CMS deployments on CernVM-FS

Andrea Valenzuela Ramírez
`andrea.valenzuela.ramirez@cern.ch`

CERN - CMS Core Software

CernVM Workshop 2022
Nikhef, Amsterdam

September 13, 2022

Outline

1. Introduction
2. CMS Main Repositories
3. Distribution of CMSSW Container Images
4. Future plans & Improvements
5. Conclusion

CernVM-FS Use Cases at CMS

The CMS collaboration deploys to CernVM-FS under different use cases:

- Distribution of experiment **production software**.
- Distribution of **Integration Builds** (IBs).
- **Continuous Integration** (CI) purposes.

Repository Name	Size	Garbage Collection	Storage	Revision	Year
`/cvmfs/cms.cern.ch`	17.1 TB	No	S3	112584	2009
`/cvmfs/cms-ib.cern.ch`	2.34 TB	Yes (weekly)	S3	239433	2016
`/cvmfs/cms-ci.cern.ch`	538 GB	Yes (weekly)	S3	30668	2020

Table: CMS main repositories and their characteristics in terms of size, garbage collection frequency, type of storage, number of commits and year of creation.

Last reporting

CernVM User Workshop 2018 ▸ Presentation

CMS Offline Software (CMSSW)

- CMSSW contains the **software collection** needed to process event data.
- It has a **large code base** hosted on Github.

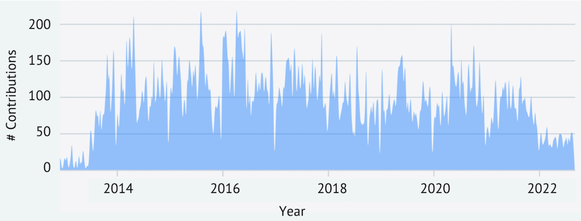


Figure: Contributions to master since 2013, excluding merge commits and bot accounts

CMSSW Code Base

- 100+ Contributors/month
- 500+ Commits/month
- 1250+ Packages
  - 3300+ Binary products

- CMSSW and its dependencies are built and distributed in form of **relocatable RPMs**.

CMS Main Repositories

- Distribution of experiment **production software**: `cms.cern.ch`
- Distribution of **Integration Builds** (IBs): `cms-ib.cern.ch`
- **Continuous Integration** (CI) purposes: : `cms-ci.cern.ch`

Software Repository `cms.cern.ch`

- All **CMSSW releases** are available via CernVM-FS in `cms.cern.ch`.

- 3K+ Full/Patch releases.
- 17 Active release cycles.

- **Releases are heavily used** by CMS Grid production and user analysis jobs, outreach, and CMSSW developers, etc.
- Automatically deployed via Jenkins on demand using **one single release manager**.

Content of `cms.cern.ch`

- CMS Offline Software.
- CMS Computing Tools/Software (e.g., CRAB, PhEDEx agents, Spacemon-client, etc)
- CMS configuration and datafiles.

Repository Statistics

The repository has a **low publishing frequency**.

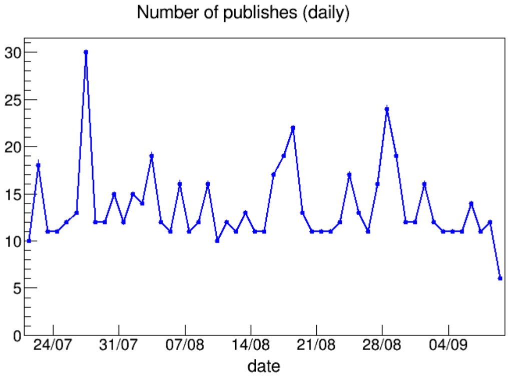


Figure: Number of `publish` operations per day from end of July to September 2022.

