

CephFS and more in Bonn

A HTC cluster with CephFS, VMs on Ceph RBD with TRIM, differential backups and more in Bonn

Oliver Freyermuth, Peter Wienemann

University of Bonn
{freyermuth,wienemann}@physik.uni-bonn.de

17th September, 2019

Physics Institute at University of Bonn

- 240 members
- 1500 registered networked devices:
 - \approx 150 managed desktops, \approx 30 managed laptops
 - $>$ 50 managed servers offering $>$ 40 services
 - 41 HTC compute nodes
 - + hundreds 'unmanaged' Windows / MacOS X / Linux systems
- Biggest particle accelerator run by a German university (164.4 m circumference) with two experiments (\approx 50 people)
- Groups from High Energy Physics, Hadron physics, detector development, photonics, theory groups

HTC cluster and other institute-wide services needed

Our main use cases for Ceph

CephFS

growing HTC computing cluster (1120 cores, > 0.5 PB CephFS)
Erasure Coding ($k = 4$, $m = 2$) and Snappy compression

RBD

growing virtualization cluster (9 hypervisors, 40 VMs),
using libvirt & QEMU / KVM (managed via Foreman)
33 TB, 3 replicas across 3 buildings

RGW

testing as Backup storage, potentially also for CernVM-FS
3 replicas across 3 buildings

CephFS

- Old cluster with Lustre, 10 Gbit/s ethernet
 - Lustre never updated
 - Increasing number of issues (broken FIEMAP etc.)
- Successor HTC cluster: New FS, InfiniBand 56 Gbit/s
- Designed for Lustre / **BeeGFS**
 - Testing successful, well performing (RDMA)
 - Free license does not cover ACLs, quotas
 - Contributing to code hard / impossible
- ⇒ Switch to Plan B in Q1 2018: **CephFS**

Hardware setup

- 3 MON + MDS + OSD nodes, all with 128 GB RAM
 - 2 with 2 SSDs with 240 GB each (*NVMe upgrade in progress*)
 - **1 with 2 NVMeS with 1 TB each**
- 7+x OSD hosts
 - 6 hosts with 192 GB RAM:
 - 32 HDDs with 4 TB each
 - 2 SSDs with 120 GB each (DB+WAL)
 - ⇒ ***NVMe upgrade in progress***
 - **1 host with 256 GB RAM:**
 - 34 HDDs with 4 TB each
 - 2 NVMeS with 1 TB each (DB+WAL)
 - soon: new host with 256 GB RAM,
 - 32 disks, 12 TB each
 - 2 NVMeS with 1 TB each (DB+WAL)
- Metadata on SSD / NVMe device class, data on HDD

