

# Feedback from CMS

Andrew Lahiff

STFC Rutherford Appleton Laboratory

Contributions from Christoph Wissing, Bockjoo Kim, Alessandro Degano

CernVM Users Workshop

5<sup>th</sup> March, 2015



## > CVMFS

- > Software deployment in CMS
- > Content
- > Installs
- > CVMFS in user space
- > CMSSW integration builds



- > Basically fully based on CVMFS
  - Recommended strategy by WLCG TEG (Technical Evolution Group) in 2011/12
    - > Established via WLCG CVMFS task force among all LHC experiments
  - All sites used by CMS provide CVMFS
    - > Includes the Tier-0 infrastructure
- > Additional deployment in AFS space at CERN
  - Backup
  - Not recommended to be used



- > Main CMS software package
- > Includes “all” from generators over simulation to reconstruction
- > Packaged as RPMs
  - Externals shared among several CMSSW releases
    - > ROOT, GEANT, GCC and various libraries
    - > Also packaged as RPMs by CMS
- > Typical numbers for a CMSSW release
  - 150k+ files
  - 10k+ directories
  - ~6GB disk space required
- > Releases typically built for different architectures or GCC versions
  - Per release on “production architecture”
  - Others are “development” version
  - All get installed in CVMFS



- > Additional input required for some workflows
- > Gridpacks
  - Generator files
- > Particle Density Functions
  - PDF sets from LHAPDF
- > Both include a few large files (100MB+)
  - Demanding for Squid infrastructure



- > Tool for submission of Analysis jobs
- > CRAB2
  - Frozen, no further updates planned
- > CRAB3
  - Recent product
- > Some additional Grid client tools are needed
  - Tools to manage VOMS proxies
  - Can be used from `/cvmfs/grid.cern.ch`



- > Executed via cron a few times per day
- > CMSSW and CRAB client installs are fully automatic
  - Check for new CMSSW releases in “CMS tag collector” and/or “release.map”
  - Check for new CRAB versions in certain repository directories
  - Install new versions
- > Gridpacks are installed via rsync from an agreed EOS directory
  - New Gridpacks are pushed to CVMFS automatically
- > LHAPDFs are downloaded from the web and installed in CVMFS
  - Some steering files to be manually edited before installation
- > For directories known to become large nested catalogs are created
- > Error notifications sent via e-mail
- > Feedback
  - Deployment machine sometimes very slow
  - Just bad interference of the VM with other VMs on same Hypervisor?



## > Use case

- Opportunistic resource without root access
- CVMFS cannot/will not be installed by administrators
- Cannot deploy Virtual Machines

## > Approach

- Parrot (from <http://ccl.cse.nd.edu/software/manuals/parrot.html>)

Parrot is a tool for attaching old programs to new storage systems. Parrot makes a remote storage system appear as a file system to a legacy application. Parrot does not require any special privileges, any recompiling, or any change whatsoever to existing programs. It can be used by normal users doing normal tasks. For example, an anonymous FTP service is made available to `vi` like so:

```
% parrot_run vi /anonftp/ftp.gnu.org/pub/README
```

- Allows to provide CVMFS in user space

## > Experiences

- > Has worked for some preliminary tests (e.g. CMS jobs at ATLAS sites)
- > Working on making it production ready





- > Periodic automatic builds of CMSSW are called Integration Builds
  - > Several combinations of platform / architecture / gcc version
  - > 7 base versions of CMSSW plus several flavor variations
  - > 19 releases produced twice a day
- > The latest two weeks of IBs are always available for developers
- > The CVMFS service installing & providing the IBs is based on a single Virtual Machine on CERN's cloud infrastructure and is configured through Puppet



- > CVMFS is resource hungry when deployed on the cloud infrastructure
  - > Need for 500 IOP/s ceph volumes to publish ~7M files (2 weeks of IBs)
  - > Requires VM with at least 8 GB RAM
  - > Huge number of file descriptors open
- > CVMFS puppet module
  - > Despite being very efficient, recreating a machine from scratch is not trivial or fast
- > Because of the above and the data we distribute, to recreate a CMSSW IB repository takes ~ 50 hours to publish these IBs
  - > However we produce more IBs during these 50 hours, so another iteration of installation
  - > Luckily after a few iterations the IBs installation catches up with the building



- > CVMFS developers added several features we appreciate
  - > We always keep the 2 most recent weeks of IBs
    - > Garbage collection eventually becomes a must
  - > FNAL replicates our repository as a stratum 1
    - > They can garbage collect even if we don't
  - > Automation is more reliable with CVMFS recovering from strange repo status and fixing itself