

LHCb Nightlies Use Case for CVMFS

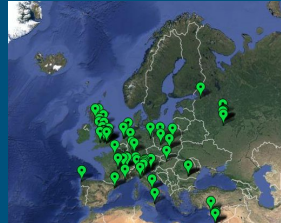
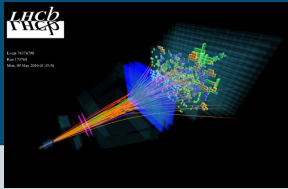
CernVM User Workshop

03/06/2019

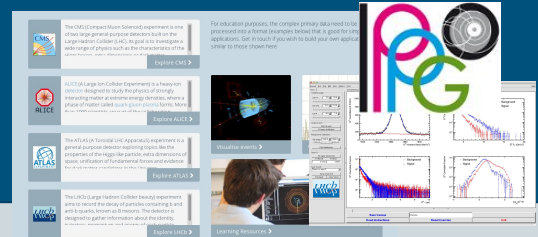
Ben Couturier, Stefan Chitic, Marco Clemencic,
Joel Closier (for the LHCb Computing team)



Where do we use CernVM and CernVM-FS ?



79 Institutes,
18 Countries



Trigger

Production

User Analysis &
Software development

Data Preservation
Masterclasses



LHCb CernVM-FS repositories

	Size	GC	Role	Comments
<code>/cvmfs/lhcb.cern.ch</code>	1.5 TB	No	<ul style="list-style-type: none">• Production software (Online/Offline)• <i>Until 2017: conditions</i>	<ul style="list-style-type: none">• Used by Grid and LHCb Trigger farm• A few transactions/day in average• Crucial for long term software preservation
<code>/cvmfs/lhcbdev.cern.ch</code>	2.6 TB	Yes	<ul style="list-style-type: none">• Continuous integration artifacts and installation• Software under test• Clone of GIT repositories	<ul style="list-style-type: none">• Deduplication is really helpful• High turnover: ~3.8 million files per day (~200 GB/day)• Short term interest• Crucial for development team
<code>/cvmfs/lhcb-condb.cern.ch</code>	~ GB	Yes	<ul style="list-style-type: none">• LHCb Conditions	<ul style="list-style-type: none">• Frequent releases• New volume to decouple the release of the GIT conditions from the long term preservation repository.

* As of 24/01/2018

LHCb Nightly Build Use Case

- AFS phase out prompted us to find a replacement to release the nightly builds output
- CVMFS is the (only?) obvious replacement candidate
 - Local installation is possible, but not necessarily desired on shared clusters
- Also used to released all sort of test/development read-only software and data
- Increased interest as LHCb is preparing for Run3:
 - LHCb will have a software only trigger, with so-called [Turbo lines](#), hence renewed effort on integration, regression and throughput test.
 - building and releasing the software with compilation flags optimized for the LHCb hardware (e.g. AVX2+FMA), as well as the standard version

LHCb Nightly “slots”

A “slot” is a consistent set of LHCb projects (each for a specific branch). We build each slot with different compilers and different compiler options.

c.f. <https://lhcb-nightlies.cern.ch/nightly/summary/>

They are critical to the LHCb software QA process. We build/test ~30 different slots, less than half are deployed to CVMFS. The characteristics are:

- Large amount of files (~ 3 to 4 millions) installed per day, amounting to ~200GB
- Short term interest
- Need to deploy quickly to keep up with the continuous integration system

