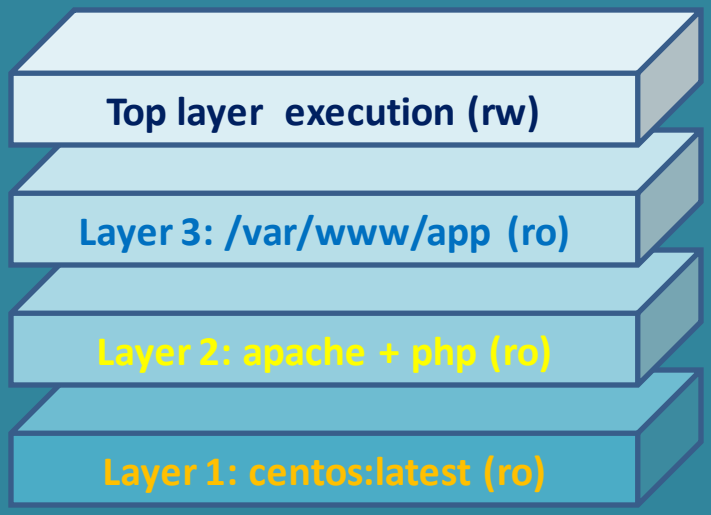


Docker images



- **Common format to distribute and manage images:**
  - Layered file-system based
    - At the host level implemented by AUFS, device-mapper thin snapshots
  - New images can be easily created from existing ones
    - Created by using **Dockerfiles** and docker build

Layers



Dockerfile

```
FROM centos:centos6
RUN yum install –y httpd php
COPY /my/app /var/www/app
EXPOSE 80
ENTRYPOINT /usr/sbin/httpd
CM D[“-D”, “FOREGROUND”]
```

Container putting it together

To create a container image:

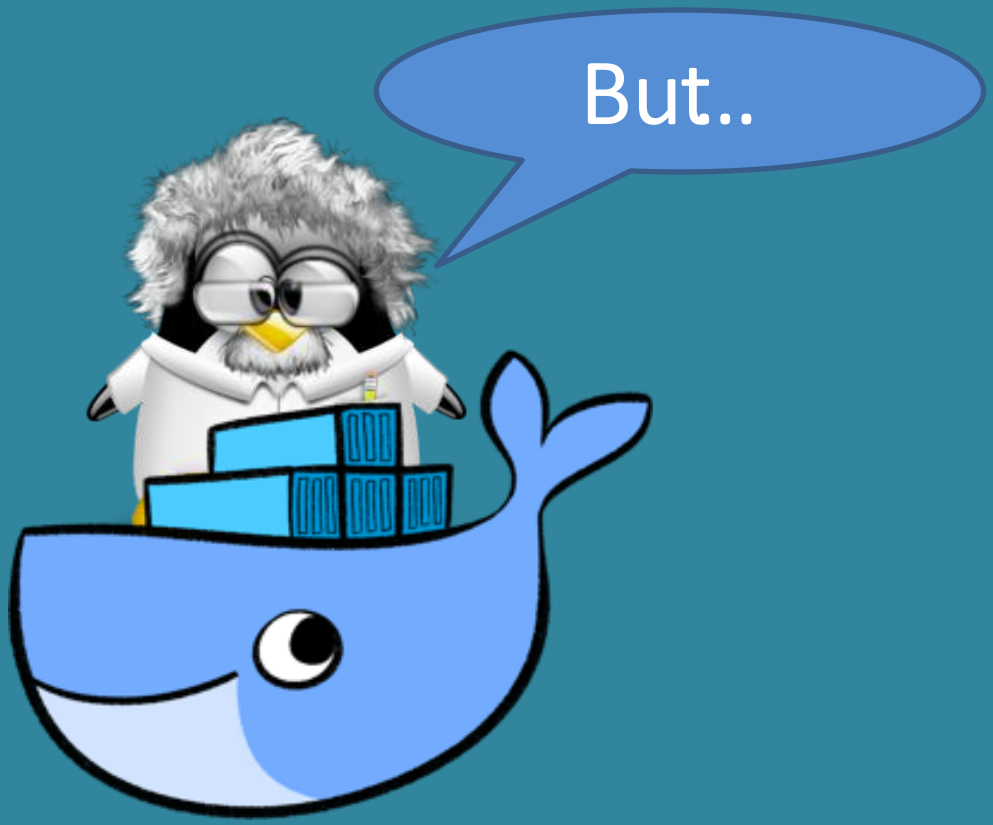
- Add the required libraries and programs to the container file-system
- Allow container access to the host required volumes and/or devices

Can I run another Linux distribution using containers ?