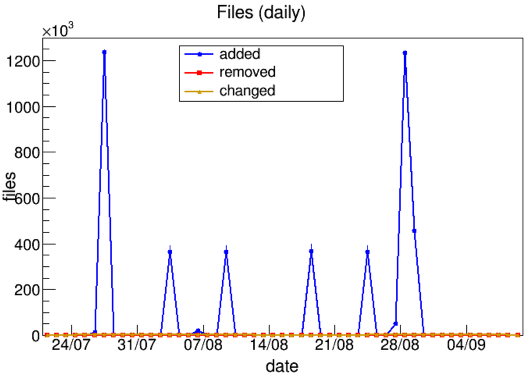


Figure: File addition/removal/changes per day from end of July to September 2022.

Deployment Workflow

A new release set is triggered by GitHub webhooks:

1. The releases are built in the form of a relocatable rpms in parallel.
2. They get uploaded to the *cmsrep* server.
3. They are deployed to `/cvmfs` using `cmspkg` tool.

- There is a priority queue to first deploy production architectures.
- Deployment time between 2-20 min per release (and OS/arch/compiler).

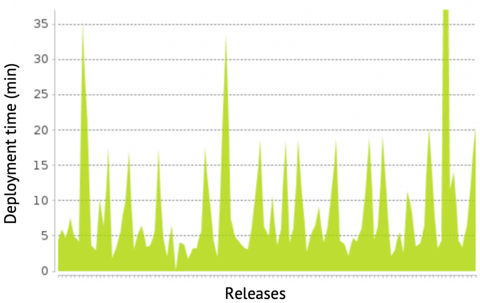


Figure: CernVM-FS deployment time for releases.

CMS Main Repositories

- Distribution of experiment **production software**: `cms.cern.ch`
- Distribution of **Integration Builds** (IBs): `cms-ib.cern.ch`
- **Continuous Integration** (CI) purposes: : `cms-ci.cern.ch`

IBs Repository `cms-ib.cern.ch`

- All **CMSSW IBs** are available via CernVM-FS in `cms-ib.cern.ch`.
- Automatically deployed via Jenkins CI **every 12 hours** using **one single release manager**.
  - Build only if there are changes with respect to the previous build.
  - Build for a set of active release cycles.
  - Build for multiple OS/archs/compilers.
  - Build different IB "flavors".
    - e.g., LTO builds for having CMSSW optimized with Linked Time Optimization.
- Once a week, fully build all release cycles.
- Around **30 IBs** are build and deployed **every day**. ▸ IB page

Content of `cms-ib.cern.ch`

- Two weeks of IBs.
- Github mirrors under `/cvmfs/cms-ib.cern.ch/git` (∼ 200 repositories).

Repository Statistics

This repository has a **higher publishing frequency** compared to `cms.cern.ch`.

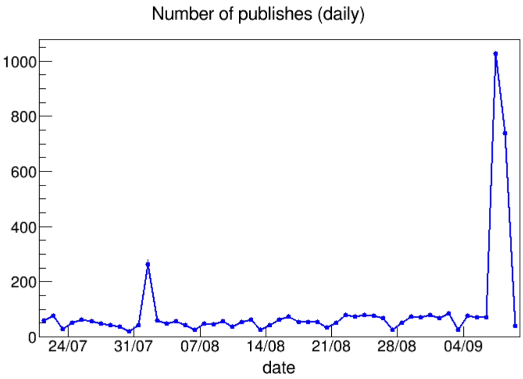


Figure: Number of `publish` operations per day from end of July to September 2022.

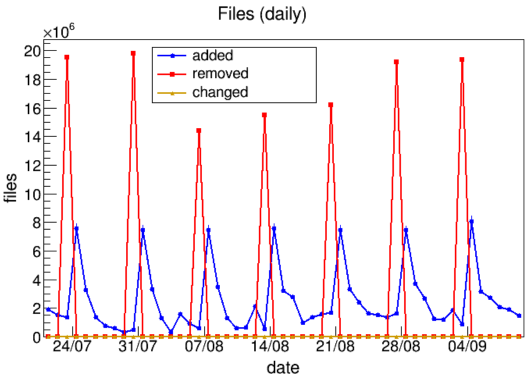


Figure: File addition/removal/changes per day from end of July to September 2022.

