
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:

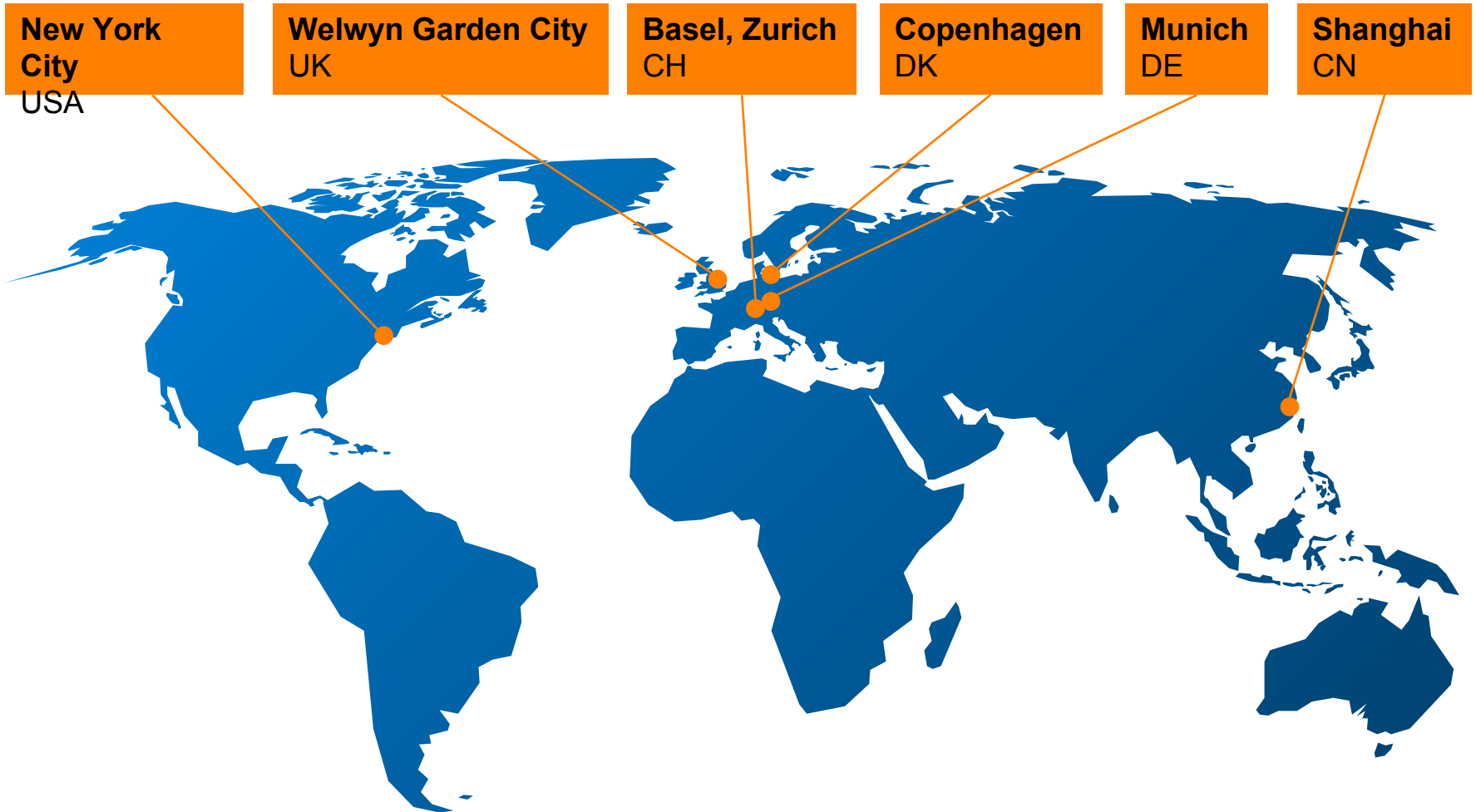
Scientists and informatics professionals united in a single organization

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



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

<code>time ml avail</code>
NFS
real 3m50.889s user 0m1.924s sys 0m0.677s

<code>time (echo {} python -m json.tool)</code>
NFS
real 0m14.233s user 0m0.119s sys 0m0.031s

CernVM-FS to the rescue!