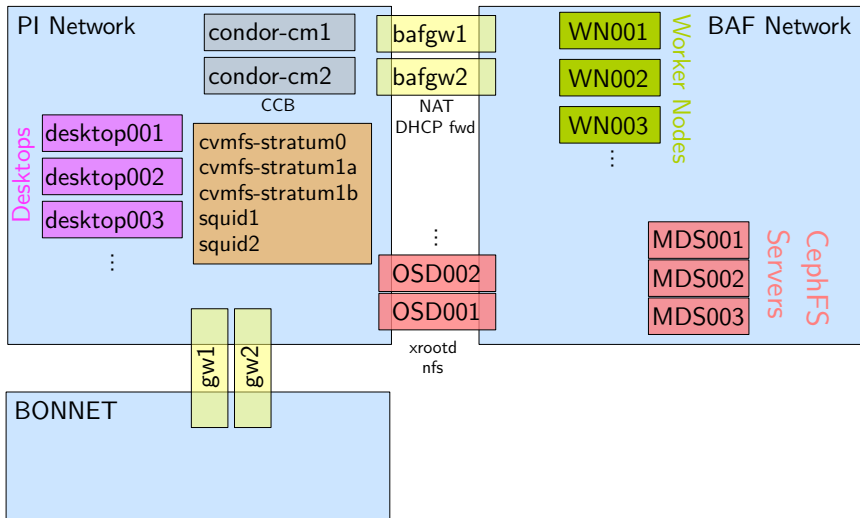
CephFS setup details

- Erasure Coding ($k = 4$, $m = 2$), Snappy compression
- All systems CentOS 7.6
- Export via NFSv4.2 to desktop machines (NFS Ganesha)
- Ceph-FUSE clients, Mimic 13.2.6 (we use quotas and ACLs)
- InfiniBand running with IPoIB (issues with RDMA), tuning yields good performance
- Grid connectivity (xrootd, WebDAV): 7 Gbit/s
- **Very** positive experience with mailing list
- Very stable operation, we already did (without downtime):
 - RAM upgrade of all servers
 - Extension: +1 disk server & +1 MON + MDS
 - Change of failure domain
 - HDD changes
 - hard lockup of (single) disk servers
 - Upgrade from Luminous to Mimic
 - Soon: Recreation of all OSDs when upgrading to NVMes

CephFS quota setup

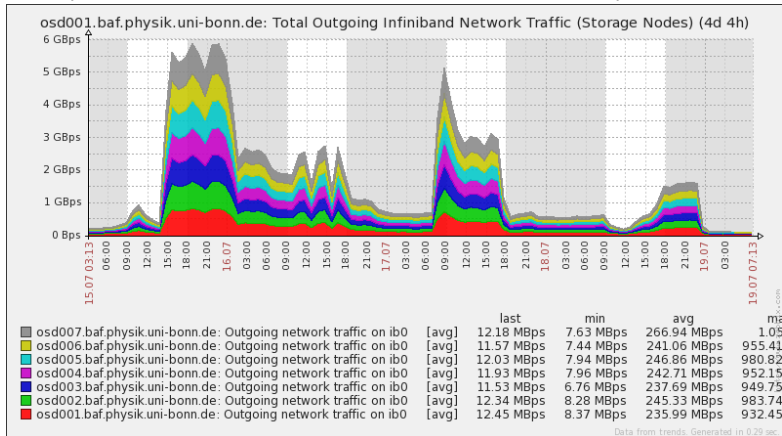
- Every user gets 500 GB + Grid storage
- File count limited to 100 000
- Our use case: Data storage, large files, mostly WORM (**W**rite-**O**nce, **R**ead-**M**any)
- Using Ceph-FUSE means slow syscalls — but FS should not store software etc., so throttling these is fine!
- Additionally, we offer CernVM-FS for software (<https://cvmfs.readthedocs.io>)
read-only FUSE-FS via HTTP ⇒ can also use S3 as backend

Network topology



CephFS details

- Effective sequential read throughput $> 3 \text{ GB/s}$, peaks of 5 GB/s
(Note: Network graph contains EC overhead!)



Ceph for virtualization (RBD)

- Past: SL6 systems with LVM on RAID 1, full daily backups
- Now: All systems CentOS 7.6
- Mimic 13.2.6
- Foreman-controlled Libvirt with RBD backend
- Ceph-FUSE clients for CephFS synchronizing libvirt XMLs
- Machines (currently) connected via 1 Gbit/s ethernet
- Writeback caching, unmap / discard:

```
1 <disk type='network' device='disk'>
2   <driver name='qemu' type='raw' cache='writeback' discard='unmap' />
3   <auth username='libvirt'>
4     <secret type='ceph' uuid='XXXX' />
5   </auth>
6   <source protocol='rbd' name='rbd/condor-ce.physik.uni-bonn.de-disk1'>
7     <host name='mon001.virt.physik.uni-bonn.de' port='6789' />
8     <host name='mon002.virt.physik.uni-bonn.de' port='6789' />
9     <host name='mon003.virt.physik.uni-bonn.de' port='6789' />
10  </source>
11  <target dev='sda' bus='scsi' />
12  <address type='drive' controller='0' bus='0' target='0' unit='0' />
13 </disk>
```

Ceph RBD writeback caching with VirtIO-SCSI

We tested the system for resilience. While VMs are writing, for more than 10 min:

- Pulling plugs of single to all(!) OSDs and MONs
⇒ Writing continued once Ceph cluster was back!
- Pulling plugs of hypervisor running the VM
Regular e2fsck run needed as expected.

