- The WLCG
- Motivation and benefits
- Container engines
- Experiments' status and plans
- Security considerations
- Summary and outlook

Status of Plans to Use Containers in the Worldwide LHC Computing Grid

SWISS EXPERIENCE

Gianfranco Sciacca

AEC - Laboratory for High Energy Physics, University of Bern, Switzerland

hpc-ch Forum on Containers for HPC - 26 October 2017, Paul Scherrer Institut

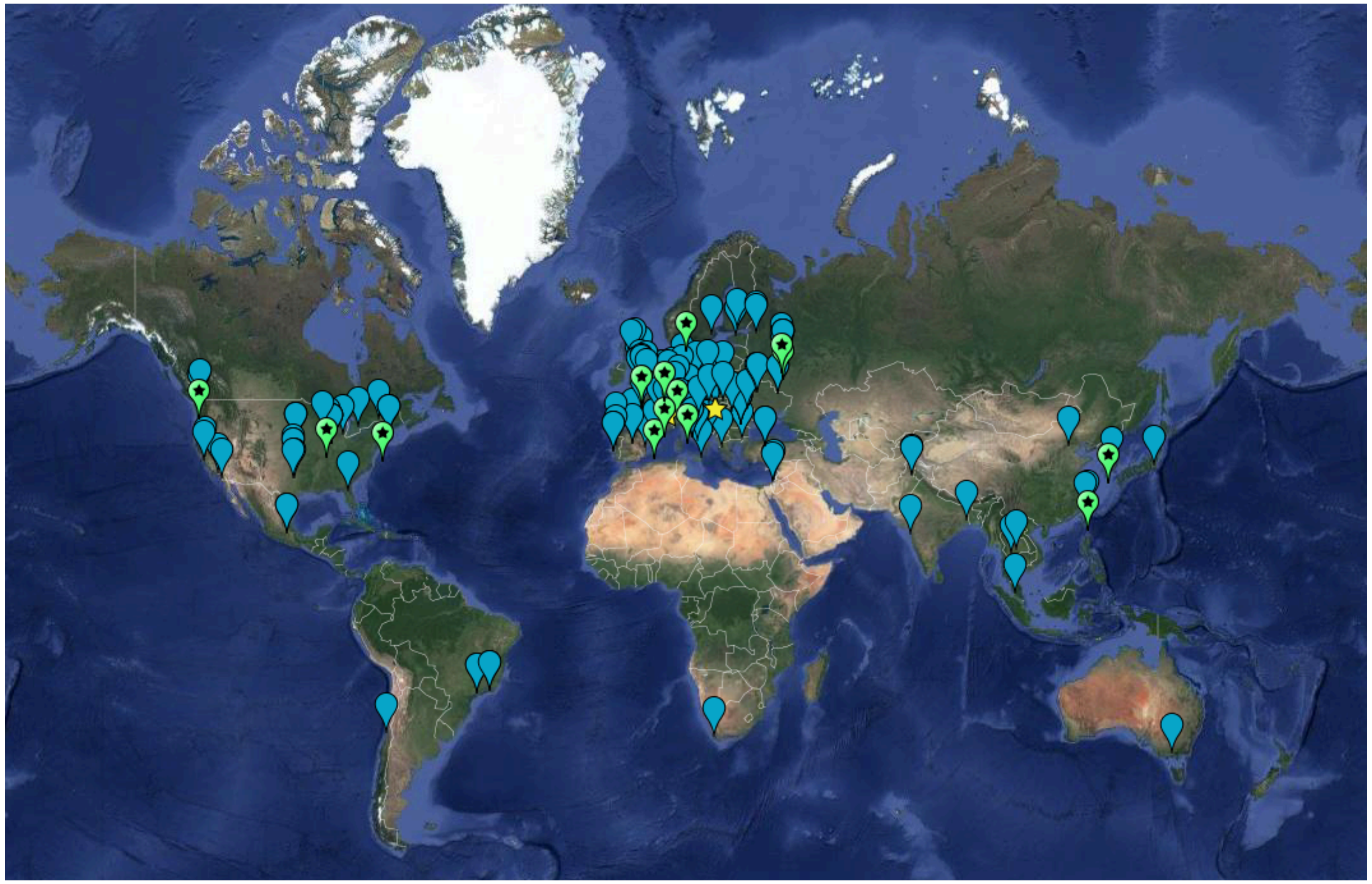
UNIVERSITÄT BERN
AEC ALBERT EINSTEIN CENTER FOR FUNDAMENTAL PHYSICS

LABORATORIUM FÜR HOCHENERGIEPHYSIK
LHEP
UNIVERSITÄT BERN

THE WLCG

http://wlcg-public.web.cern.ch

- **The Worldwide LHC Computing Grid (WLCG)** project is a global collaboration of more than 170 computing centres in 42 countries, linking up national and international grid infrastructures.
- The mission of the WLCG project is to provide global computing resources to store, distribute and analyse the ~50 Petabytes of data expected in 2017, generated by the Large Hadron Collider (LHC) at CERN on the Franco-Swiss border.
- WLCG is co-ordinated by **CERN**. It is managed and operated by a worldwide collaboration between the experiments (**ALICE, ATLAS, CMS and LHCb**) and the participating computer centres.



MOTIVATION AND BENEFITS

1/3

- **General Aim to:**

  reduce operational effort on the sites, and to meet the needs of the ongoing analysis reproducibility work going on in the experiments.

- **One of the current systems' limitation: all jobs use the same root filesystem as the host**
  - **– i.e. workloads tied to the host OS**
    - **an SL6 host can only run SL6 workloads**
    - **software/OS dictated by the LHC experiments**
    - **slow evolution because of stability needed for data taking**