- **Yes, sure!**
- **The Linux kernel ABI remains largely unchanged across versions**

Containers are usually started by the root user:

- Some operations require privileges
- Can be root user inside a container without affecting the host or the other containers (with POSIX capabilities, seccomp and namespaces)


But..

Docker limitations (on this context)

Require root privileges to install, setup and run

- Security concerns especially in multi-user environments

Docker API does not limit privileged actions

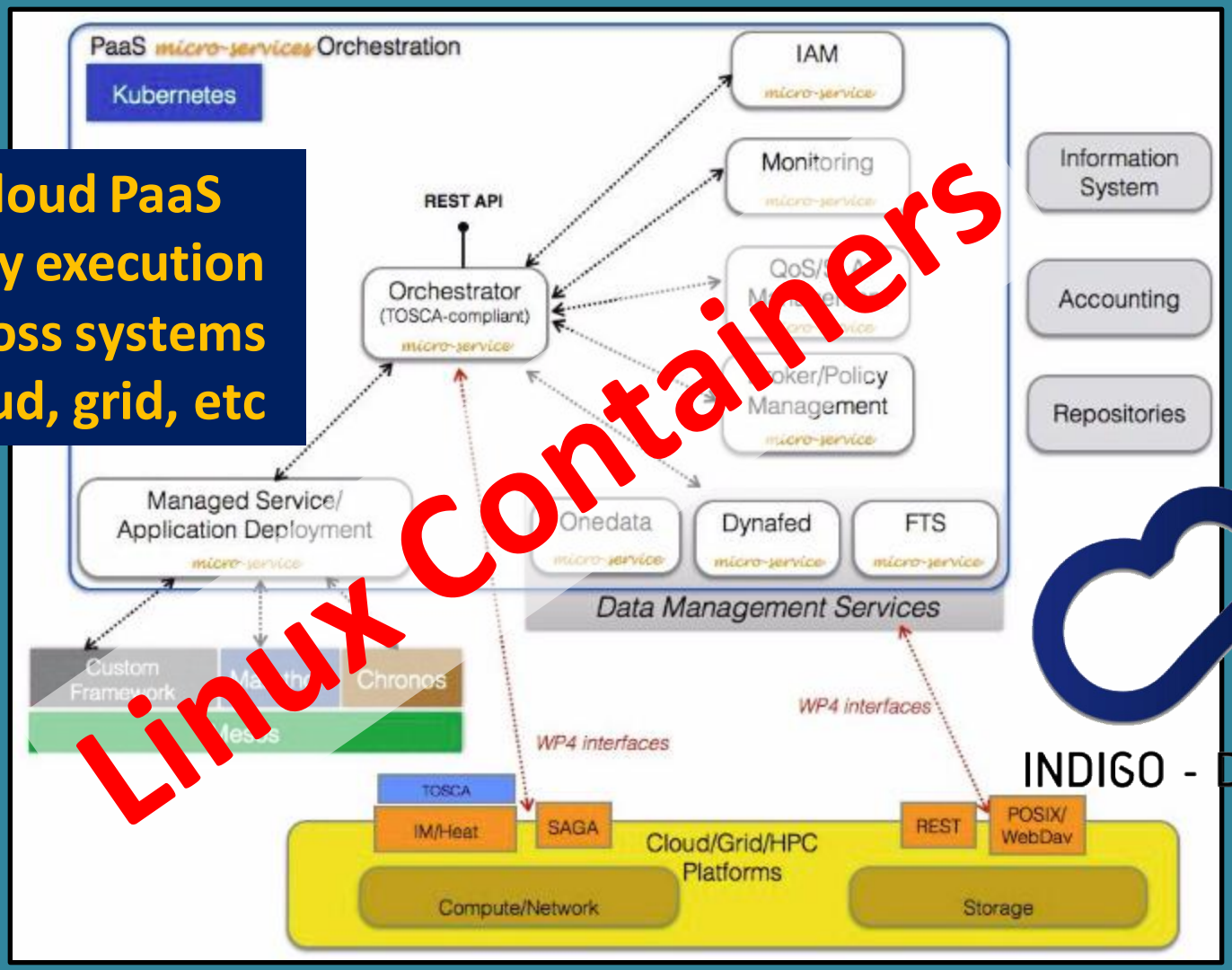
- Users with direct access to the API can do anything
- e.g: through the API users can mount local file systems, make devices accessible, erase disks etc.