Important gotchas (before you 'try this at home')

- Unmap / Discard only supported in virtio-scsi in LTS distros!
virtio-blk learned this in 2019: [Kernel commit](#), [QEMU commit](#)
- virtio-scsi is subject to 30s SCSI timeout, will not recover!
Fixed [in-kernel](#) in 2017, backported to RHEL 7

Ceph RBD hardware

- 3 MON + MDS nodes with 32 GB RAM
- 3 OSD nodes with 32 GB RAM
 - 5 HDDs with 4 TB each
 - 1 SSD with 240 GB each
- 3 OSD nodes with 64 GB RAM
 - 5 HDDs with 4 TB each
 - 2 SSDs with 1 TB each
- 3 replica configuration
- OSD nodes can house more HDDs
- Currently spread across 3 rooms in 2 buildings, soon 3 buildings ('datacenters')

Ceph RBD Backup

Backup with dailies, weeklies, few monthlies in form of snapshots (hot) and incrementally backed up (larger retention).

Backup Phase 1 (on each hypervisor node)

- 1 Instruct qemu-guest-agent to trim filesystems:

```
virsh domfstrim ${VM}
```

- 2 Instruct qemu-guest-agent to freeze filesystems:

```
virsh domfsfreeze ${VM}
```

- 3 Take snapshots of all block devices of the domain.
- 4 Thaw filesystems via qemu-guest-agent.

Ceph RBD Backup

Backup Phase 2 (on backup machine)

- 1 Back up all not yet backed up snapshots incrementally:
 - 1 Using Backy² (<http://backy2.com/>)
will soon fade this out (SQLite support broken)
 - 2 Using Benji backup (<https://benji-backup.me/>).
- 2 Remove old backups.
- 3 Scrub backups partially.
- 4 Remove old snapshots.

Ceph RBD Backup: Backy² and Benji

- Benji is a more active fork of Backy² with more features.
- Used by us:
 - Incremental RBD backup (using `rbdiff`), backs up to chunks with checksums
 - Strong compression with `zstandard` (Benji only)
 - Scrubbing backups
 - Mounting backups via NBD
- Not (yet) used:
 - Encrypting backups
- Backup to a machine with `ext4` on a RAID 6.
- Differential backups take a few seconds to minutes only!
- Restores to Ceph or raw images work very well.
- Testing Ceph RBD Mirroring to a separate cluster right now.

Ceph RBD Backup: Interesting observations

- For common VMs with low I/O (apart from automatic updates) number of backed up chunks scales with volume size.
- Backups compressible with ratios between 10 and above 100 using zstandard on level 22.
- Backups are **fast** (seconds to minutes per volume including sanity checks).

Space usage for 40 VMs

- 'Live' RBD with snapshots (4 monthly, 8 weekly, 14 daily): 1 TB
- Backy2 with about 1 month more data: 1.5 TB
- Benji with highest zstd level (22), 8 monthlies: 0.27 TB

Ceph RBD Backup: Interesting observations

- For common VMs with low I/O (apart from automatic updates) number of backed up chunks scales with volume size.
- Backups compressible with ratios between 10 and above 100 using `zstandard` on level 22.
- Backups are **fast** (seconds to minutes per volume including sanity checks).

Conclusions

- Backups are mostly chunks with `ext4` superblock copies.
- Compression helps **significantly** also when trimming: Only used parts of chunks backed up!
- For servers with low I/O turnaround: Cheap to keep months of backups.

Ceph-based backup system

- Started using Nautilus release just last week. . .
Very delighted by new Dashboard and PG scaling!
- Offering storage via RGW for:
 - Backups with Restic (<https://restic.net/>) from Linux
 - Backups from Windows, MacOS, Linux with Duplicati (<https://www.duplicati.com/>)
- First tests with a single MON, single OSD, single RGW setup very encouraging (backup speed of 50 MB/s and higher)
to be scaled and distributed across 3 buildings
- Discussing need for other interfaces on top of CephFS (SFTP, TimeMachine, xrootd), e.g. backup storage for local experiments
- Successfully using RBD-mirror to this cluster (data stream of ≈ 1 MB/s)
Looking forward to Octopus feature to mirror only snapshots without journaling overhead.

Conclusions

- CephFS works very well also for HTC clusters!
Note: Separate FS / HTCondor file transfer for software.
- Using writeback-caching and trim/discard with RBD works well.
- RBD backup using Benji and mirroring can be very space-efficient.
- Taking first successful steps with RGW as backup service now.
- The Ceph community and mailing lists are better than any commercial support I have encountered so far!

Thank you!

Thank you
for your attention!

