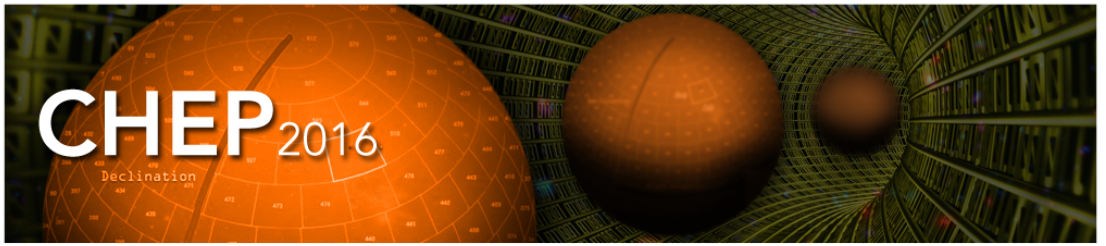
Consolidation of Docker use in HTC and its evolution to HPC at INFN-Pisa

S. Arezzini, A. Ciampa, A. Formuso, E. Mazzoni

October 12nd, 2016



22nd International Conference on Computing in High Energy and Nuclear Physics, Hosted by SLAC and LBNL, Fall 2016

INFN-Pisa in numbers

- 100sqm room
- About 10000 cores used both for HTC and HPC
- 2PB of disk storage in one GPFS file-system
- More than 20 Grid VO supported
- More than 200 users geographically distributed accessing the site and using resources
- Tier2 for CMS and Belle2

HW/SW ecosystem

Worker nodes HW of many different type

- CPU from Opteron 2218 to last Intel Xeon
- Memory/CORE 1GB → 16GB

HPC cluster

- Special networks (i.e. IB DDR or QDR)

Many types of software

- Standard HEP (i.e. LHC experiments)
- Open source theoretical software (i.e. gromacs) or user developed
- Commercial software (i.e. CFD)

How to manage this diversity in a single structure?

The solution

Standardize access to the resources

- Use only LSF both for batch and interactive

Standardize disk space

- GPFS: users data areas
- AFS: software distribution (system and groups/users)
- CVMFS: for Grid VO's software

Decoupling SysAdmin land and User land

- Needs of OpSys stable, certified for GPFS and HW
- Users needs environment certified for their software
- VM: high overhead in both CPU and memory, performances penalty (IB)
- Very light virtualization is the only way

First implantation in 2010 using chroot

The chroot implementation

The bare metal is the SysAdmin land

- Installed with SLES
- All file-system natively mounted

A chroot disk partition is the User land

- SL environment and user software
- All file-system mounted via *bind* inside chroot
- LSF services running inside chroot → user job lands inside it

It works, it is very light but

- You need to preload the User land on the system (tar.gz)
- Manage the chroot start/stop at boot time → in house scripts
- Mange the "images" life cycle → in house scripts
- Keep the chroot images on systems up to date → in house scripts

The system is too heavy to manage and prone to errors

docker

It is the natural evolution of chroot solution

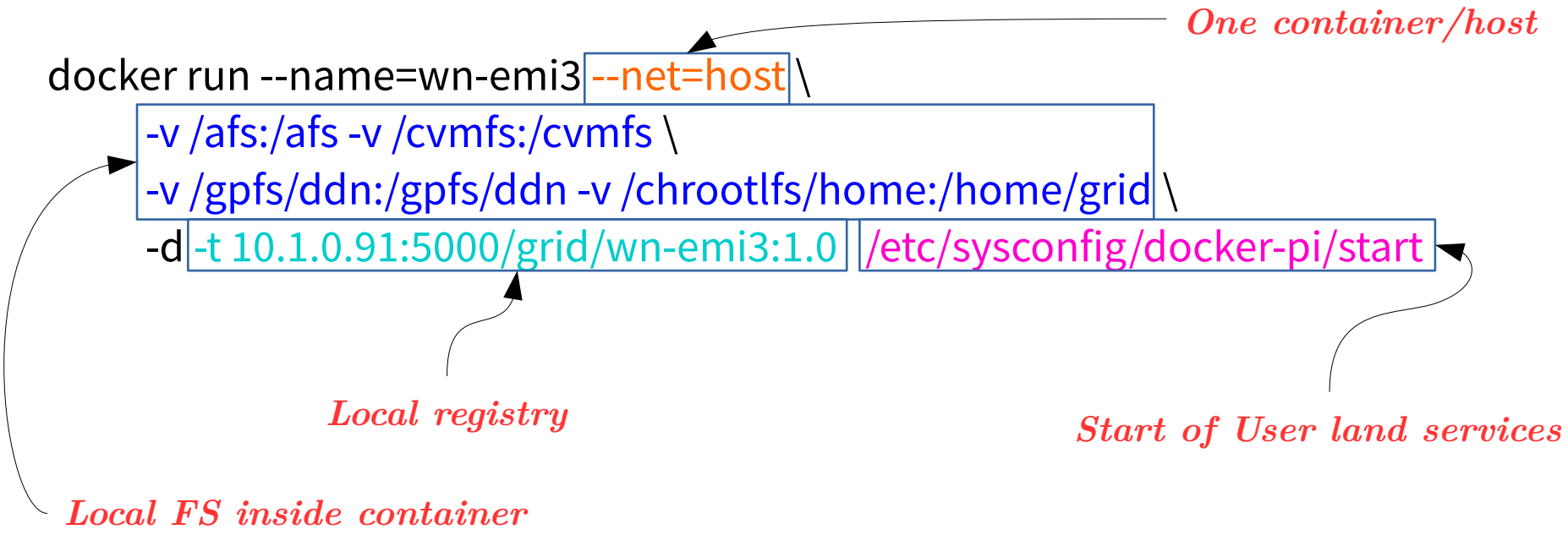
- We kept the good things of chroot (file-systems)
- No more tar files and complicated management
- Very simple images management, directly from your laptop
- Created local registry to avoid exposure of sensitive information
- We take advantage of GPFS to manage container on nodes