Limiting design decisions for end users

- Docker is designed to be used as an hypervisor by operators
- Difficult to use on batch systems because of process control and security (not suitable)

INDIGO-DataCloud H2020 (**2015**-2017)

Cloud PaaS
easy execution
across systems
cloud, grid, etc



udocker

INDIGO - DataCloud
Better Software for Better Science

Linux containers on batch systems

- How to run Docker in batch systems ?
  - Can we run Docker in batch system ?
  - If so how to integrate it with the batch system ?
  - How to make it respect batch system policies ?
  - How to make it respect batch system actions ?
  - How to collect accounting ?

bdocker

- How to run containers without Docker ?
  - Can we download container images ?
  - Can we run without a layered filesystem ?
  - Can we run them as normal user ?
  - Can we enforce container metadata ?

udocker

Then udocker must...

- 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

udocker is not bound to a single Linux kernel technology such as Linux namespaces

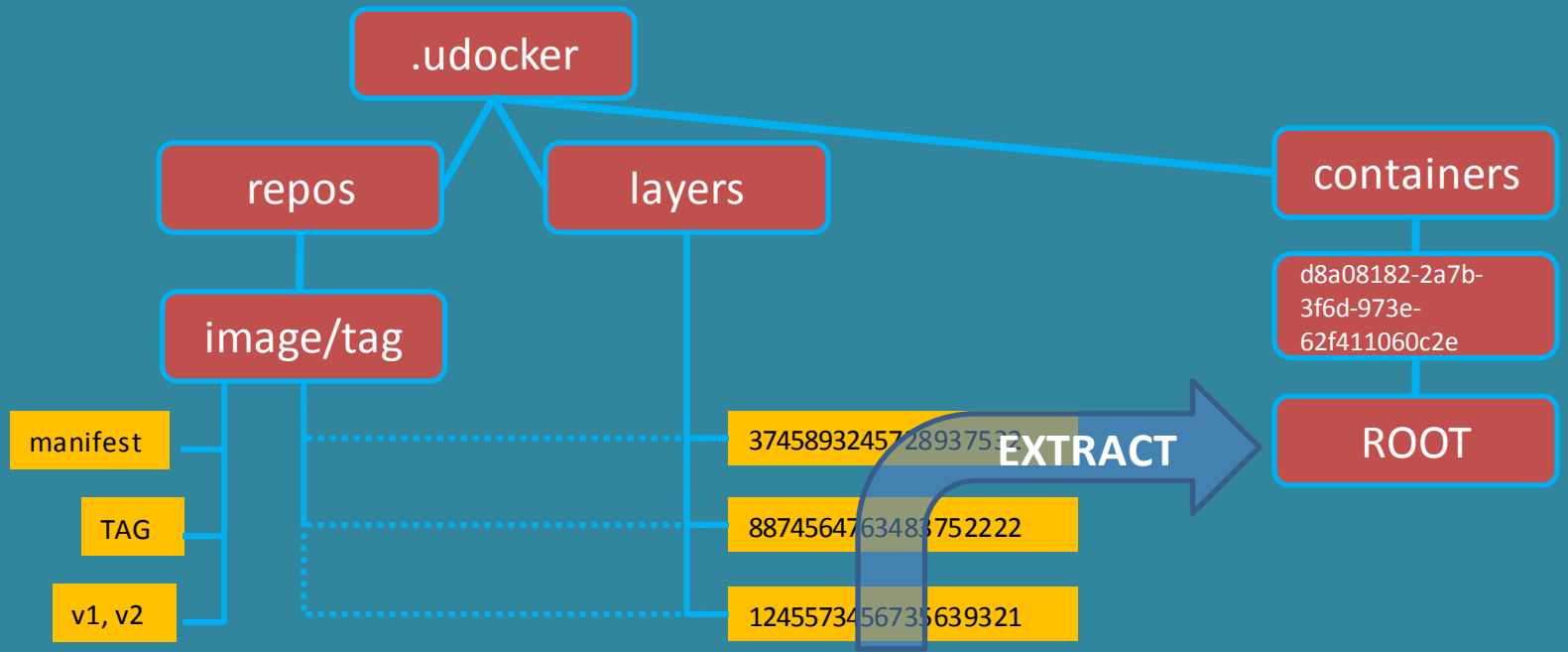
udocker can run containers in:
CentOS 6 systems
older 2.6 Linux kernels
well suited for computing farms and grids

udocker

- Written in
  - **Python**, C, C++, Go
- Runs in:
  - CentOS 6, CentOS 7, Fedora >= 23
  - Ubuntu 14.04, Ubuntu 16.04
  - Any distro that supports python 2.7
- Features:
  - Command line interface Docker like
  - Pull of containers from Docker Hub
  - Local repository of images and containers
  - Execution of containers with modular engines

