Software Distribution with CernVM-FS

Carlos Fenoy
Agenda

pREDi introduction

What is CVMFS

How It Works

Use Cases
pRED Informatics – Scientific informatics experts

*We are:*

**Scientists and informatics professionals united in a single organization**

**The scientific data experts within Roche R&D**

**Connecting research, knowledge and people across Roche R&D**

**Information technology scouts for Roche R&D**
pRED Informatics Organisation

New York City
USA

Welwyn Garden City
UK

Basel, Zurich
CH

Copenhagen
DK

Munich
DE

Shanghai
CN

Status January 2016
pRED clusters

2 main clusters
- Europe
- USA

1 Application filesystem for each geographical zone.

All servers and users workstations mount this filesystem through NFS.

All applications managed by EasyBuild and exposed to the users with Lmod

Currently ~1000 modules (applications and libraries)
Problems accessing applications

NFS over WAN is not performing

time ml avail | time (echo {} | python -m json.tool)
---|---
| NFS | NFS
| real | real
| 3m50.889s | 0m14.233s
| user | user
| 0m1.924s | 0m0.119s
| sys | sys
| 0m0.677s | 0m0.031s
CernVM-FS to the rescue!
What is CernVM-FS

The CernVM File System provides a **scalable**, **reliable** and low-maintenance software distribution service.

- **Scalable**
  - Horizontally scalable with mirrors
  - Heavily cached

- **Reliable:**
  - Data integrity verified with cryptographic hashes
  - Failover ability using multiple servers
  - Loosely coupled

https://cernvm.cern.ch/portal/filesystem
What is CernVM-FS

- Implemented as a POSIX Read Only filesystem using FUSE.
- Based on content-addressable storage and Merkle trees.
- Uses outgoing HTTP connections only (no firewall issues).
- Data and meta-data is transferred on demand.
CVMFS concepts

Stratum 0: Source of the repository

Stratum 1: 0 or more replica servers for reliability and load balancing

Manifest: File containing fundamental data about the repository

Catalog: SQLite database with the repository files metadata contents
How it works

Creating a repository

Adding files

Publishing files
How it works

Creating a repository

cmvfs_server mkfs myapps.example.com

- Creates repository structure and adds an entry in apache to serve the files.
- Creates the key pair to sign the files
- Creates config files for server and clients.
- Creates the first version of the repository
- Mounts the repository under /cvmfs/<FQRN> (fully qualified repository name)
Stratum 0

new_repository
How it Works

Adding files to a repository

cvmfs_server transaction myapps.example.com

Creates a new writable layer and overlays it on top of the current repository
How it Works

Publishing the changes

cvmfs_server publish myapps.example.com

Synchronises the changes with the repository
root

- new_repository

- new_application

Stratum 0
Mounting on clients

- copy the repository public key to
  `/etc/cvmfs/keys/myapps.example.com.pub`
- create the repository config file in
  `/etc/cvmfs/config.d/myapps.example.com.conf`

```
CVMFS_SERVER_URL=http://stratum0.example.com/cvmfs/@fqrn@
CVMFS_HTTP_PROXY=DIRECT
```

- mount

```
mount -t cvmfs myapps.example.com /cvmfs/myapps.example.com
cvmfs2 myapps.example.com /cvmfs/myapps.example.com
```
Scaling it up

Prepare a server to become a Stratum 1

cvmfs_server add-replica
    http://stratum0.example.com/cvmfs/myapps.example.com

Start the synchronization

cvmfs_server snapshot myapps.example.com
A bit about snapshots

One of the strengths of CernVM-FS is the usage of snapshots/revisions.

Revisions allow to access the filesystem as it was at any point in time.

# cvmfs_server tag

<table>
<thead>
<tr>
<th>Name</th>
<th>Revision</th>
<th>Channel</th>
<th>Timestamp</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>trunk-previous</td>
<td>2</td>
<td>trunk</td>
<td>12 Jun 2017 10:42:03</td>
<td>default undo target</td>
</tr>
<tr>
<td>trunk</td>
<td>3</td>
<td>trunk</td>
<td>12 Jun 2017 18:57:03</td>
<td>current HEAD</td>
</tr>
</tbody>
</table>

Listing contains 5 tags.
Use cases

- Remote sites
- Docker
Remote sites

CVMFS allows us to provide fast access to applications on remote sites

time (echo {} | python -m json.tool)

<table>
<thead>
<tr>
<th></th>
<th>NFS</th>
<th>CernVM-FS</th>
</tr>
</thead>
<tbody>
<tr>
<td>real</td>
<td>0m14.233s</td>
<td>0m0.426s</td>
</tr>
<tr>
<td>user</td>
<td>0m0.119s</td>
<td>0m0.036s</td>
</tr>
<tr>
<td>sys</td>
<td>0m0.031s</td>
<td>0m0.019s</td>
</tr>
</tbody>
</table>
```
time ml avail

<table>
<thead>
<tr>
<th></th>
<th>NFS</th>
<th>CVMFS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>real</strong></td>
<td>3m50.889s</td>
<td>0m6.102s</td>
</tr>
<tr>
<td><strong>user</strong></td>
<td>0m1.924s</td>
<td>0m1.750s</td>
</tr>
<tr>
<td><strong>sys</strong></td>
<td>0m0.677s</td>
<td>0m0.277s</td>
</tr>
</tbody>
</table>

~1000 modules
```
Docker

- Scientists find it difficult to create docker containers with their applications
- Creating a docker image for each individual case takes a lot of effort

- CVMFS provides a Docker volume manager
- A docker container can mount a cvmfs repository easily

```bash
docker run -it --volume-driver cvmfs -v myapps.example.com:/myapps centos:7
```
Scientific reproducibility

- Scientists want to be able to reproduce their experiments
- This clashes with the installation of new and decommissioning of old applications.

- CVMFS repository revisions can be used to ensure scientific reproducibility
- A user can mount an specific revision at any time

```bash
docker run -it --volume-driver cvmfs -v myapps.example.com@trunk-previous:/myapps centos:7
```
Conclusions

- CernVM-FS is a good alternative to NFS for application distribution
  - only outgoing HTTP connections
  - loosely coupled mount
  - aggressive cache at multiple levels
  - easy to setup
Followups

• CernVM-FS can be used as a substitute docker graph driver (next release)
  – reduce size of storage used by docker images by removing the layer concept
• Create a VM for scientists to run their applications anywhere using CernVM-FS to server basic filesystem and applications.
Acknowledgements

CernVM-FS Development Team
(https://cernvm.cern.ch/portal/filesystem)

EasyBuild Team
(https://easybuild.readthedocs.io)

Lmod developers
(https://www.tacc.utexas.edu/research-development/tacc-projects/lmod)
Doing now what patients need next