A virtualized cluster testing framework

DSS Data Storage Services

Justin Lewis Salmon, Lukasz Janyst

CERN IT Department

1. The Challenge…

- How can we do functional tests with distributed software?
  - We need a distributed setup (cluster) to test real-world scenarios
- How do we orchestrate the behavior of each machine?
  - We need to be able to explicitly control the actions of each participating machine
- How do we automate the process?
  - We want some kind of continuous integration support
- We do not necessarily have the resources for a large physical setup
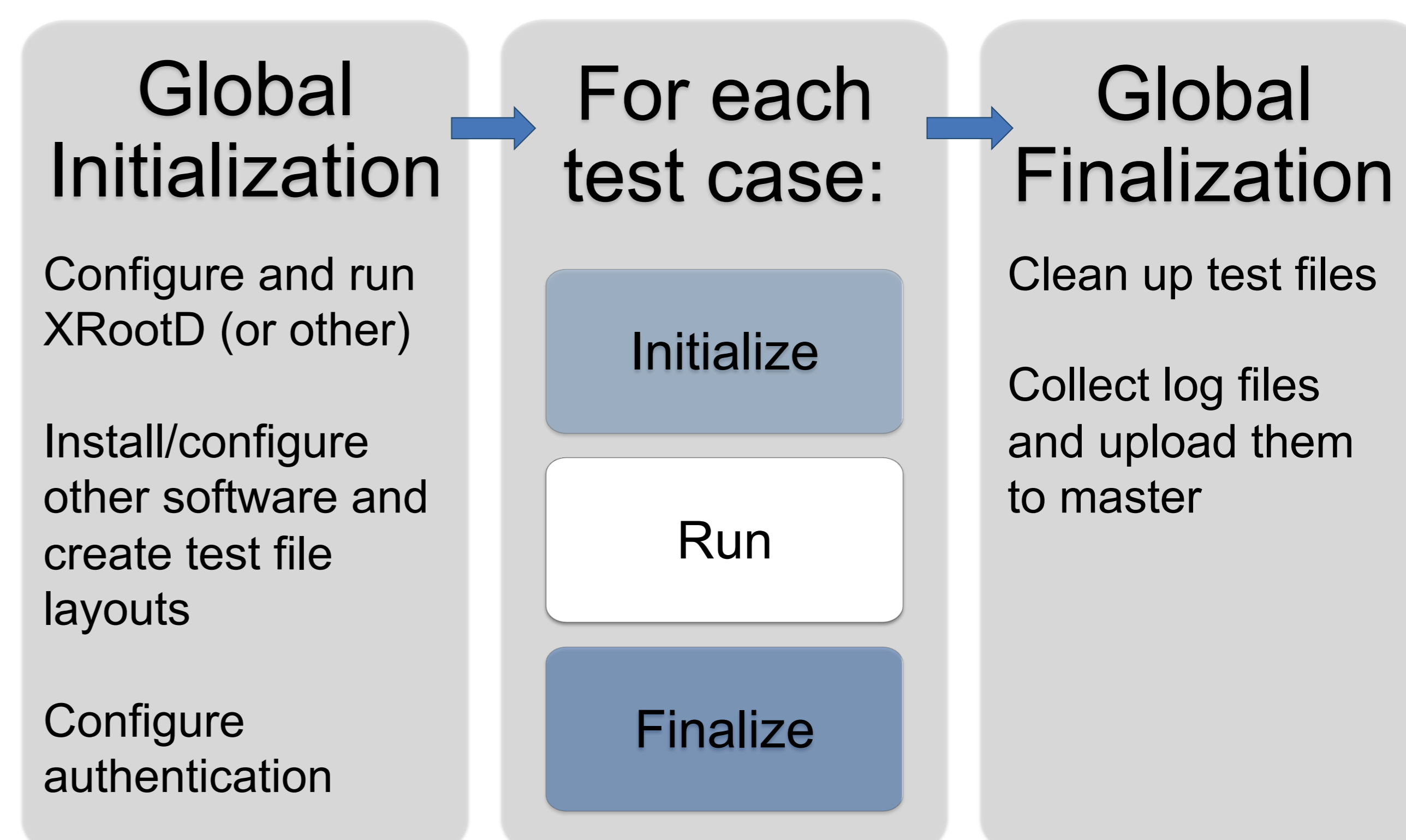
2. A Solution