What is CernVM-FS

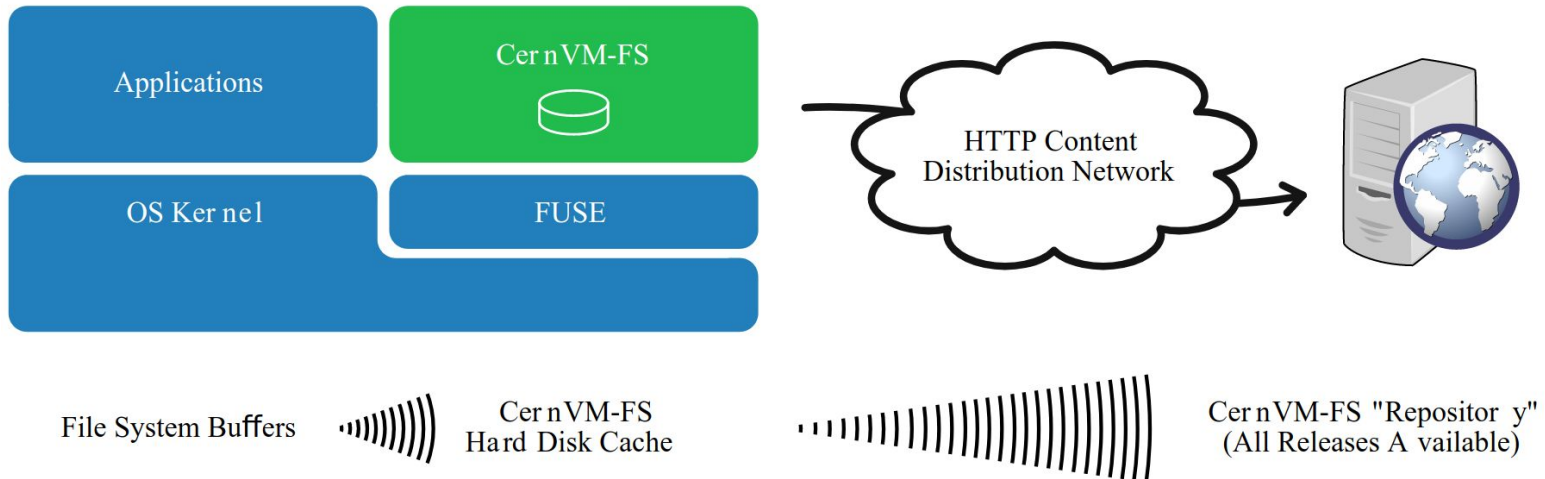
The CernVM File System provides a **scalable**, **reliable** and low-maintenance software distribution service.

<https://cernvm.cern.ch/portal/filesystem>

- Scalable
 - Horizontally scalable with mirrors
 - Heavily cached
- Reliable:
 - Data integrity verified with cryptographic hashes
 - Failover ability using multiple servers
 - Loosely coupled