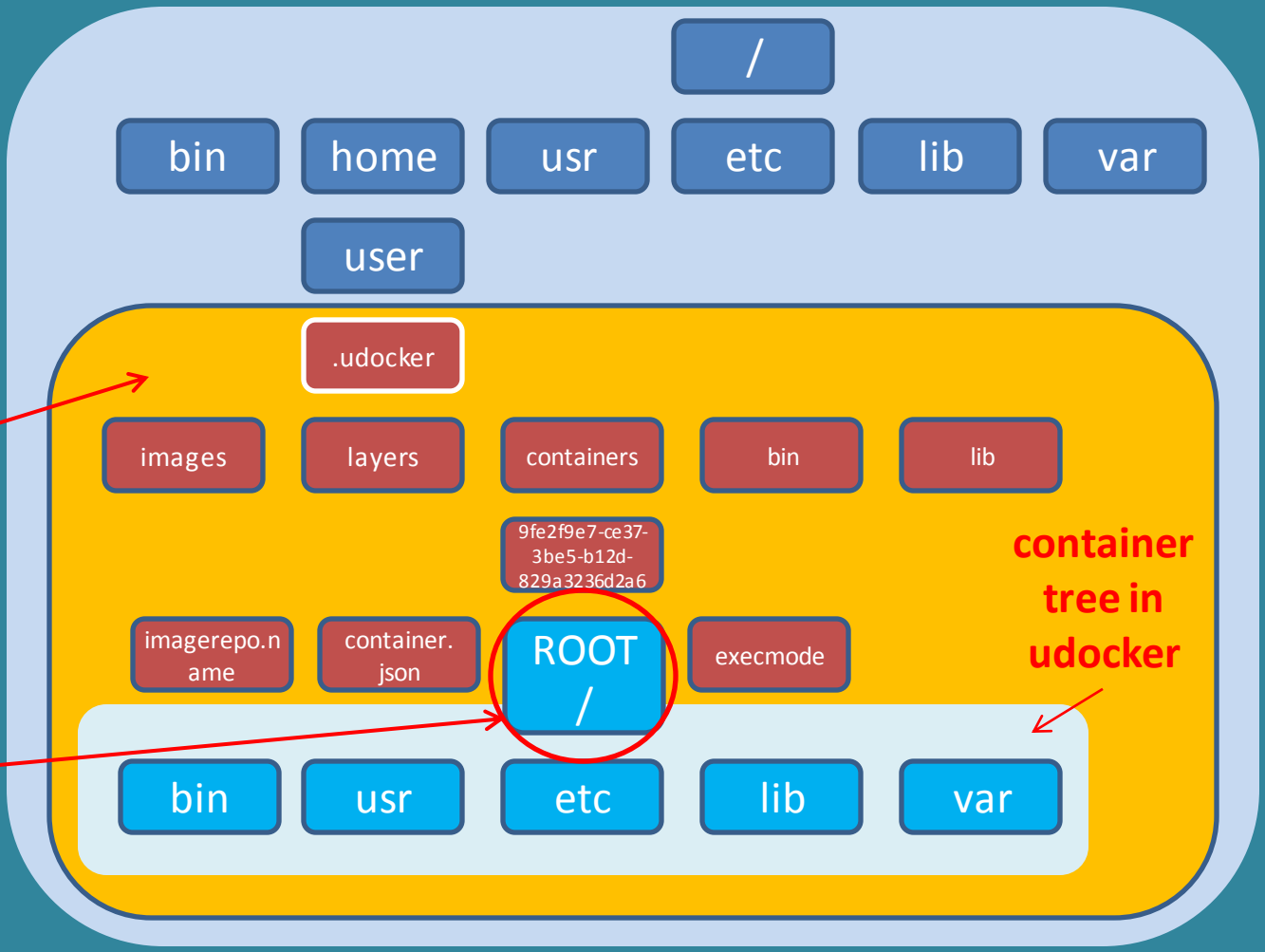
udocker containers

- 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



Directory Tree – Container and Host

Execution chroot-like



Execution

- The run command provides a chroot like environment to execute programs in the flattened directory trees
- The run command does not provide isolation between the container and the host, only mimics a chroot
- No privileges are involved therefore actions that only the root user can perform will likely not work within udocker