First container created from a chroot tar

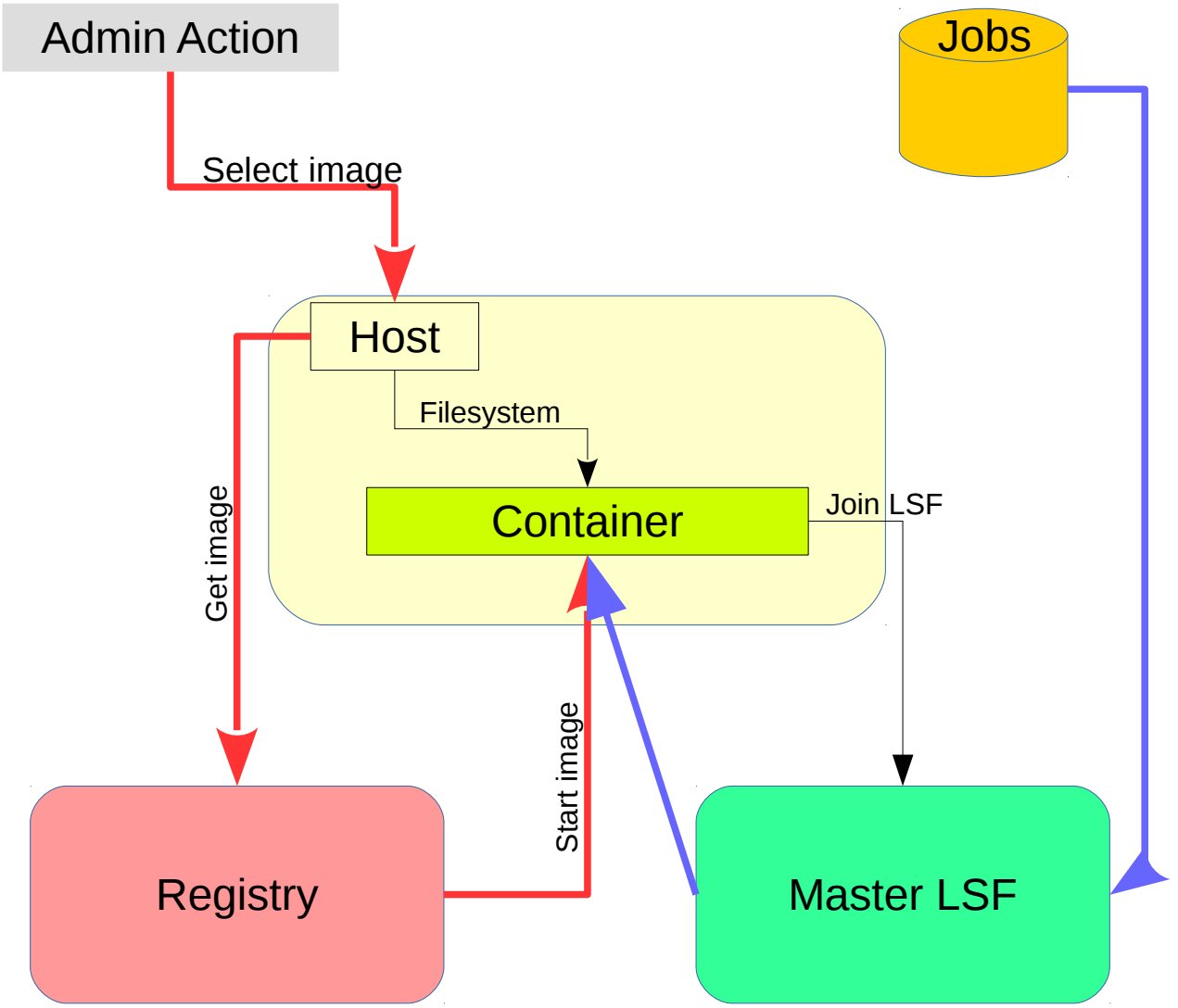
```
docker import http://swsrv.pi.infn.it/chroot/SL-6x-x86_64-EMI3WN.tar.gz wn-emi3
```