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

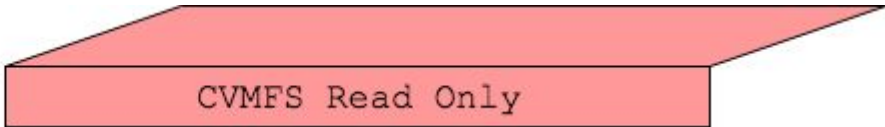
How it works

Creating a repository

```
cvmfs_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)

`/cvmfs/myapps.example.com` →



CVMFS Read Only

Stratum 0



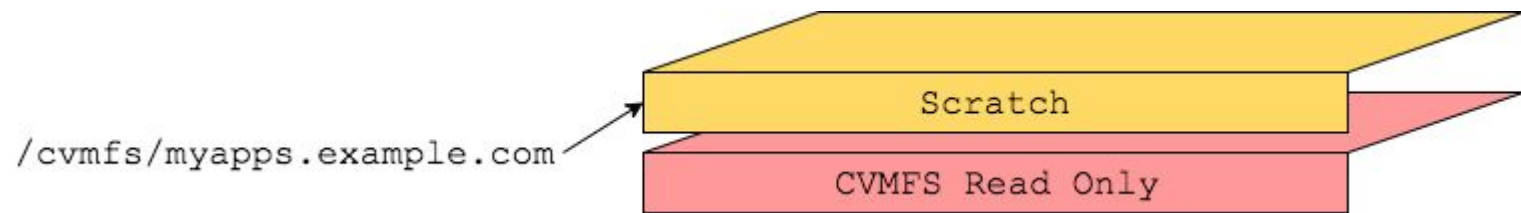
new_repository

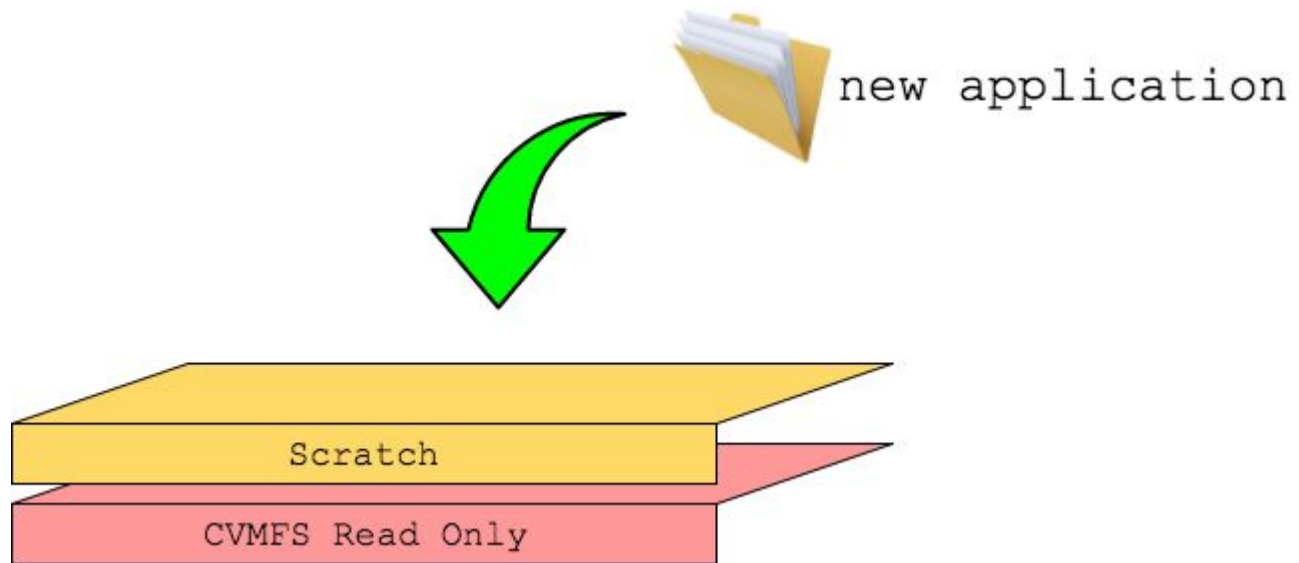
How it Works

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





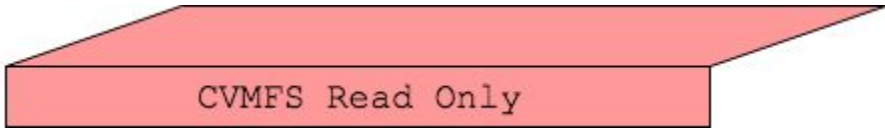
How it Works

Publishing the changes

```
cvmfs_server publish myapps.example.com
```

Synchronises the changes with the repository