- We can use virtualization to help us
  - libvirt virtualization API
- We use a 3-layer hierarchy, comprised of one master, one or more hypervisors, and multiple slaves
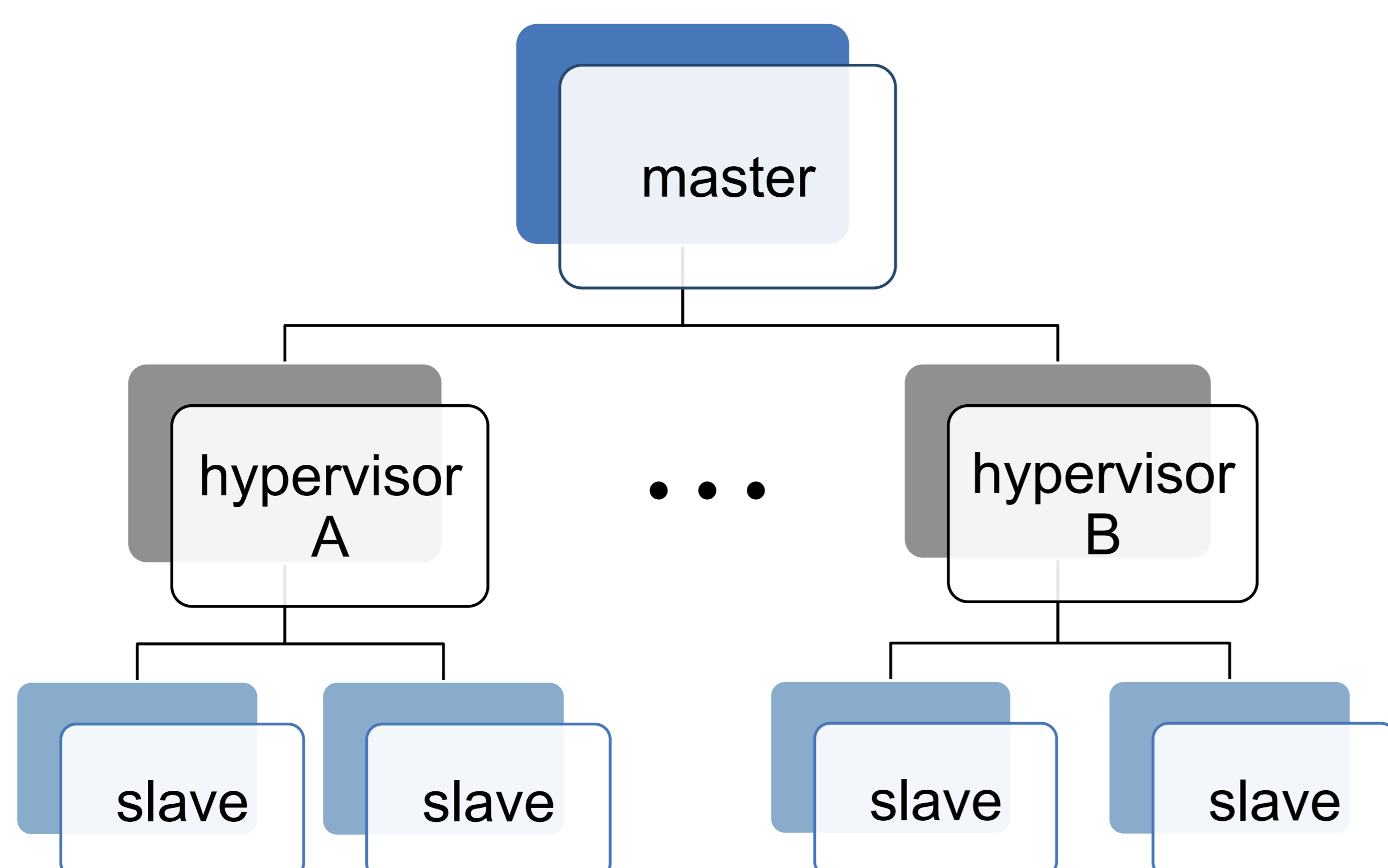
- Slaves are virtual machines (kvm/qemu)
- Hypervisors are real machines, which start/stop/configure the slaves, using libvirt
- The master is a real machine, which sends commands to the hypervisors (e.g. "start a particular cluster", "run test suite X")
- Slaves are connected to form a cluster using a virtualized network
- Flexible machine and network configuration possible through simple definition files
  - Disks (number, size, mount point)
  - CPU architecture
  - Memory size
  - Network topology
  - Load-balancing IP aliases

3. Multi-stage test suites

- Test suite has 3 main stages: initialization, run, and finalization
- The "run" stage executes test cases, which are also divided into initialization, run and finalization

Global Initialization — Configure and run XRootD (or other); Install/configure other software and create test file layouts; Configure authentication

For each test case: — Initialize / Run / Finalize

Global Finalization — Clean up test files; Collect log files and upload them to master

- Each stage is synchronized across all slaves, so no stage is run before the previous has finished on all slaves
- Everything happens via templated shell scripts which are delivered to each slave
- Detailed output from each stage is recorded and displayed together with historical runs via web interface provided by the master
- Test suites are version controlled using git repositories
- The framework automatically acts on changes in the repsitories

4. Status and outlook

- We currently have a working prototype running at CERN
- In the short term we plan to use it to test all the XRootD builds coming from the continuous integrations system and, possibly, other projects within the group

CERN IT Department
CH-1211 Genève 23
Switzerland

http://cern.ch/it-dss

A virtualized cluster testing framework
(CHEP 2013, 14th – 18th October, Amsterdam, The Netherlands)

XRootD   UWE BRISTOL   CERN