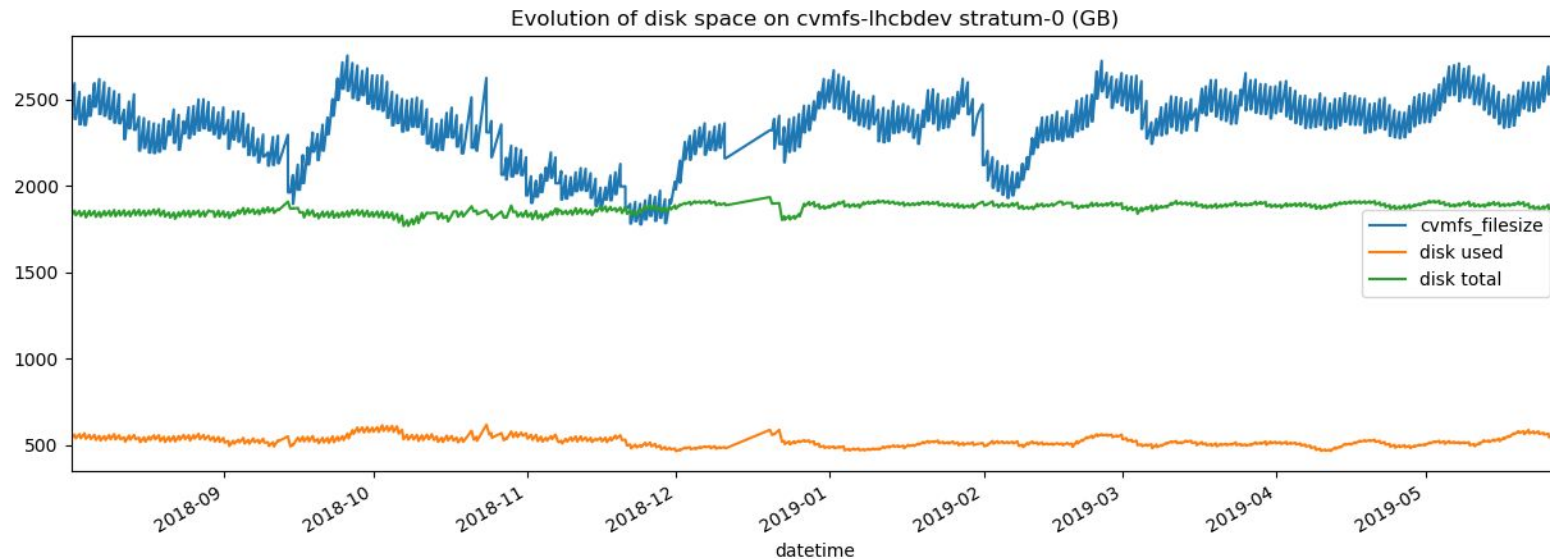
`/cvmfs/lhcb-dev.cern.ch` content

LHCb nightly slots installations, but not only:

- Clone of various GIT repositories
- Installation of LCG packages in test (on demand)
- LHCbDirac test versions (on demand)
- Conda environments (experimental, on demand)

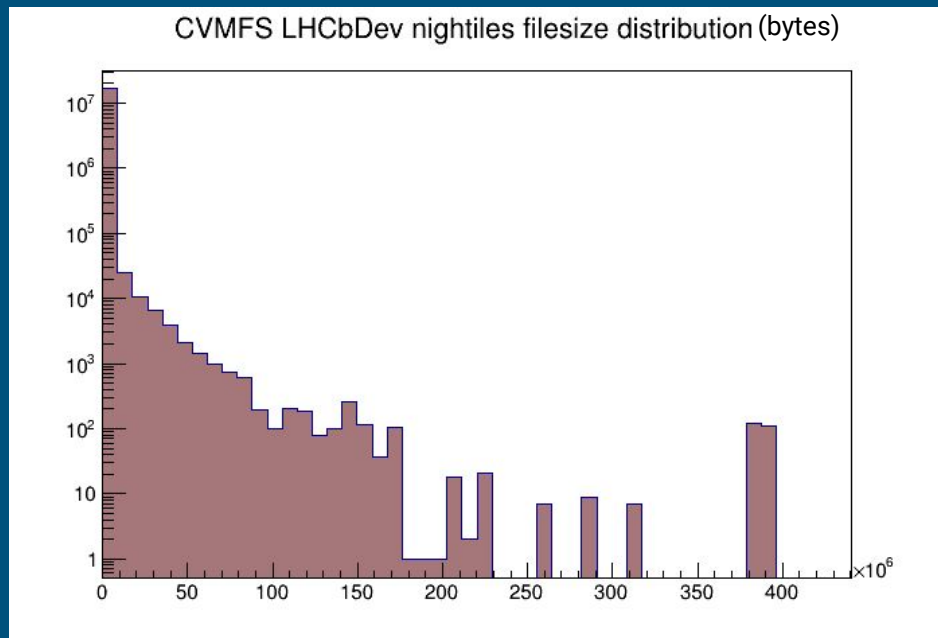
Removing old nightly builds or versions of the software to limit disk usage

Disk usage



Deduplication is crucial for the continuous build releases

File sizes



Content consists mostly of compilation artefacts and sources of various branches of the LHCb codebase with different compilation options (a.k.a nightly slots)

- Average file size of 400 Kib
- Dominated by tiny files
- High level of duplication between nightly slots makes CVMFS very useful

Status

The system has been in place for a few years now, with **excellent stability** !