docker

We mimic a VM using a container



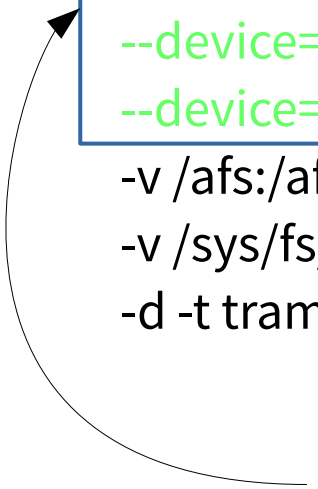
docker


Admin Action
Jobs
Select image
Host
Filesystem
Container
Join LSF
Get image
Start image
Registry
Master LSF

docker

We moved also HPC clusters into docker paradigm

```
docker run --name=wn-hpc1 --net=host \
    --ulimit memlock=-1 \
    --device=/dev/infiniband/rdma_cm \
    --device=/dev/infiniband/uverbs0 \
    --device=/dev/infiniband/ucm0 \
    -v /afs:/afs -v /gpfs/ddn:/gpfs/ddn -v /chrootlfs/home:/home/grid \
    -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
    -d -t tramontana.pi.infn.it:5000/hpc/wn-hpc1 /etc/sysconfig/docker-pi/start
```



To use IB inside the container

And the bare metal?

During these years we concentrate on User land

- It is the part that changes more frequently
- Using docker we got a very flexible and manageable solution

SysAdmin land untouched since 15 years ago

- Tools to manage bare metal are very old
- We use a mix of DHCP, PXE and in house scripts to install and manage the bare metal
- Same considerations as before about scalability and errors

It is possible repeat the story?

- Start to look for a standard tool to manage the bare metal
- Many available out there

Let's start testing

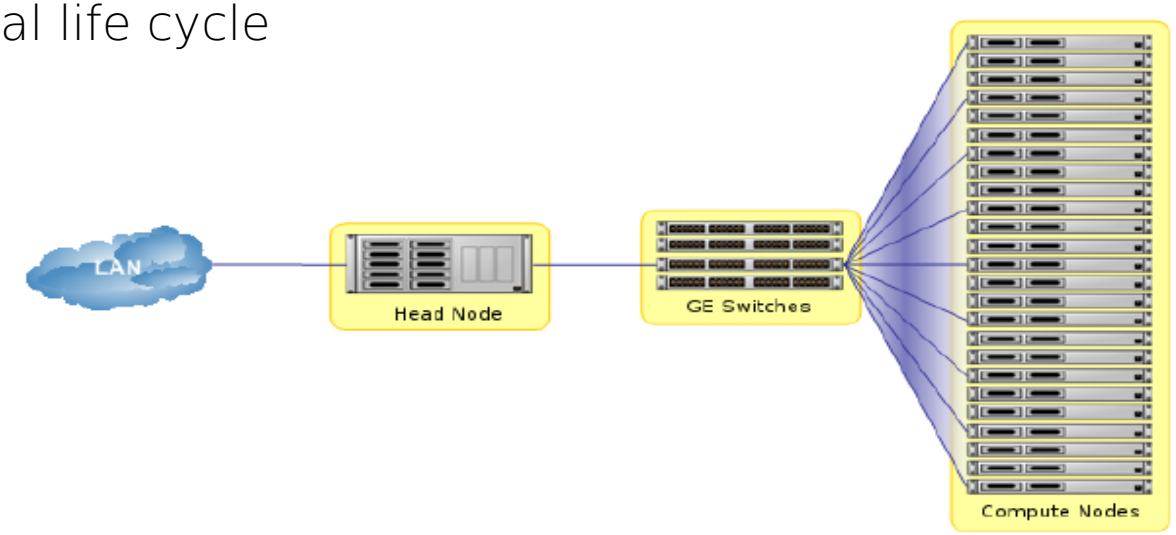
Bright Cluster Manager

Widely used in HPC

- Re-engineering of installation node process

Testing Bright features

- Installation of a test bed cluster
- Test of integration with production infrastructure (LAN, DNS, DHCP)
- Test of compatibility with our SysAdmin env (GPFS, AFS, CVMFS ecc…)
- Centralization of bare metal life cycle



Bright Cluster Manager

Very promising results

- Integration with production infrastructure OK, both using Bright CM services or the production ones
- SysAdmin env ~OK, AFS and CVMFS no problem
- Test still in progress for GPFS. It is a node symmetry breaker. We need to assure that Bright CM operation do not interfere with GPFS cluster operation

Conclusions

Docker is the perfect solution for User land

- Simplified management and life cycle
- No performance penalty
- Transparent migration to the new paradigm
- Solid foundation for future needs

Exported lesson learned to the SysAdmin land

- Good candidate
- Test are in progress

Simplified and strengthened the management of the site

Thank you!