Execution Engines

- **ptrace** (default)
  - Intercepts system calls and changes pathnames
  - Works in most circumstances but can be slow
- **library call interception**
  - Override of shared library functions
  - Host performance in most cases faster than ptrace
- **rootless namespaces**
  - Uses namespaces but without root privileges
  - Closest to what Docker does

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 syntax

- mimics as much as possible Docker syntax

udocker search
udocker pull
udocker create
udocker run
udocker import
udocker load
udocker images
udocker rmi
udocker ps
udocker rm

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: search images

```
$ udocker search ubuntu
```

```
NAME                    OFFICIAL DESCRIPTION
ubuntu                  [OK] Ubuntu is a Debian-based Linux operating system based on free
ubuntu-debootstrap      [OK] debootstrap --variant=minbase --components=main,universe
ubuntu-upstart          [OK] Upstart is an event-based replacement for the /sbin/init daemon
rastasheep/ubuntu-sshd ---- Dockerized SSH service, built on top of official Ubuntu images.
```

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: 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: create container from image

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

container-alias

```
9fe2f9e7-ce37-3be5-b12d-829a3236d2a6
```

container-id

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
```

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 container as yourself

```
$ udocker run  --user=jorge  -v /home/jorge  \
  -e HOME=/jorge/home  --workdir=/home/jorge  ub14
```

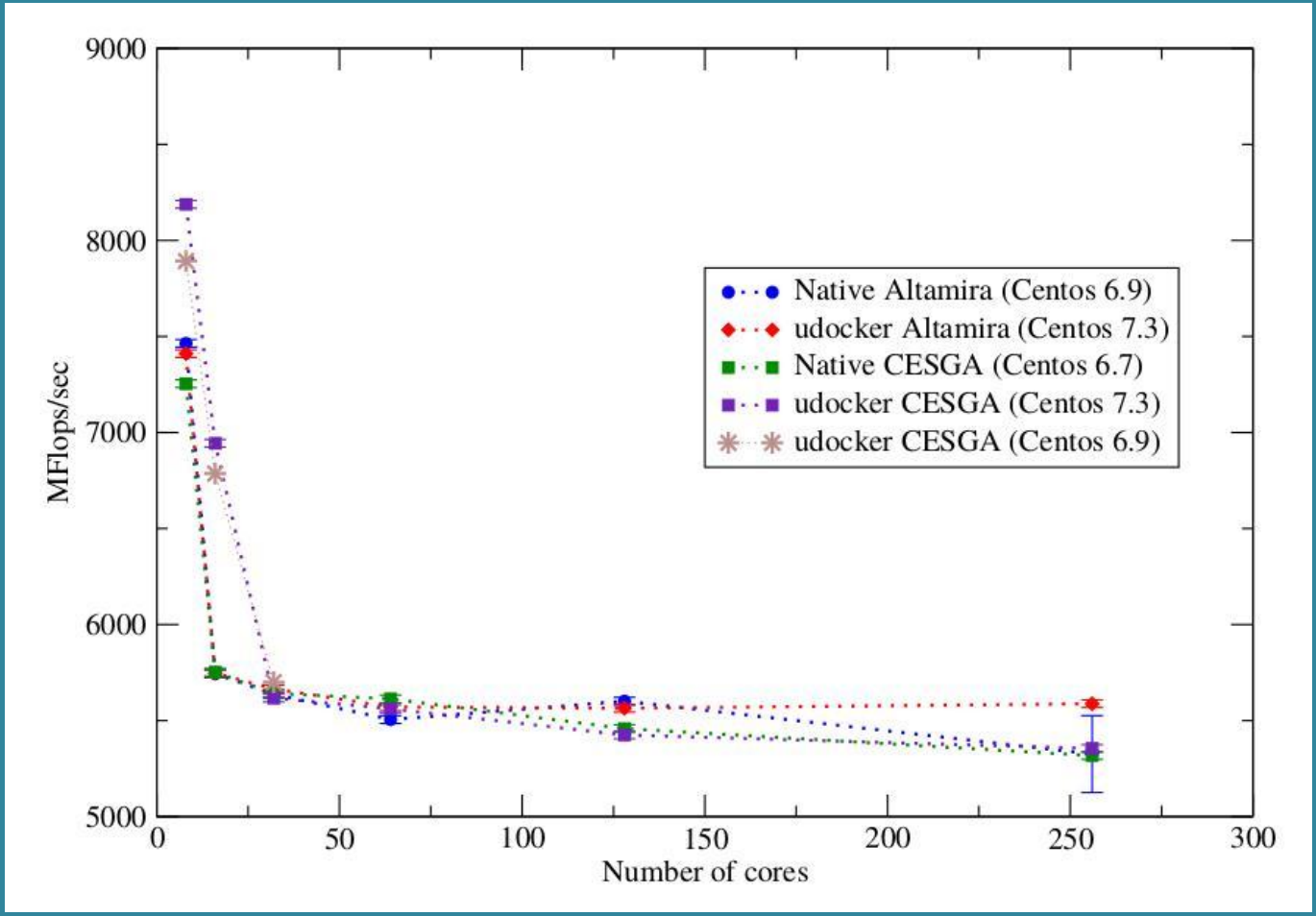
```
Warning: non-existing user will be created