Deployment Workflow

Builds for a new set of IBs are scheduled in Jenkins:

1. IBs are built in parallel.
2. Deployment to `/cvmfs` is triggered as soon as the building process finishes.

- There is a priority queue to first deploy production architectures.
  - Deployment order is also dependent on the build time.
- Deployment time between 2-15 min/arch.
- **Waiting times for starting the deployment are larger.**

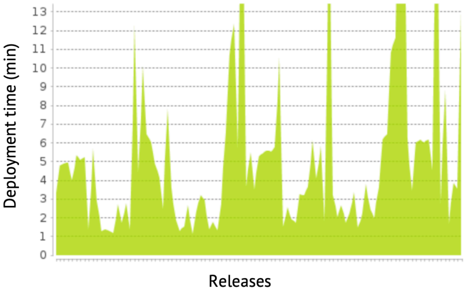


Figure: CernVM-FS deployment time for IBs.

Performance Bottleneck

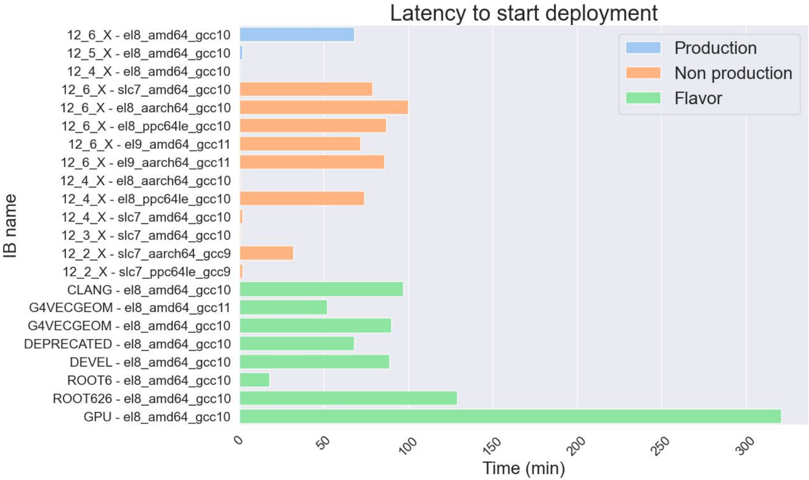


Figure: Waiting time of deployment jobs to `cms-ib.cern.ch` on 28/09 11pm IBs.

CMS Main Repositories

- Distribution of experiment **production software**: `cms.cern.ch`
- Distribution of **Integration Builds** (IBs): `cms-ib.cern.ch`
- **Continuous Integration** (CI) purposes: : `cms-ci.cern.ch`

CI repository `cms-ci.cern.ch`

- **CI build artifacts** are deployed to `cms-ci.cern.ch` for each CMSSW GitHub Pull Request testing request.
- Although deployment is done by using **one single release manager**, having the new build deployed on `/cvmfs` has benefits:
  - It allows for parallel testing.
  - Developers can use this deployment to provide fixes in less time.
  - It saves a lot of build time in GPU nodes.
- Around **60 PRs/week**.

Content of `cms-ci.cern.ch`

- Two weeks of CI build artifacts.
- CMSSW baselines for comparison during testing.

Repository Statistics

CI artifacts are deployed daily to `cms-ci.cern.ch`.

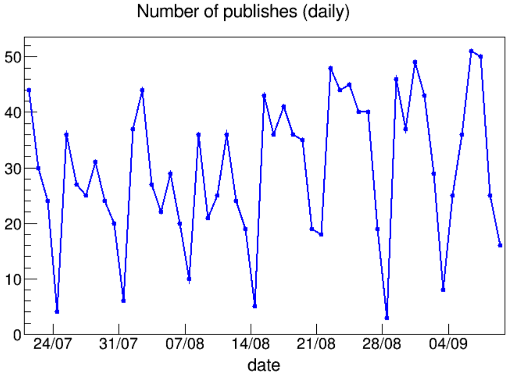


Figure: Number of `publish` operations per day from end of July to September 2022.