MOTIVATION AND BENEFITS

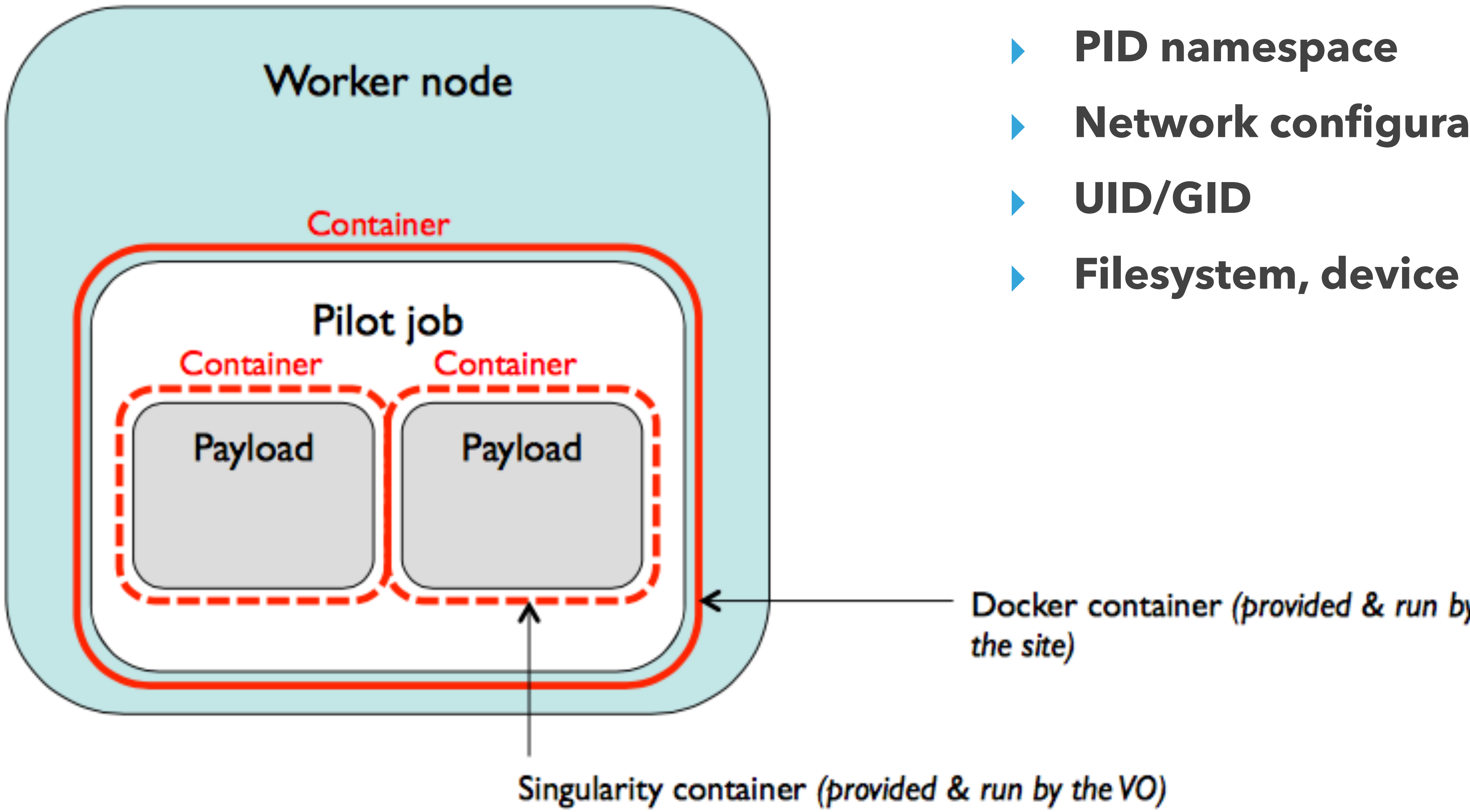
2/3

- **Containers are similar to VMs but more flexible and no performance loss**
  - **Native performance as compared to true virtualisation (also on HPC systems)**
  - **Effectively using a custom set of OS libs and software, apart from sharing the kernel with host OS (similar to chroot)**
- **Provide independence of the execution environment from the OS**
  - **Isolate experiments from site choices/upgrades**
  - **Isolate sites from experiment constraints**
- **Make it easy to create test environments**
  - **Several different environments can be used at the same time on the same site**
- **Common approach for execution, software distribution for all sites (including HPC)**

MOTIVATION AND BENEFITS

3/3

- **Isolation of payload**
  - **payload jobs cannot see other processes on the host or even processes from the pilot**
  - **payload jobs cannot see any files from the pilot**



- **PID namespace**
- **Network configuration**
- **UID/GID**
- **Filesystem, device access**

CONTAINER ENGINES

1/2

- **Main products : Docker, Singularity**

- **Dominated by Docker, better suited for full encapsulation**
  - **Analogous of VMs, full software and environment stack**
  - **Arbitrary workflow or service execution**
  - **Docker instances can be *long lived* - service deployment model**
  - **Or application oriented - execution of complex workflows**
  - **Provisioning model similar to VM**

- **Singularity: new engine from the HPC world, very lightweight, removing the unnecessary parts from Docker in our context**
  - **OS encapsulation but use as much as possible from host OS, lower initialisation latency**
  - **Designed for batch job execution, focusing on simplicity and minimal configuration**
  - **E.g.: singularity <OS Image> <command>**