********************************************************************************
*                                                                              *
*            STARTING 9fe2f9e7-ce37-3be5-b12d-829a3236d2a6                     *
*                                                                              *
********************************************************************************
executing: bash
jorge@nbjorge:~$ id
uid=1000(jorge) gid=1000(jorge) groups=1000(jorge),10(uucp)
jorge@nbjorge:~$ pwd
/home/jorge
jorge@nbjorge:~$
```

Issues and limitations

- ptrace
  - Performance varies across applications
  - Relies on complex stack rewriting
- library call interception
  - Does not offer root emulation or capabilities
  - Issues with applications that also perform themselves shared library call interception
- rootless namespaces
  - Some system calls will not work properly due to the kernel
  - Some functionalities available in the other methods do not work with rootless namespaces
  - Recent and not extensively tested

udocker & Lattice QCD



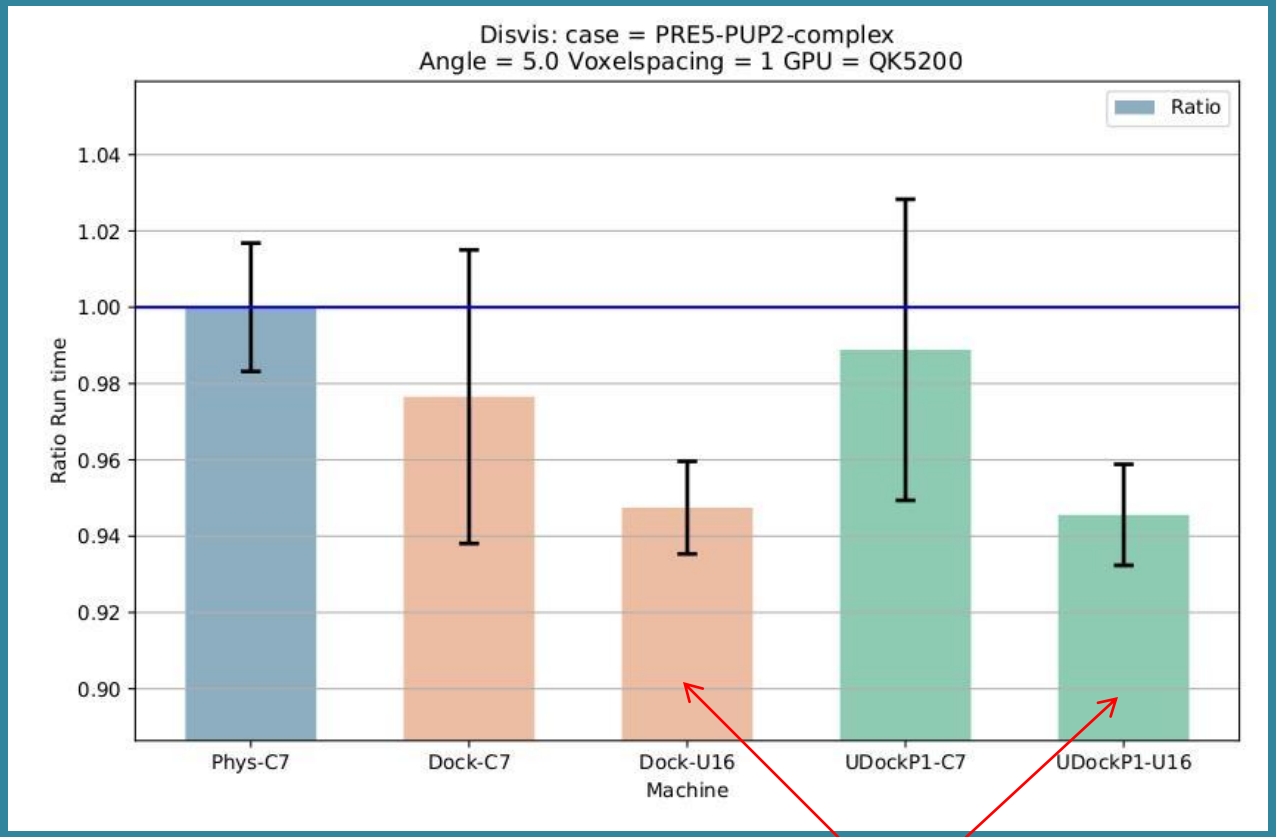
OpenQCD is a very advanced code to run lattice simulations

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

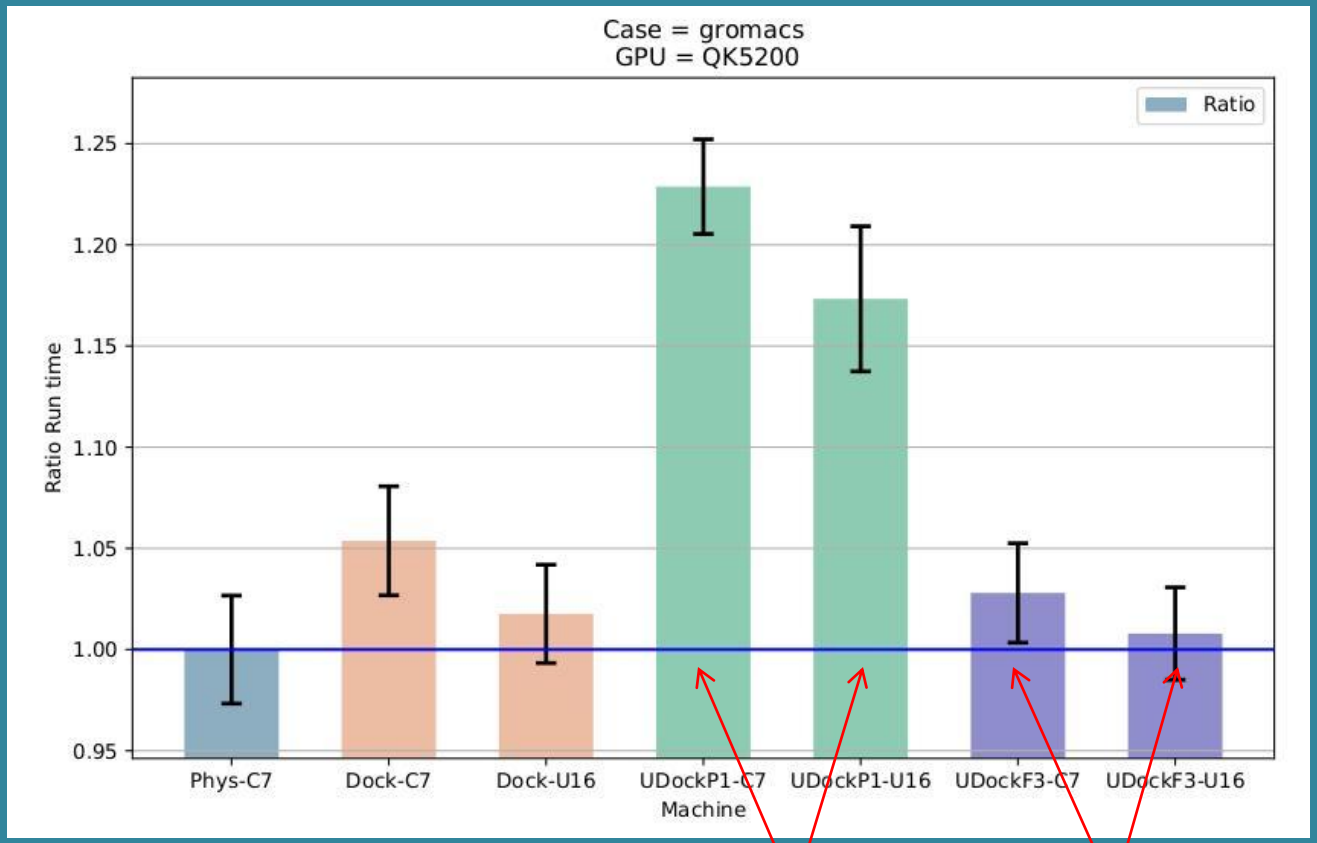
Performance with docker and udocker are the same and very similar to the host.

Using OpenCL and NVIDIA GPGPUs

udocker in P1 mode

