Karlsruher Institut für Technologie

# NVIDIA GPGPU within a Docker environment

**Andrei Shkarin**
andrei.shkarin@kit.edu

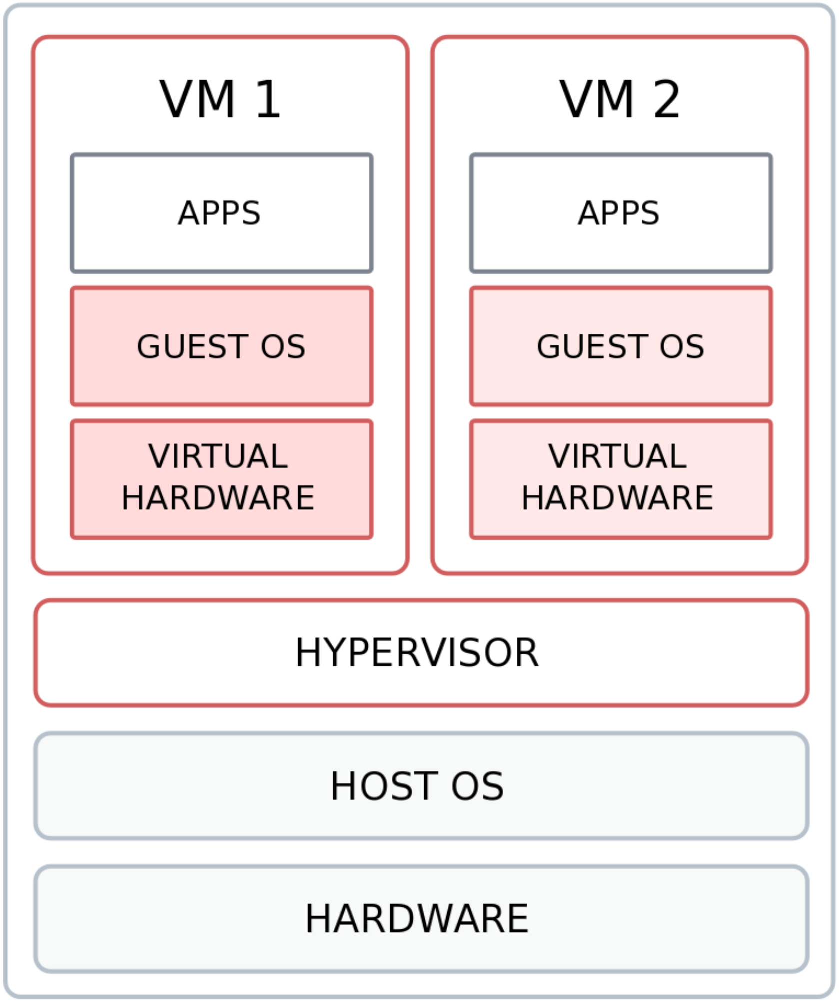
KIT, Institute for Data Processing and Electronics (IPE)

# Outline

- Virtualization
- Example
- The Docker
  - Structure of a container image
  - Docker and NVIDIA GPU
- Demonstration
  - Installation and Commands
  - Play with container

