CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich

Survey on Container technology
Shifter improvements

ADAC 5 - Tokyo

Maxime Martinasso, ETHZ/CSCS

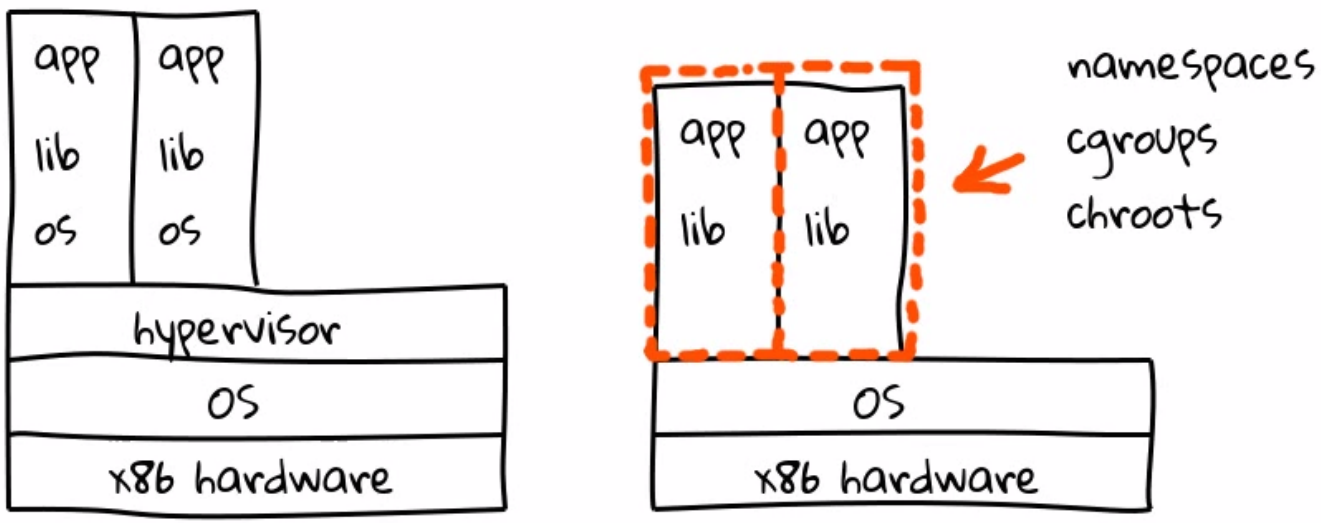
February 15, 2018

Containers

Why containers are of interest for HPC?

- Software stack current alternatives:
  - admin-managed pre-fixed list of software (modules)
  - self-compile by user (dependencies, maintainability)

→ Container provides an user-defined software stack



Containers

Container technology

- Kernel namespaces: isolates and virtualizes system resources, a process can only see the resource associated to its namespace (PID, FS, network,...)
- cgroups: control groups limits resource usage of a group of processes (CPU, memory,...)
- chroot(): changes the apparent root directory for a process

→ Container technology is a set of kernel features

Container workflow and HPC Centre

Build anywhere and run on HPC Centre!

Anywhere	create an image (use a file)
Anywhere	build an image (maybe not on HPC Centre)
Anywhere/HPC Centre	publish image in repository (optional) or move image to HPC center
HPC Centre	get and deploy image on compute nodes (using a scheduler plugin/orchestration service)
HPC Centre	run a command in batch or interactive shell

Containers for HPC

Container requirements for HPC

- Access specific devices (GPU, FPGA,...)
- Use vendor specific kernel drivers (NVidia driver,...)
- Use vendor specific software stack (MPI, scientific library,...)

Container challenges

- Architecture specific (x86/x86_64)
- Limited portability: kernel, drivers
- Managing file paths requires extra effort (volume binding)
- Replace on-demand software stack in an image

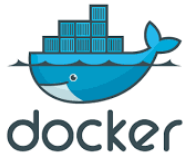
Containers survey with focus on HPC

CharlieCloud

Singularity

- Enterprise focus versus HPC focus

- Security
  - Limit privileged access
- Usability
  - Build/deploy container
- Maintainability
  - Integration into HPC stacks
- Performance
  - Access to HPC HW & SW



Docker

- Widely used, focus on enterprise micro-services application
- Docker engine: can use other engines (LXC)
- Builds image from a file
- Manages images: DockerHub or save/load images as tarball

→ Well-designed and documented way to manipulate images

- Root access: mapping + network isolation = poor perf.
- Image needs to be loaded at each compute nodes
- Needs daemons (root) on nodes
- Complex inter-operability with parallel FS

 Security  Maintainability  Usability  Performance

rkt

- Meant to inter-operate with other containers
  - Complex network configuration
- Images are compatibile with Docker
- Different levels of isolation: coreos, host, fly and kvm

- Focus on enterprise micro-services
  - Difficult to use MPI, GPUs and other HPC technology
- CoreOs has been bought by RedHat
  - Will rkt be integrated in OpenShift?