- We routinely run 100s of transaction per day/100s of GB without errors

But with some **limiting factors**

- Propagation time to clients
- Constraints of release to stratum-0 (single user, serializing transactions)
- Installation time on stratum-0

Propagation time to CVMFS clients

- Used to be slow a few years back
- Improved by the CVMFS team and CERN IT

Now seem to be of the order of 10s of minutes

- *When will the new notification service be put in place ?*

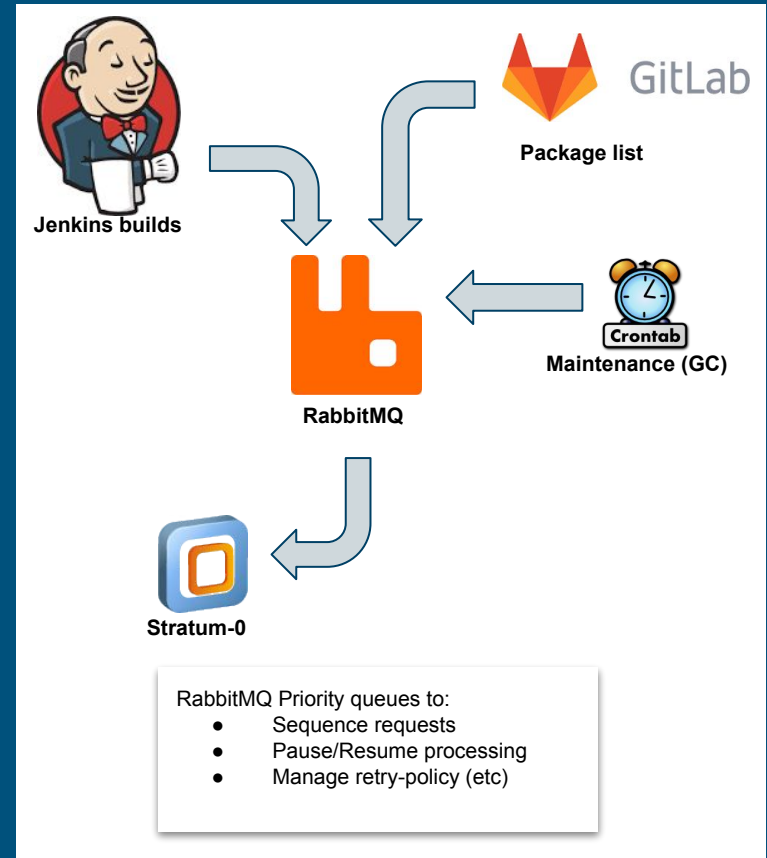
Release to stratum-0

Issue: Need to install packages from several sources, with different requestors

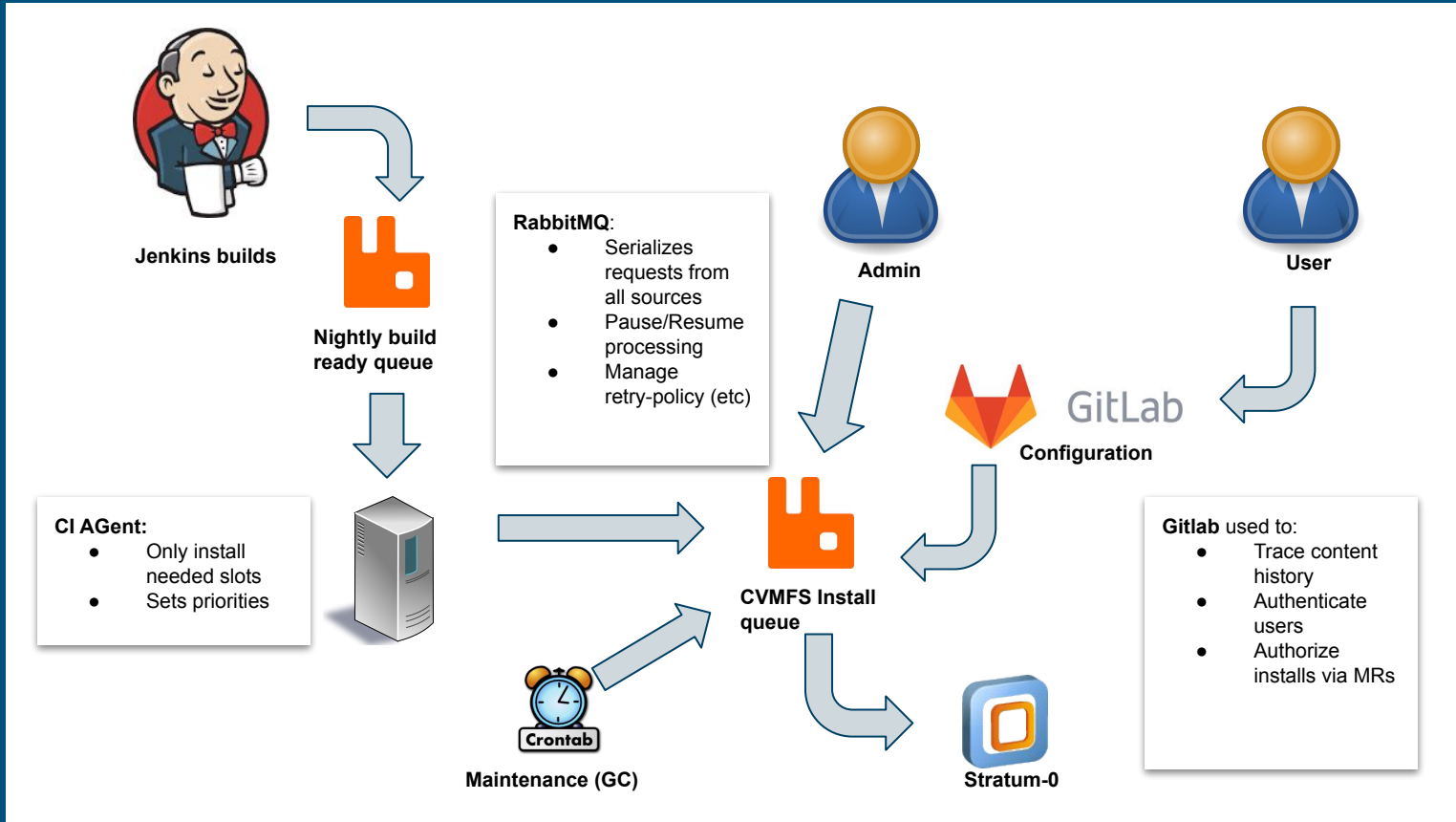
- Continuous build output installed when ready
- LCG Packages (Generators) installed on request
- GIT clone of master branches of the projects
- ...

Software installation take $O(\text{hours})$, how to organize/schedule/prioritize all those installations?

- Using a prioritized queue in RabbitMQ
- Multi-user aspect managed via metadata in Gitlab (for authentication, authorization and traceability)



LHCb deployment system



Installation time to stratum-0

- Has been a problem for a long time, we are currently not managing to install all we want every day (we manage to install 50% to 75% of what was requested for installation)
- With the current setup:
 - Installation speed seems I/O limited
 - Garbage collection is also time consuming (and has to be run regularly to avoid taking too long).

This led to the development of the features in the current deployment system:

- *fine grained package deployment system*
- *priority queue to deploy most important slots and platforms first*