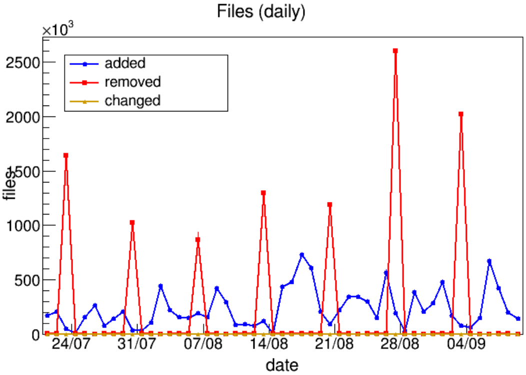


Figure: File addition/removal/changes per day from end of July to September 2022.

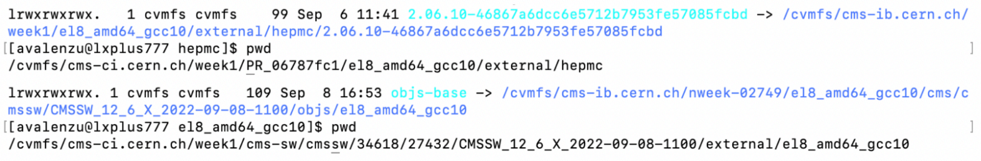
Deployment Workflow

- Externals are deployed while CMSSW is building, if modified.
- CMSSW is deployed from a `tar` file with the modified modules.

- Only testing for production architecture.
- Deployment time $<$ 5 min for externals and $<$ 2 min for CMSSW.

Lightweight repository

External packages (by version) are symlinked to `/cvmfs/cms-ib.cern.ch`.



```
lrwxrwxrwx.   1 cvmfs cvmfs    99 Sep  6 11:41 2.06.10-46867a6dcc6e5712b7953fe57085fcbd -> /cvmfs/cms-ib.cern.ch/
week1/el8_amd64_gcc10/external/hepmc/2.06.10-46867a6dcc6e5712b7953fe57085fcbd
[[avalenzu@lxplus777 hepmc]$ pwd                                                                             ]
/cvmfs/cms-ci.cern.ch/week1/PR_06787fc1/el8_amd64_gcc10/external/hepmc

lrwxrwxrwx. 1 cvmfs cvmfs   109 Sep  8 16:53 objs-base -> /cvmfs/cms-ib.cern.ch/nweek-02749/el8_amd64_gcc10/cms/c
mssw/CMSSW_12_6_X_2022-09-08-1100/objs/el8_amd64_gcc10
[[avalenzu@lxplus777 el8_amd64_gcc10]$ pwd                                                                   ]
/cvmfs/cms-ci.cern.ch/week1/cms-sw/cmssw/34618/27432/CMSSW_12_6_X_2022-09-08-1100/external/el8_amd64_gcc10
```

Distribution of CMSSW Container Images

- **CMSSW environment** images are distributed via CernVM-FS in `unpacked.cern.ch`.
- Jenkins also provides the infrastructure to build container images.
- There are jobs scheduled for triggering a new build of the container if:
  - Base image is updated.
  - Some selected `rpms` are updated, e.g., `ssl` certificates, `grid` certificates.