 Security  Maintainability  Usability  Performance

LXC/LXD

LXC - Container engine

- Set of tools to control containers
- Provides API: C, Python, ...

LXD - Container system

- Uses of pre-existing images (complex to create new image)
- Scalable design for deployment on thousand of nodes
- Integration with OpenStack + REST API
- GPU access support and docker compatibility

→ Focus on enterprise but a lot of potential for HPC

 Security  Maintainability  Usability  Performance

Singularity

- Focus on HPC
- Minimalist container: includes just the app and dependencies
- Packaging over portability on different systems
  - Need to rebuilt containers after a system upgrade?
- Runs as much possible as a user process
  - Built-in support for GPU
- Can import Docker image
- Binds and mounts path into the container
- Works with Slurm

- New company: Sylabs Inc.
- Wants to re-create an eco-system - new image format


Security

Maintainability

Usability

Performance

CharlieCloud

- Focus on HPC
- Lightweight implementation (about 900 LOC)
- Minimalistic set of features - container core functionality
  - Hacker friendly
- Binds and mounts path/devices into the container
- Binds and mounts configuration of the host (if required)
- Compatible with Docker image and DockerHub
  - Builds image in a sandbox


Security

Maintainability

Usability

Performance

Shifter

- Focus on HPC
- Imports image from DockerHub
  - Converts Docker image (removes root, replaces passwd/group)
  - Uses a gateway architecture and an image repository
- Chroot docker image in RO mode (Squash-FS)
- Integration with Slurm by using a SPANK plug-in
- Binds and mounts paths into the container
- Transparent access to GPU
- Transparent MPI library swapping (ABI)
- Supports Cray systems

 Security  Maintainability  Usability  Performance

Shifter improvements

Image management

- Gateway architecture has been removed
  - User-based image storage
- Only one executable in user space (no root)
- Improve access to external repository
  - Use private repository from DockerHub
  - Connect to 3rd party repositories
- Faster image download
- Use image directly as a tarball

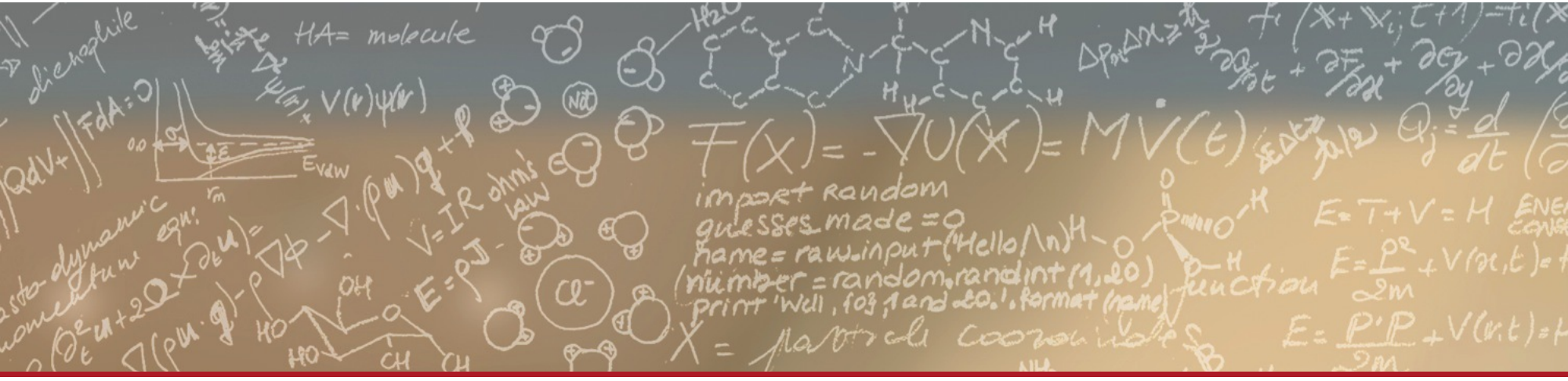
Improved functionality

- Similar CLI as Docker
- Writable volatile directory, better support of volume binding

Conclusion and container technology direction

Shifter, Singularity and LXC(?) are good container solutions for HPC very active community.

- Open Container Initiative (OCI)
  - Open industry standards around container formats and runtime
  - Docker, RedHat, Google, AWS, ...
  - Are HPC interests represented? focus is more on enterprise and cloud
- Container orchestration
  - Deploys and scales containers on multiple hosts
  - Kubernetes, SwarmKit, Univa Grid Engine
- Container for persistent storage
  - Container as a media to bundle and to access data
  - Moving data closer to the application
  - Software-defined storage



Thank you for your attention.

OCI members - Feb 2018