```
/cvmfs/myapps.example.com →
```

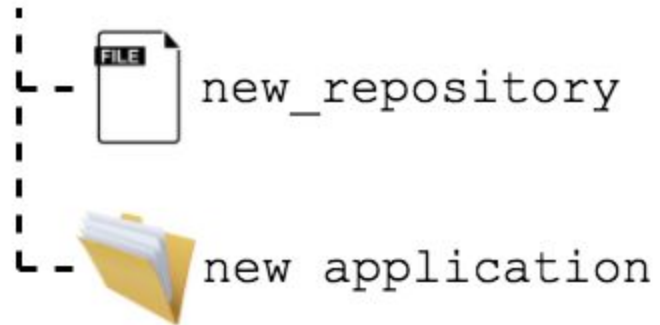
A diagram showing a red 3D trapezoidal shape representing a repository. The top surface is a trapezoid, and the bottom surface is a rectangle. The text "CVMFS Read Only" is written on the bottom surface. An arrow from the text "/cvmfs/myapps.example.com" points to the left vertical edge of the bottom surface.

CVMFS Read Only

Stratum 0



root



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`

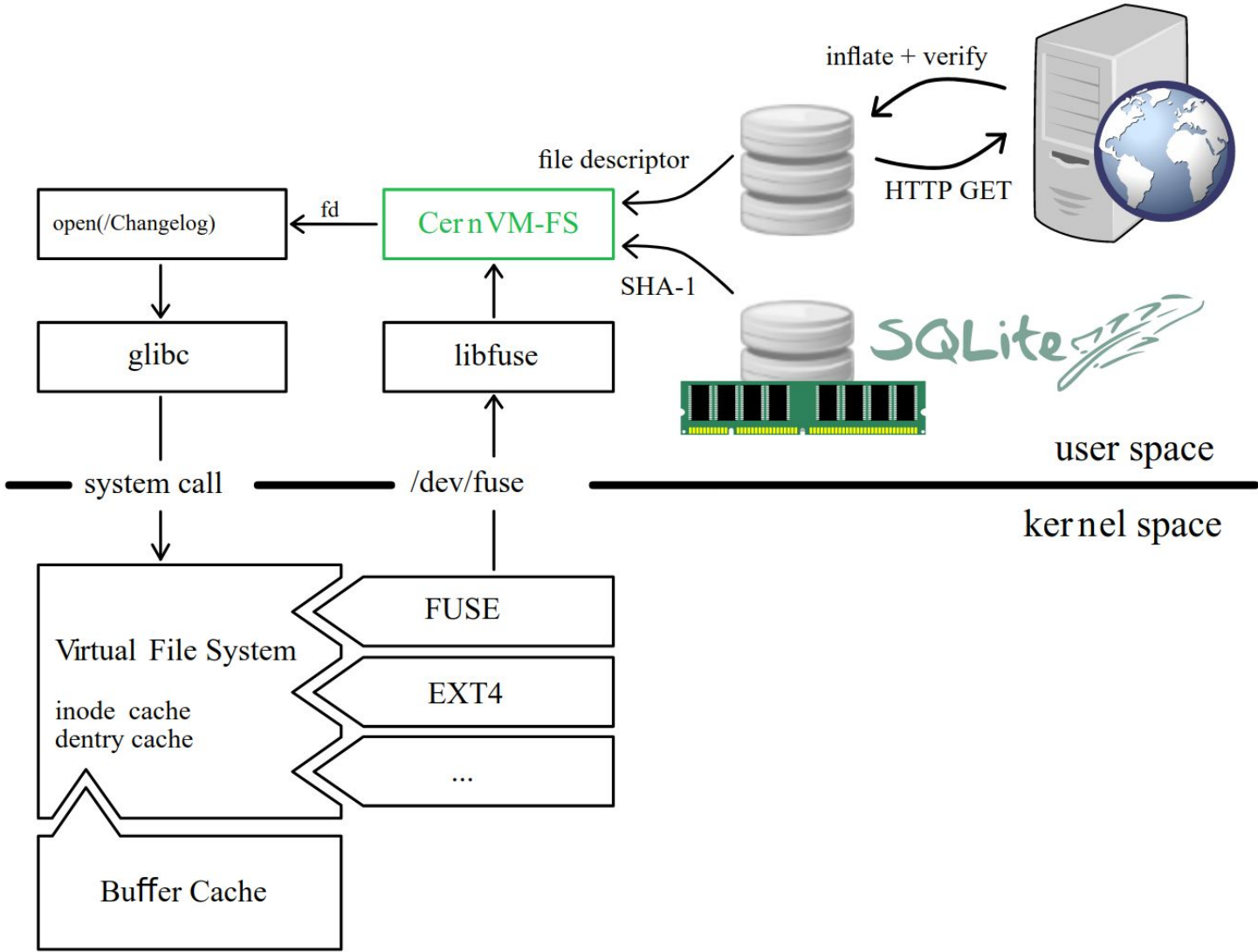
```
CVMFS_SERVER_URL=http://stratum0.example.com/cvmfs/@fqrn@
```