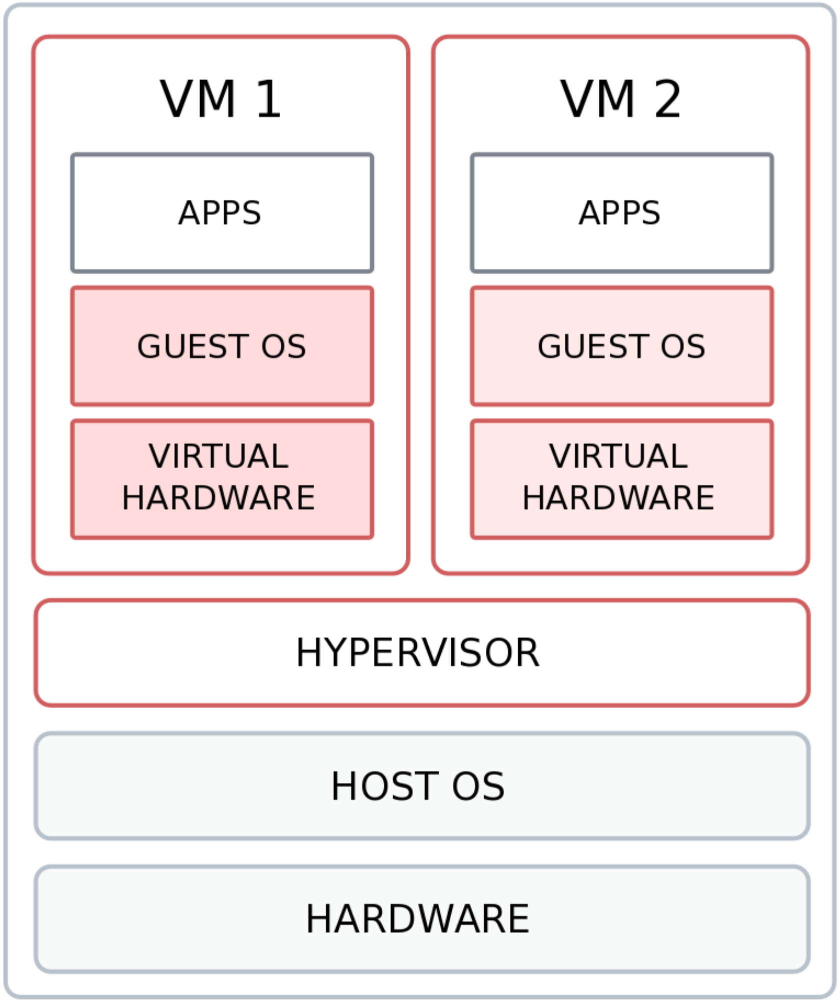
# Virtualization

## Hardware level

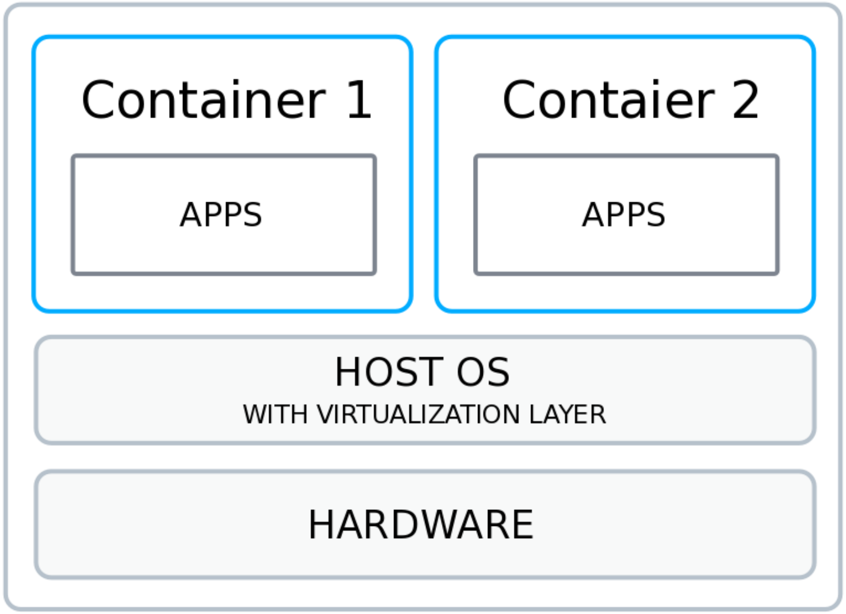


# Virtualization

## Hardware level



## Operation System level



# Example: Volunteer computing

Project:

Cosmology@Home

VM Image
369 MB

VM-based application

Application inside a VM

Positive:

- Can be run on most of the systems;
- Easy to suspend, resume or make a snapshot;