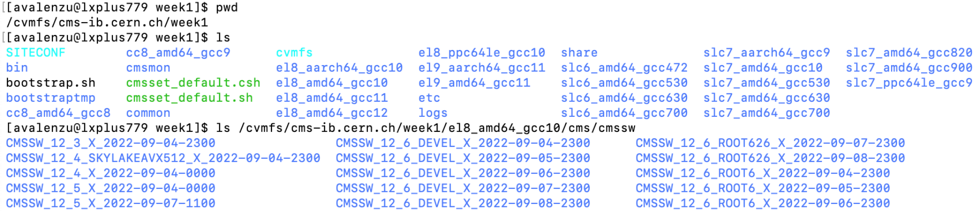
- Container images are heavily used for CI and IB jobs, but also by users.
- 500+ CMSSW container images deployed in `unpacked.cern.ch` under `/cvmfs/unpacked.cern.ch/registry.hub.docker.com/cmssw`.

Opportunity for improvements

- Increasing the frequency of the CernVM-FS job that deploys to `unpacked.cern.ch` will help to fix container images faster. It currently runs once a day.

Future plans & Improvements

- Moving to a multi-release manager setup to allow parallel publishing.
  - Artifacts are already published in different paths based on `<os_arch_compiler>`, release cycle and date.



```
[[avalenzu@lxplus779 week1]$ pwd
 /cvmfs/cms-ib.cern.ch/week1
[[avalenzu@lxplus779 week1]$ ls
 SITECONF        cc8_amd64_gcc9      cvmfs              el8_ppc64le_gcc10  share              slc7_aarch64_gcc9  slc7_amd64_gcc820
 bin             cmsmon              el8_aarch64_gcc10  el9_aarch64_gcc11  slc6_amd64_gcc472  slc7_amd64_gcc10   slc7_amd64_gcc900
 bootstrap.sh    cmsset_default.csh  el8_amd64_gcc10    el9_amd64_gcc11    slc6_amd64_gcc530  slc7_amd64_gcc530  slc7_ppc64le_gcc9
 bootstraptmp    cmsset_default.sh   el8_amd64_gcc11    etc                slc6_amd64_gcc630  slc7_amd64_gcc630
 cc8_amd64_gcc8  common              el8_amd64_gcc12    logs               slc6_amd64_gcc700  slc7_amd64_gcc700
 [avalenzu@lxplus779 week1]$ ls /cvmfs/cms-ib.cern.ch/week1/el8_amd64_gcc10/cms/cmssw
 CMSSW_12_3_X_2022-09-04-2300                CMSSW_12_6_DEVEL_X_2022-09-04-2300      CMSSW_12_6_ROOT626_X_2022-09-07-2300
 CMSSW_12_4_SKYLAKEAVX512_X_2022-09-04-2300  CMSSW_12_6_DEVEL_X_2022-09-05-2300      CMSSW_12_6_ROOT626_X_2022-09-08-2300
 CMSSW_12_4_X_2022-09-04-0000                CMSSW_12_6_DEVEL_X_2022-09-06-2300      CMSSW_12_6_ROOT6_X_2022-09-04-2300
 CMSSW_12_5_X_2022-09-04-0000                CMSSW_12_6_DEVEL_X_2022-09-07-2300      CMSSW_12_6_ROOT6_X_2022-09-05-2300
 CMSSW_12_5_X_2022-09-07-1100                CMSSW_12_6_DEVEL_X_2022-09-08-2300      CMSSW_12_6_ROOT6_X_2022-09-06-2300
```

  - Support CI testing for `aarch` and `powerpc` architectures.
  - Avoid blocking the release manager with non-production architectures.
  - Get rid of the current bottleneck in the deployment of IBs.
- Support more than 2 weeks of IBs, which will imply an increase of data on the repository.

Conclusion

- CernVM-FS and containers technologies are crucial for CMSSW.
  - It helps in development, distribution and preservation of software.
- We are happy with CernVM-FS performance.
  - It is a highly reliable service that CMS uses extensively.
  - Bottlenecks in `publish` were not due to CernVM-FS performance.
- S3 storage improved performance in garbage collection.
- We solved manual sub-cataloging by using nested catalogs with `.cvmfsdirtab`.

Questions? :)