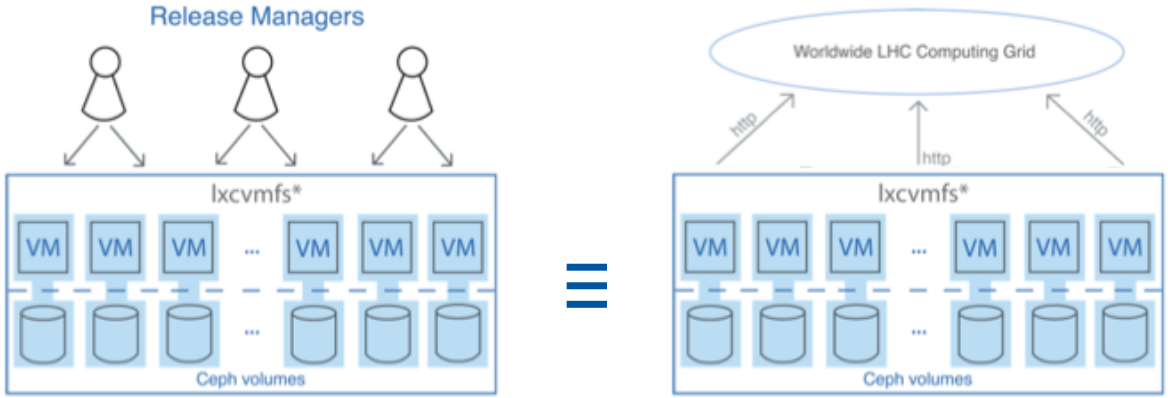
Containerized CVMFS Server

Dan van der Ster (CERN IT-ST)
Lillian Huang (Former CERN IT-ST)

CernVM Users Workshop – 31 Jan 2018

Repo Storage vs Publishing



Repo publishing work is done on the same VM as the storage

Several shared lxcvmfs* machines:
- Around 25 machines for ~40 repos
- New repo needs a new VM / data volume

Idle repos waste CPUs
Can't scale CPU or storage independently
Stratum 0 VMs are each a SPoF
Customized publishing environments are difficult

Proposed Stratum 0 Model

- Separate the storage and the compute
  - Focus on highly available, scalable storage
  - Release managers push changes from remote “build” nodes or workstations
- Example storage could be S3 or HA filers
- Publishing side could be lxplus or Jenkins or dedicated build clusters, etc.
  - Key is to containerize the release manager processes

Proof of Concept

1. Create an S3-backed repository

```
cvmfs_server mkfs -s /etc/cvmfs/cvmfs_stratum0_testng/mys3.conf \
    -w http://cs3.cern.ch/cvmfs-stratum0-testng testng.cern.ch
```

2. Dockerized cvmfs_server:

```
docker run -i -t -e REPO_NAME=$repo_name -v $command_file:/dockerfile_commands.sh \
   -v /var/spool/cvmfs -v $config_file_dir:/tmp/config_files:ro \
   --cap-add SYS_ADMIN --device /dev/fuse --rm $image_name
```

Dockerfile builds an image with cc7 + the cvmfs_server rpms

`dockerfile_commands.sh` instantiates the publishing env for a specific repo (conf + keys)

Conclusions

- Our student built this prototype in a few weeks; idea seems feasible.
- Simple benchmarks showed the S3 backend can be more performant for small-medium sized change-sets (up to *O(10^4)*),
  - Parallel uploads makes it quite fast even for large change-sets *O(10^5)*
- *Next step: sync with proposed publishing gateway model*