CernVM-FS Release 2.6

J Blomer for the CernVM Team
SFT Meeting
15 April 2019
Reminder: Release Plan

Release 2.5
- Gateway service
- AWSv4 protocol support for S3 backend
- Smart automatic garbage collection
- Automatic handling of DNS server change

Release 2.6
- Shrinkwrap utility for HPC
- Publish metrics
- Direct tarball ingestion
- Container publishing service
- Notification service
CernVM-FS 2.6: New Satellite Services

- Tarball Ingestion
- Container Publishing
- Notification Service
- Call Tracer
- Repository Statistics
- Shrinkwrap
New services are still experimental!
To attract users, they need to stabilize towards the 2.7 release
Official UNCVMS: export bulky /cvmfs subtrees into “fat containers”.
Requested by ATLAS and CMS for US HPCs, also used by IT/HEPiX benchmark working group.

cvmfs_shrinkwrap -r sft.cern.ch \
 -t sft.cern.ch.spec \
 -z /export/cvmfs ...

sft.cern.ch.spec
/lcg/releases/ROOT/6.16.00-fcdd1/*
/lcg/releases/gcc/*
...

/export/cvmfs/.provenance/...
/export/cvmfs/.data/...
/export/cvmfs/sft.cern.ch/...

Compared to rsync:
- Faster: 50 MB/s vs. 30 MB/s
- Data de-duplication through hardlinks
- Efficient synchronization and GC
- Aware of CernVM-FS specifics

Shrinkwrapping is a rather heavy-weight process, dedicated “bridge nodes” recommended.
Precise, file-system level trace of /cvmfs accesses

1. Specification input for `cvmfs_shrinkwrap`
2. Instrumentation tool for benchmark analysis

```bash
$ echo "CVMFS_TRACEFILE=/tmp/trace.@fqrn@.csv" > /etc/cvmfs/default.local
$ mount -t cvmfs repo.cvmfs.io /cvmfs/repo.cvmfs.io
$ # Run testee from /cvmfs/repo.cvmfs.io
$ sudo cvmfs_talk -i repo.cvmfs.io tracebuffer flush
```

CSV

```
"1555099772803.948","-1","Tracer","Trace buffer created"
"1555099776596.462","6","","getattr()"
"1555099776596.700","2","","opendir()"
"1555099776599.053","4","/lcg","lookup()"
"1555099777187.145","2","/lcg","opendir()"
"1555099777351.414","4","/lcg/app","lookup()"
```

CSV
Precise, file-system level trace of /cvmfs accesses

1. Specification input for `cvmfs_shrinkwrap`
2. Instrumentation tool for benchmark analysis

```
$ echo "CVMFS_TRACEFILE=/tmp/trace.@fqrn@.csv" > /etc/cvmfs/default.local
$ mount -t cvmfs repo.cvmfs.io /cvmfs/repo.cvmfs.io
$ sudo cvmfs_talk -i repo.cvmfs.io tracebuffer flush
```

CSV

"1555099772803.948","-1","Tracer","Trace buffer created"
"1555099776596.462","6","","getattr()"
"1555099776596.700","2","","opendir()"
"1555099777187.145","2","/lcg","lookup()"
"155509977599.053","4","/lcg","lookup()"
"155509977187.145","2","/lcg","opendir()"
"155509977351.414","4","/lcg/app","lookup()"

Additional future columns: **pid, uid**

* e.g. to separate pilot from payload
**CernVM-FS Container Integration**

- **Goal**: avoid network congestion by starting unpacked containers from CernVM-FS
- Client / worker node: requires CernVM-FS plug-ins for
  - Docker (available)
  - Singularity (only for unprivileged access, planned)
  - containerd (in contact with upstream developers)
- CernVM-FS repository: efficient publishing of containers

**Container Publishing Service**

Add-on service on the publisher node to facilitate container conversion from a Docker registry

[Simone's SFT presentation]
Container Publishing Service: Workflow

Wishlist  https://gitlab.cern.ch/unpacked/sync

`version: 1
user: cvmfsunpacker
cvmfs_repo: "unpacked.cern.ch"
output_format: >
  https://gitlab-registry.cern.ch/unpacked/sync/$(image)
input:
  - 'https://registry.hub.docker.com/library/fedora:latest'
  - 'https://registry.hub.docker.com/library/debian:stable'
  - 'https://registry.hub.docker.com/library/centos:latest'

Push Docker Image  \--------\  MR to Wishlist  \--------\  Unpack and Publish  \--------\  Push Thin Image

User  \--------\  CernVM-FS

Wishlist  https://gitlab.cern.ch/unpacked/sync

```
# Singularity
/registry.hub.docker.com/fedora:latest -> \
/cvmfs/unpacked.cern.ch/.flat/d0/d0932...
# Docker with thin image
/.layers/f0/1af7...
```

Currently ∼25 test images available for ATLAS and CMS

**Compared to experiment repositories: expected increase of scale by an order of magnitude**

- Expect 1 final image per analysis → 1000 – 10 000 images / year
- 250 M to 2.5 B files per year, 5 TB to 50 TB / year  [250 k files and 5 GB per image]
- Garbage collection required for image development phase
Container Publishing Service: Workflow

Wishlist  https://gitlab.cern.ch/unpacked/sync

version: 1
user: cvmsunpacker
cvmfs_repo: 'unpacked.cern.ch'
output_format: >
  https://gitlab-registry.cern.ch/unpacked/sync/$(image)
input:
- 'https://registry.hub.docker.com/library/fedora:latest'
- 'https://registry.hub.docker.com/library/debian:stable'
- 'https://registry.hub.docker.com/library/centos:latest'