  - **Can also run in user space (no SUID) with limited functionality (eg no bind mounting)**

CONTAINER ENGINES

2/2

- **Docker and singularity can use identical images, the usage of either is purely context dependent**
- **Singularity is better suited for simple job execution at sites, while docker requires more complex deployment and more privileges on a site**
- **Other engines not as popular in our community: Linux containers (lxc), Rocket (rkt), systemd**
- **Ability to build container clusters with orchestrators**
  - **Mesos (good for long-running service), Kubernetes (availability to build small clusters), …**

EXPERIMENTS' PLANS AND STATUS: ALICE

1/1

- **ALICE still familiarising with the technology**
- **First step is deploying the experiment specific services at sites (VO-box) in a container**
  - **Currently in pre-production**
- **Find that singularity is a simple and good isolation method**
- **Currently in the development plan**
  - **Expect to be integrated in the "Job Agent" (pilot) code by end of 2017**
- **Container setup: centralised and simple container for jobs would be the best approach**

EXPERIMENTS' PLANS AND STATUS: ATLAS

1/2

- **ATLAS plan is to use both**

- **Singularity:**
  - **Well suited to be used everywhere *making the site SW specifics irrelevant to ATLAS***
  - **Jobs can be executed on every site regardless of the site OS and do not require any customisation at the site.**
  - **Site upgrades decoupled from ATLAS SW requirements**
  - **ATLAS can use several OS versions at the same time matching the experiment software release requirements**
    - **E.g.: Run-1 analysis on SLC5, SLC6 images, Run-2 on SLC6, CC7, …**

- **Docker:**
  - **Currently the best way to encapsulate more complex tasks, such as software development/testing or analysis preservation**