```
CVMFS_HTTP_PROXY=DIRECT
```

- **mount**

```
mount -t cvmfs myapps.example.com /cvmfs/myapps.example.com
```

```
cvmfs2 myapps.example.com /cvmfs/myapps.example.com
```



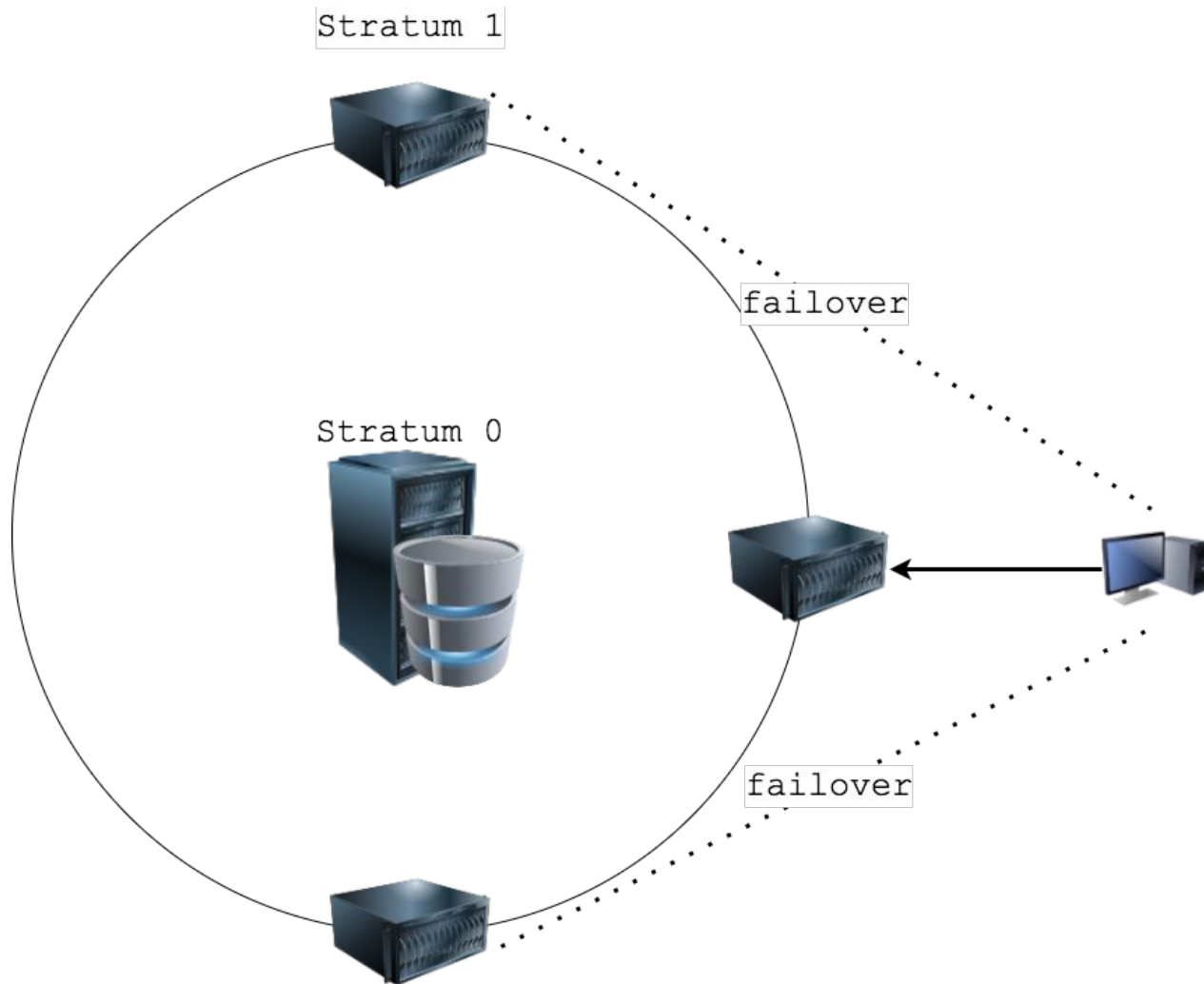
Scaling it up

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
```

Name	Revision	Channel	Timestamp	Description
generic-2017-06-12T08:32:29Z	1	trunk	12 Jun 2017 10:32:29	
trunk-previous	2	trunk	12 Jun 2017 10:42:03	default undo target
generic-2017-06-12T08:38:35Z	2	trunk	12 Jun 2017 10:42:03	
trunk	3	trunk	12 Jun 2017 18:57:03	current HEAD
generic-2017-06-12T16:19:47Z	3	trunk	12 Jun 2017 18:57:03	

```
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)	
NFS	CernVM-FS
real 0m14.233s user 0m0.119s sys 0m0.031s	real 0m0.426s user 0m0.036s sys 0m0.019s

time ml avail	
NFS	CVMFS
real 3m50.889s user 0m1.924s sys 0m0.677s	real 0m6.102s user 0m1.750s sys 0m0.277s

~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

```
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

```
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***