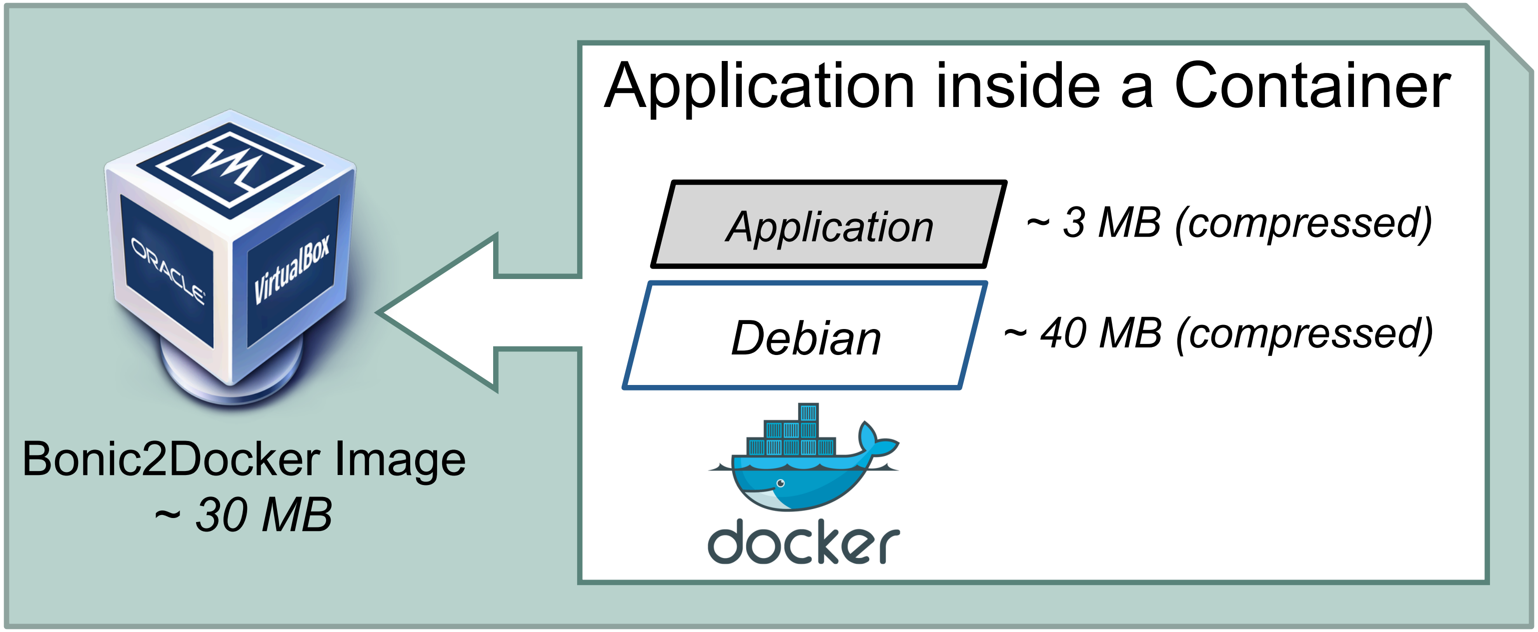
Negative:

- Hard to update the application.

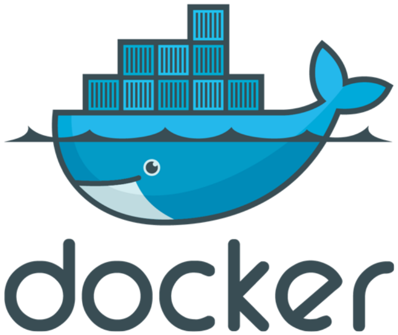
# Example: Volunteer computing

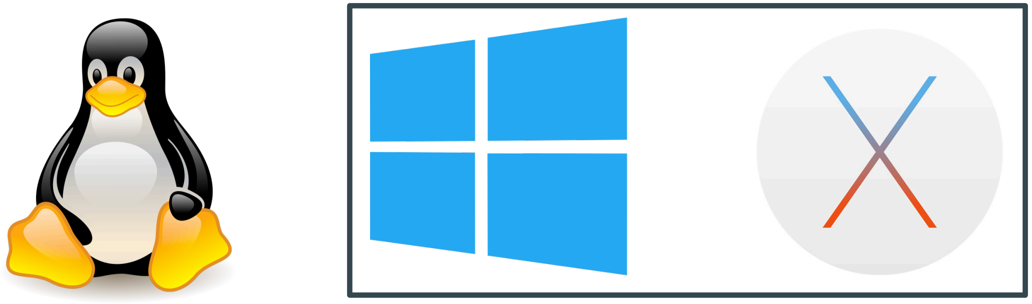
Project:

Cosmology@Home

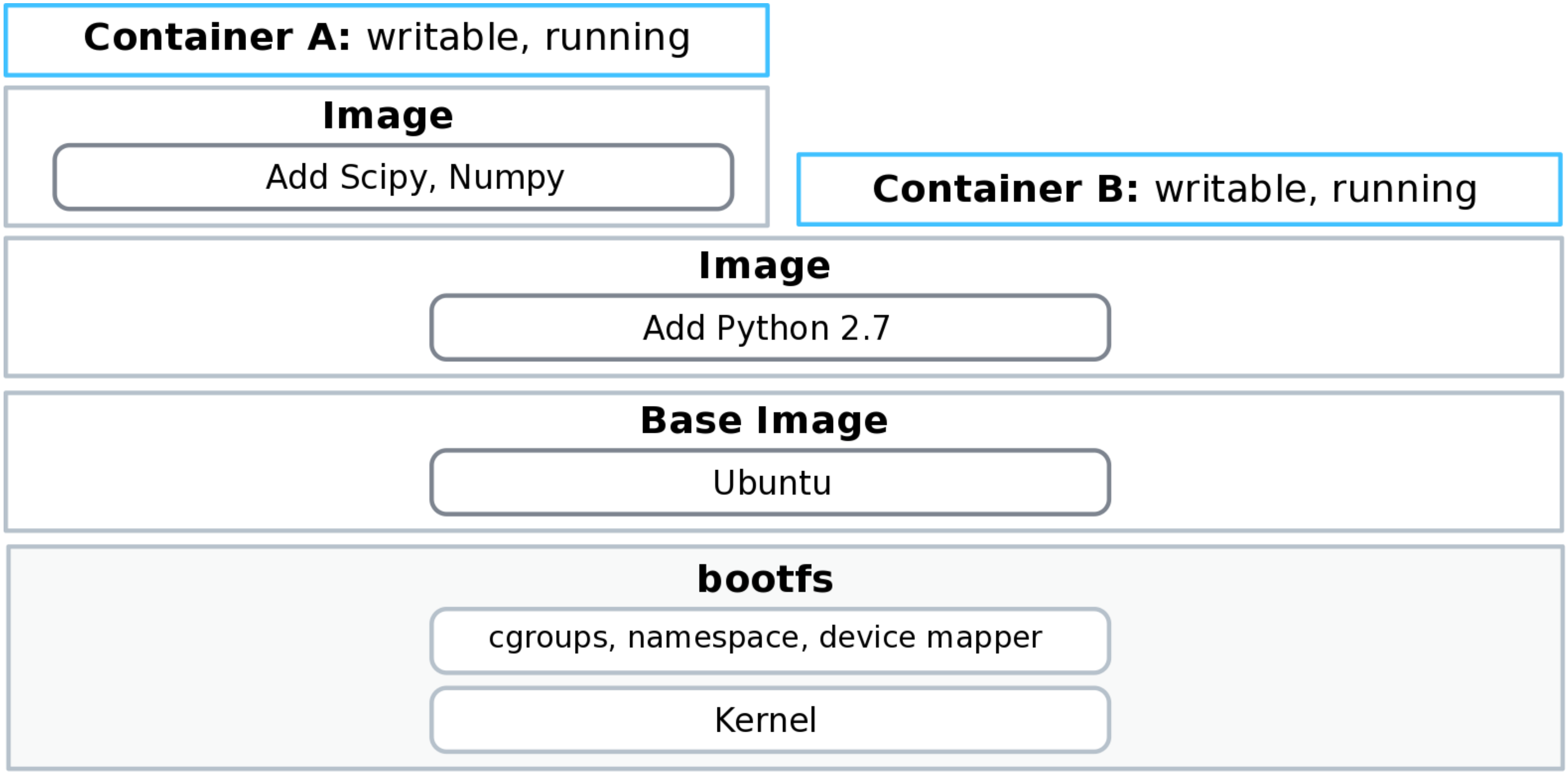


VM-based application





Containers run in a Virtual Machine



**Container A:** writable, running

**Image**

Add Scipy, Numpy

**Container B:** writable, running

**Image**

Add Python 2.7

**Base Image**

Ubuntu

**bootfs**

cgroups, namespace, device mapper

Kernel

**To make a GPU available within a Container:**

- Mount related device file into the Container;
- Mount libraries related to the driver into the Container;
- CUDA installed in the container must be compatible with driver at host.

**Can containers share a GPU?**

- Yes, as like they would be processes running at the host.

*https://github.com/NVIDIA/nvidia-docker*

# Benefit

- Optimization of the disk usage;
- Process isolation and resource control;
- Using devices directly inside several virtualized environments;
- Improving user and developer experience.