EXPERIMENTS' PLANS AND STATUS: ATLAS

2/2

- **Current deployment focus:**

- **Singularity should be widely deployed on most of the computing resources both pledged an opportunistic**
  - **Job step containerised execution:**
  - **Stagein, RunJob, stageout pilot steps are each executed in a separate container instance**
  - **Proof of concept tested, will need more pilot code refactoring**

- **Docker is for now not considered yet, although some proactive sites are already supporting it**
  - **To be addressed in 2018**

- **Several sites (~10) are using singularity already for ATLAS production, although in a way which is not controlled by ATLAS**
  - **Eg: forcing automatic execution of all ATLAS jobs in SLC6 containers**

- **Some HPCs using shifter with Docker (WNs run as Docker containers)**

EXPERIMENTS' PLANS AND STATUS: CMS

1/1

- **CMS plan is to use singularity:**

- **Provides the isolation needed by CMS, does not do resource management (the batch system does)**
  - **No daemons, no UID switching (glexec)**
- **Easy to install: default configuration is OK, no need to edit config files**
- **User gains no privilege being inside the container**
  - **E.g. all setuid binaries disabled in the container**
- **Will allow to decouple the OS installed (and used by the pilot) from the one used to execute the payload**
- **The pilot is in charge of instantiating the appropriate container: can use a different container for each payload it schedules**

- **CMS decided to use Docker images rather than native ones**
  - **allows to easily import the images into their own distribution system**

EXPERIMENTS' PLANS AND STATUS: LHCB

1/2

- **LHCb plan to use singularity, but not as a hard requirement yet:**

- **LHCb work a lot with specially defined VMs and use their own VM provisioning engine**
- **Working on integrating the singularity functionality into their own WM pilot-based framework**
  - **Isolation: no more UID switching to run each payload (glexec) in the pilot**
  - **Useful, but not a hard requirement for provisioning SL6 on CentOS7 worker nodes. Docker and VM regarded as possible alternatives**

- **Now developing a generic LHCb container definition based on their VM experience**
  - **Uses Docker and the generic CERN root image**
  - **Overlays as needed LHCb specific setup scripts, sourcing minimal dependancies from the CernVM-FS and/or from their Web servers**
  - **Must be compatible with the current VM provisioning engine**
  - **Do not expect to need special LHCb images**

EXPERIMENTS' PLANS AND STATUS: LHCB

2/2

- **Longer term: containers as a user job format?**
- **There is a lot of interest from LHCb users in packaging their jobs in, say, Docker images**
- **Allows reuse of other people's code and management of what the user has changed**
- **Makes analysis more reproducible and easier to recreate in the future**
- **Asses effort vs benefit**

SECURITY CONSIDERATIONS

Strengths and Issues

- **Benefits**
- **Containers decouple provisioning and experiments**
  - OS/library independent from experiments
  - No experiment libraries leaking to provisioning
- **Containers provide a better isolation than UID switch (glexec)**
  - WN processes and files invisible/not accessible
  - cgroups to manage resources used

- **Potential issues**
- **Young technology: new classes of bugs in the kernel, missing support and the ecosystem changing fast**
- **Most kernel bugs can still be exploited with containers: still need the ability to do emergency updates**
- **Singularity is still SUID: could disappear in the future but a sysctl configuration might be needed**
  - Disabling suid will disable OverlayFS
- **Singularity is an attractive technology to replace the UID switch, but would rely on kernel security updates**
  - No central service required: simpler configuration means less failures but at the price of no traceability to the end-user. It needs the experiments to do the appropriate logging (some do it already)
  - Potential impact on the way central banning is done: move from site-based central banning to VO-based central banning

SUMMARY AND OUTLOOK

Use of containers in WLCG

- **The 4 LHC experiments all plan to leverage the container technology to some extent. Pretty much work in progress for all of them**
- **Singularity with Docker based images is the prevailing trend at the moment**
- **WLCG looking at co-ordinating the efforts to the possible extent. Green light given to wide deployment of singularity**
  - **Experiments should collect the experience, site specific requirements or configuration specifics in the next few months**
- **Main immediate benefits:**
  - **Decouple experiment needs from provisioning**
  - **Isolation**
- **Longer term**
  - **Reproducibility of analysis**
  - **Containers as user job format**
- **Points to evaluate:**
  - **Can the experiments use common images?**
  - **Are unprivileged containers enough (can run completely in user space)?**
  - **Are there custom configuration requirements on sites?**
- **Some sites already using containers on their own initiative**
- **Some security issues to keep under the radar**