Better performance with Ubuntu 16 container

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 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

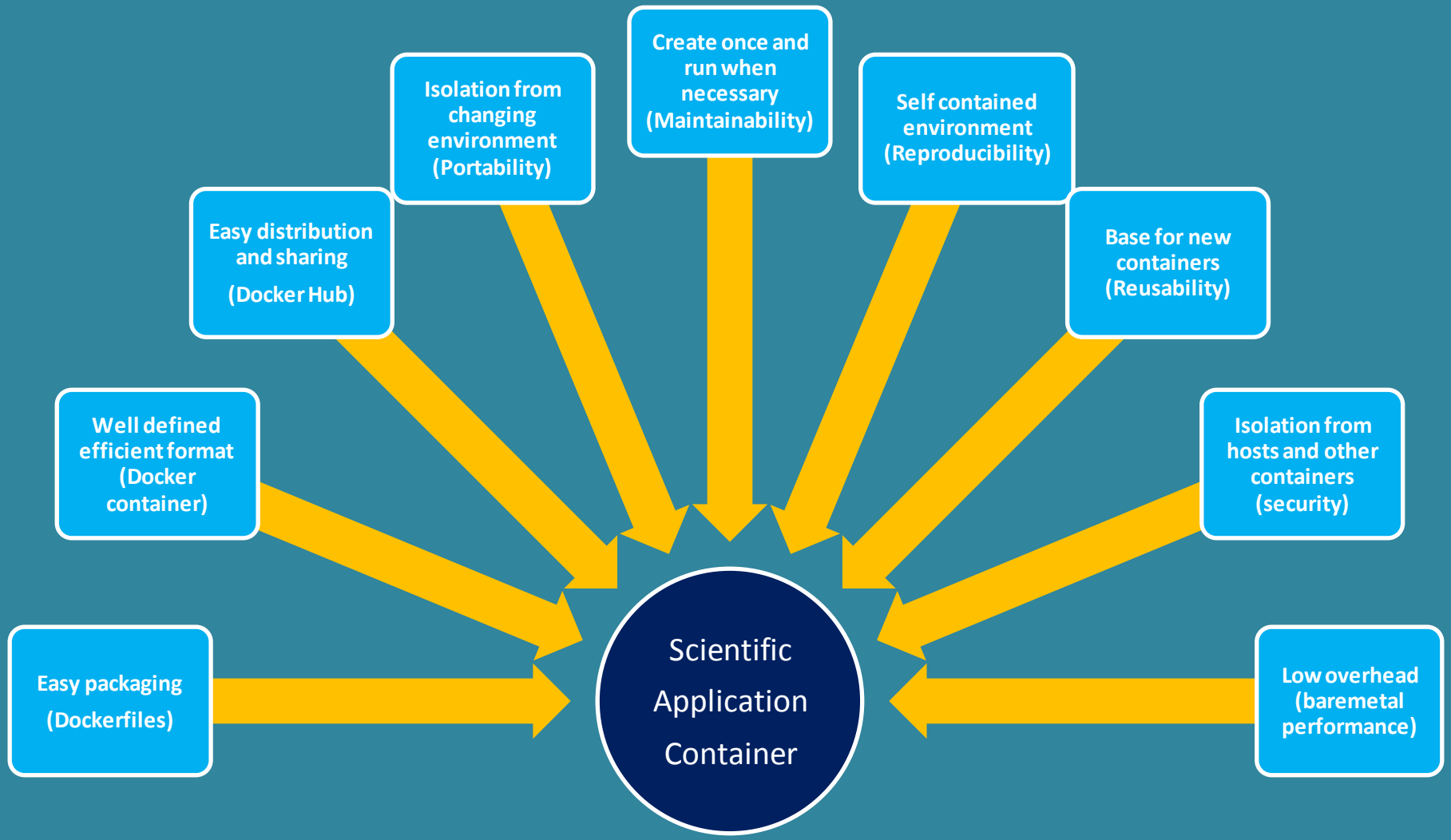
udocker in P1 mode

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"
```

Scientific computing and containers



udocker Publication

https://goo.gl/UZG35s

easybuild

3rd EasyBuild User Meeting
31 Jan 2018 – Amsterdam, NL

Thank You

See more || get || give feedback || contribute:

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

Infraestrutura Nacional de **Computação Distribuída**

LABORATÓRIO DE INSTRUMENTAÇÃO E FÍSICA EXPERIMENTAL DE PARTÍCULAS

udocker

INDIGO - DataCloud
Better Software for Better Science