Future option: wildcard images, e.g. https://registry.hub.docker.com/library/fedora:*

契今日, 25 test images available for ATLAS and CMS

Compared to experiment repositories: expected increase of scale by an order of magnitude

- Expect 1 final image per analysis → 1000 – 10 000 images / year
- 250 M to 2.5 B files per year, 5 TB to 50 TB / year  [250 k files and 5 GB per image]
- Garbage collection required for image development phase
Enabling Feature for Container Publishing: Tarball Ingestion

Direct path for the common pattern of publishing tarball contents

$ cvmfs_server transaction
$ tar -xf ubuntu.tar.gz
$ cvmfs_server publish

$ zcat ubuntu.tar.gz | \ 
cvmfs_server ingest -t -

Performance Example
Ubuntu 18.04 container – 4 GB in 250 k files: **56 s untar + 1 min publish** vs. **74s ingest**
Notification Service

Fast distribution channel for repository manifest: useful for CI pipelines, data QA

- Optional service supporting a regular repository
- Publish/subscribe utility in `cvmfs_swissknife`
- Subscribe component integrated with the client, automatic reload on changes

→ CernVM-FS writing remains asynchronous but with fast response time in $O(\text{seconds})$
Better tooling for maintainers of heavy-duty repositories, requested by LHCb

- New feature: every transaction logs key metrics, e.g. # files, upload volume, etc.
- Stored in SQLite database, accessible to ROOT

```cpp
auto rdf = ROOT::RDF::MakeSqliteDataFrame(
    "/var/spool/cvmfs/sft-nightlies.cern.ch/stats.db",
    "SELECT * FROM publish_statistics;"
);

// ...
```

→ Enables repository insights and quality monitoring
Repository Statistics: File De-Duplication

sft-nightlies.cern.ch, 2019-04-10 – 2019-04-12

Preliminary, still sorting out some accounting irregularities
Repository Statistics: Data Compression and De-Duplication

sft-nightlies.cern.ch, 2019-04-10 – 2019-04-12

Preliminary, still sorting out some accounting irregularities
Repository Statistics: Publish Performance

sft-nightlies.cern.ch, 2019-04-10 – 2019-04-12

Preliminary, still sorting out some accounting irregularities

Not limited by the software but by the current centrally provided storage backend
This chart shows the number of issues **created** vs. the number of issues **resolved** in the last **200** days.
**Release Plan**

### 2.6: Released features
- Shrinkwrap utility for HPC [N Hazekamp, S Teuber]
- Publish metrics [D-F Dosaru]
- Direct tarball ingestion [Simone]
- Container publishing service [Simone]
- Notification service [Radu]

### Planning for the future: 2.7
- Renovated `cvmfs_server` tool suite
- Publish in ephemeral container (*experimental*)
- Fully unprivileged fuse client
Release Plan

2.6: Released features
- Shrinkwrap utility for HPC [N Hazekamp, S Teuber]
- Publish metrics [D-F Dosaru]
- Direct tarball ingestion [Simone]
- Container publishing service [Simone]
- Notification service [Radu]

Planning for the future: 2.7
- Renovated `cvmfs_server` tool suite
- Publish in ephemeral container (experimental)
- Fully unprivileged fuse client
## 2.6: Released features

- Shrinkwrap utility for HPC [N Hazekamp, S Teuber]
- Publish metrics [D-F Dosaru]
- Direct tarball ingestion [Simone]
- Container publishing service [Simone]
- Notification service [Radu]

**touched ~30 000 code lines**

## Planning for the future: 2.7

- Renovated `cvmfs_server` tool suite
- Publish in ephemeral container *(experimental)*
- Fully unprivileged fuse client

**Next months: critically low head count!**
### Outlook to CernVM-FS 2.7

<table>
<thead>
<tr>
<th><strong>Ephemeral Publish Container</strong></th>
<th><strong>Unprivilegded Fuse Client</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Eliminate the need for dedicated publisher nodes</td>
<td>Leap in support of opportunistic resources</td>
</tr>
</tbody>
</table>

Ephemeral Publish Container:

- $ cvmfs enter hsf.cvmfs.io /users/joe
  - ...Opens a shell in an ephemeral container with write access to the repository
- $ cvmfs publish
  - ...Back to read-only mode

- Requires the gateway service
- Will require major renovation in the `cvmfs_server` tool chain
- Will enable cvmfs publisher clusters (e.g. “lxcvmfs”)

Unpriviledged Fuse Client:

- Only privileged operation required: `mount()`
  - Currently handled by `fuse suid` binary
  - Reason why cvmfs needs to be “installed”
- As of RHEL8 with new kernel and `libfuse3`:
  - Limitations on `mount()` lifted
  - **Possibility of a “super-pilot” comprising cvmfs and singularity**
- CernVM-FS 2.6 released several new satellite services supporting **HPC sites, container-based workflows**, and the **publishing process**

- New functionality will stabilize in patch releases during the upcoming months

- CernVM-FS 2.7: revisit client and server intrinsics in order to better exploit opportunistic resources and to provide more flexible publishing workflows

- **CernVM Workshop 2019**
  
  **Dates:** June 3 – 6
  
  **Themes:** Serverless computing, HPC integration, container integration
  
  **Confirmed Speakers:** Harris Hancock (CloudFlare), Jesse Williamson (SuSE), Dorian Krause (Jülich), Michael Bauer (Singularity)