# DEMONSTRATION

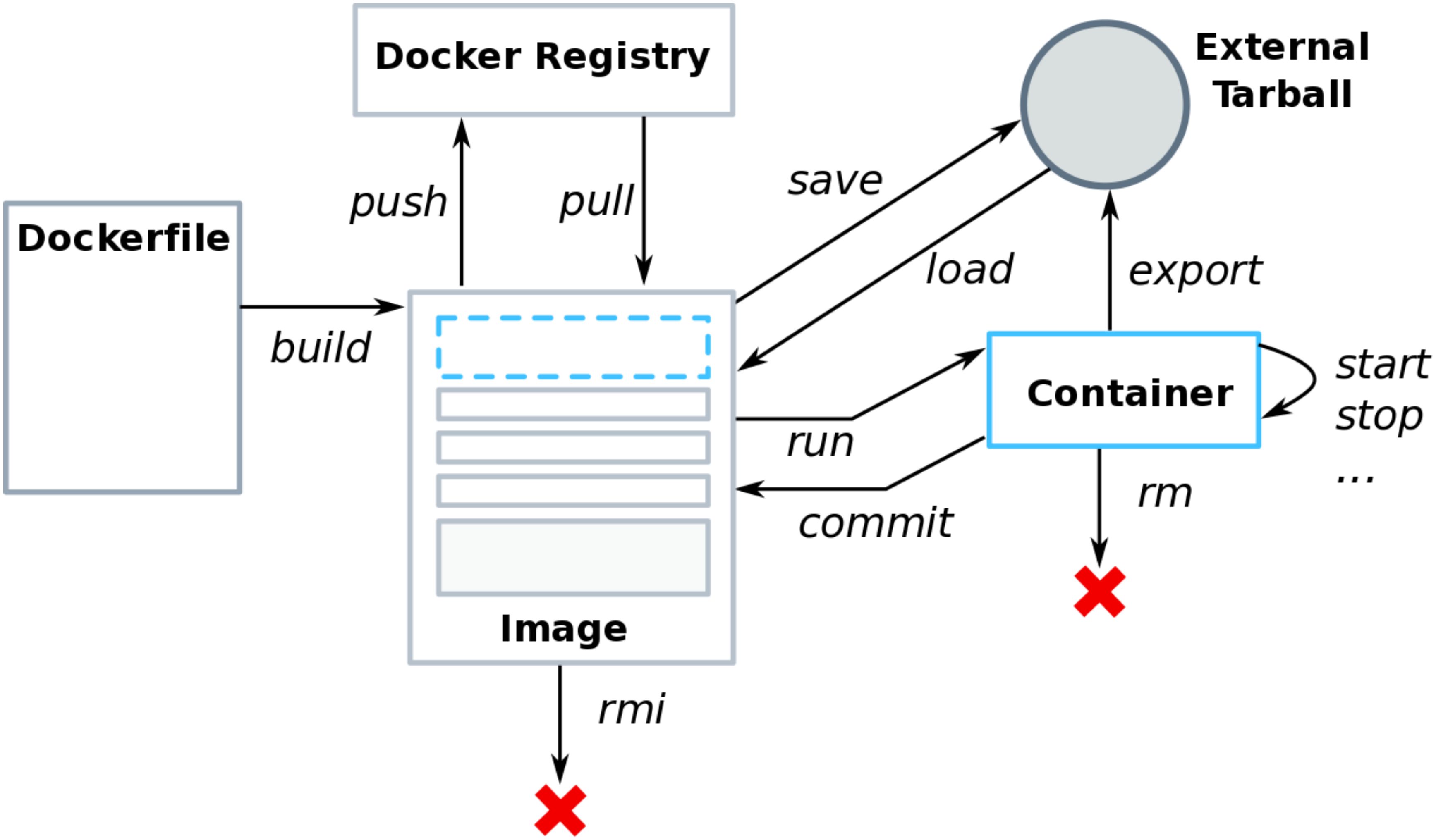
# Download

- Mac, Windows, Linux: *https://docs.docker.com/linux/*

- If your OS is not supported:

  Binary of latest release from:
  *https://github.com/docker/docker/releases*

# Commands


Docker Registry
External Tarball
Dockerfile
push
pull
save
load
export
build
Container
start
stop
...
run
commit
rm
Image
rmi