CVMFS in ATLAS

Alessandro De Salvo

Oana Boeriu

Asoka De Silva

Alessandra Forti

Lorenzo Rinaldi

on behalf of the ATLAS Collaboration

Jan 30th 2018 – CernVM Users workshop

CVMFS usage in ATLAS

- **Different usage areas (and repositories)**
  - **Production area**
    - **All production releases, users' software interface, setup tool for production, diagnostic tools, etc**
    - **Stratum0 + WLCG stratum-1s**
  - **Nightlies**
    - **Nightly releases**
    - **Fast cycles, only stratum-0 (at the moment)**
    - **Used not just by users but also for automated testing on the Grid**
  - **Conditions data**
    - **All condition pool files, used together with Frontier**
  - **Data preservation? (in the future)**

ATLAS CVMFS production area

- **~3100 “releases” currently in CVMFS**
  - **Base releases**
  - **TDAQ releases**
  - **Patch releases (AtlasProduction, AtlasDerivation, …)**
  - **Analysis releases (several projects)**

- **Almost no obsolete release since the migration to CVMFS**
  - **All releases >= 15.0.0, with a few exceptions, are still considered production mode**
  - **Very high load on site validations, although they are not currently tested on sites since the releases 21.X (>= 2017)**
  - **Probably less than a half of the releases are really used in production**

ATLAS Conditions data distribution on CVMFS

- **Poolcond Files and PF Catalogues**
  - **Used by almost all MC/data grid jobs**
    - **Script running on cvmfs-atlas-cond every 4 hours (few minutes)**
      - **Script (rucio-)downloads new PF and creates new PFC**
- **DBReleases (*Full and Custom*)**
  - **Collection of sqlites and Poolcond files**
    - **Used for MC production, mainly on HPC sites**
    - **Loaded on CVMFS via pacman/rpm (~monthly)**

At present time, system is very stable (not very large volumes of data are managed, no heavy load on CVMFS observed)

ATLAS Nightlies Installations on CVMFS

- **The distribution code resides in: https://gitlab.cern.ch/atlas-sit/code-distribution/tree/master/atlas-rpm-install**
  - **python scripts used for the CVMFS installation have ca 2780 lines of code (large part belonging to monitoring).**
    - **the main install.py script has around 400 lines.**
    - **performing ca 19 installation daily for opt, dbg and opt-dbg platforms;**
    - **daily clean-up of older than 30 days installations.**
    - **branches installed automatically based on the RPMs produced by our nightly builds.**
- **Since we install multiple date-time stamp nightlies under the same branch directory, a clean-up of the .yumcache and .rpmdb directories necessary in order to avoid warnings (large files) from the cvmfs system.**
- **Most installation problems encountered are due to dependencies not being found on the official web sites (e.g. too new LCG, tdaq- / dqm-common versions) and more rarely now connection issues to web sites where repodata/RPM info is read from.**
- **We observe sometimes longer synch times with the outside world, especially when installing dependencies for the first time under a new branch - this can delay subsequent installations and start of following tests (ART/RTT).**

ATLAS Containers in CVMFS

- **Only Grid containers are stored in CVMFS, FAT containers for HPC will be distributed in a different way**
  - Grid containers are “light”, i.e. they do only need the OS and the runtime libraries + some of the middleware libs
  - Automated build via Github + Docker Hub
    - https://github.com/atlasadc/atlas-grid-docker
    - https://hub.docker.com/u/atlasadc/
  - Docker containers are imported into Singularity and distributed to the users on demand via CVMFS
    - Full images are stored in CVMFS directly
    - Unpacked images are stripped to save space (rm /var/lib/yum,/var/cache/yum,/usr/lib/locale,/usr/share/locale,/usr/share/doc,/boot)
  - Currently distributed in CVMFS (ATLAS tree) in two flavors
    - Monolithic Singularity images: /cvmfs/atlas.cern.ch/repo/containers/images/singularity
    - Unpacked Singularity images: /cvmfs/atlas.cern.ch/repo/containers/fs/singularity
- **Image sizes**
  - Full images are ranging from 1GB to 1.5 GB
    - Hard to be handled by CVMFS, max 1GB
    - These images must _NOT_ be used in production, only for testing
    - Size could improve for full images if we make use of squashfs, but is it really needed?
  - Unpacked images are ranging 750MB to 1.5GB
    - Easy to be handled by CVMFS, as you can use the single unpacked files
    - Some problems in running with these images (to be solved by the Singularity team)

ATLAS Users' software interface

- **ATLAS provides a menu driven UI for users to setup software from cvmfs.**
  - **It hides details of paths and dependencies from users.**
  - **It provides diagnostics tools to help with user support.**
  - **This is a consistent look and feel on lxplus, CernVM, laptop, etc.**
  - **This is also used on the grid to set up various tools.**
  - **It all works very well as we test on platforms prior to deployment.**

- **Usage examples**
  - **eg on lxplus, users login and type setupATLAS**
  - **Working on lxplus or elsewhere, do**

```
export ATLAS_LOCAL_ROOT_BASE=/cvmfs/atlas.cern.ch/repo/ATLASLocalRootBase
alias setupATLAS='source ${ATLAS_LOCAL_ROOT_BASE}/user/atlasLocalSetup.sh'
setupATLAS
```

  - **We also provide a mechanism to setup a singularity container; eg `setupATLAS -c slc6` will setup a slc6 container from cvmfs with the menu for various tools.**

Conclusions

- **CVMFS very well performing so far**
  - **In different environments**
  - **Good and fast support from the developer team**
    - **Crucial to have an open channel with the developers, as we might have urgent requests popping up very quickly**
  - **Quite happy for now with the propagation times**
    - **Reduced to only 15 minutes, works well with all our workloads, especially for quick turnaround data like the Atlas Grid Information System (AGIS) cache**
    - **…But in the future we could also benefit from a kind of ”publish-driven” propagations to stratum-1s, which is anyway foreseen already**
  - **Still problems with the increased usage of the inodes due to publishing and 32 bit applications**
    - **Not easy to fix at the CVMFS level, but it will possibly disappear once no 32 bit (old) releases will be used anymore**
  - **Good progress for containers, to be followed up in the next months, especially for the Docker interface**