Possible improvements

- Using server with SSDs
 - Tests by Jakob show an order of magnitude improvement
 - Possible as we can stay within a few TB
 - Need two hosts for continuity of service (but easy to put in place with stratum-0 to stratum-1 replication)
- Using multiple release managers + one gateway
 - Maybe not useful if the Gateway is the bottleneck?
- CVMFS Ingest available since 2.6.0
 - Haven't had time to investigate benefits
- Using S3 backend
 - Currently investigating S3 Service provided by CERN IT
 - Interesting solution but still seems to require several hours/day of installation
 - Allows shared storage for stratum-0 and stratum-1

Looking forward to see how to apply new developments presented in this workshop

Conclusions

*LHCb is very happy with the CernVM, CernVM-FS for production file deployments
Many thanks to the development team and people running the infrastructure*

CVMFS has been very successful for development artefacts distribution

- Request to deploy more and more files “on demand”
- Nightlies release use case very demanding in terms of I/O
- We still need to improve the time to release the nightly slots

Common Tools to provide “CVMFS deployment as a service” are very welcome

- Have to see how this could be done, needs a customizable framework as we install different types of payloads (tar files, RPMs...)
- Removal and garbage collection are crucial to stay within disk resources
- Installation/removal and garbage collection time have to be optimized

Happy to see from this workshop that there is very active development on CernVM/CernVM-FS