News from the IT department, Cluster best practices and a deep dive into file systems

PI IT Team

Antonio Figueiredo[1], *Oliver Freyermuth*, Frank Frommberger, *Michael Hübner*, Daniel Jonas[2], Ernst-Michail Limbach-Gorny, Andreas Wißkirchen & more helping hands in projects and from the HISKP IT Team

it-support@physik.uni-bonn.de

31st October, 2024

[1] started December 2023

[2] with us for 3 months

Outline

1. News
2. Funded IT R&D projects on Research Data Infrastructures
3. Cluster best practices
4. Behind the scenes: A deep dive into file systems

Personnel Changes

- since 2020: FTD IT position not filled

- Daniel Jonas is with us for 3 months (IT specialist trainee)

Project-specific helpers

- Development Team for web and database projects: Jan Heinrichs, Oliver But
- Research data infrastructure projects: N.N.
- IT specialist trainees: 3 months every year in cooperation with HRZ

(several personnel changes in the past years also in these projects)

Personnel Changes

- since 2020: FTD IT position not filled

- Daniel Jonas is with us for 3 months (IT specialist trainee)

PI Web team

Ian Brock, Florian Kirfel, Barbara Valeriani-Kaminski
(coordinating also with FTD, HISKP and department web teams)

Flexible project developers

Antonio Figueiredo

Projects (highlights)

Selection of projects

- still ongoing
  - Web development team: ongoing **development of HR system**
  - Joint project of PI & HISKP IT teams, secretaries, HRZ: **Indico**
  - Development of **common firewall** (HISKP, PI, FTD)
- new in 2024
  - Indico will be used for the interim's building
  - Autodesk licences are now distributed via licence server (if you need Autodesk contact us)
  - Work behind the scenes (construction planning, OS / software upgrades etc.)
  - Central HPC 'Marvin' open to everyone
- upcoming
  - Debian 12 for centrally managed desktops
  - New printer contract coming up, likely Autumn 2025

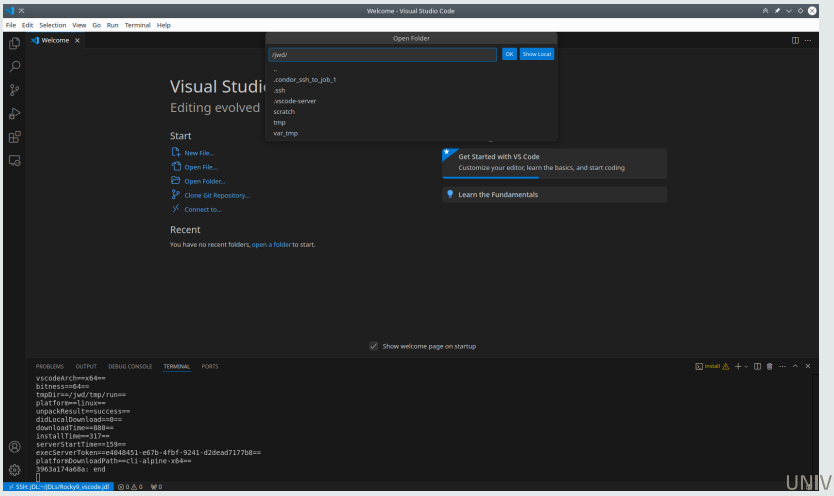
Projects (highlights)

Indico signage

Photo taken by and work being done by D. Jonas

Projects (highlights)

Remote editing on the cluster



Projects (highlights)

Registration process within the PI

- Electronic de-/registration with the institute
  - registration
  - de-registration
  - Important to fill this out ⇒ info will be used for business cards and mailing lists
  - Access to resources will be coupled to registration (e.g. keys, cluster access)
- Semi-automatic generation of business cards on websites (PI + PI people on Fachgruppen website)
  - PI website
  - Fachgruppen website (only used for PI personnel)
- Automatically filled mailing lists for all working groups
  - We will contact group leaders for the migration
  - Mailing lists are hosted on listen.uni-bonn.de (i.e. Sympa)

Projects (highlights): ROT building

- New server room with new cooling and network technology
- Dedicated printer rooms
- Upcoming: Move of CIP pool, all servers,...
- Rooms bookable via Indico (across faculties)
- Largest number of network outlets of all buildings of University of Bonn ($\mathcal{O}(4000)$)

Projects (highlights): ROT building




- Half of the racks with active fans for powerful cooling
- Each rack can cool up to 35 kW (but only 320 kW total for 16 racks)
- Note one rack can hold up to 48 thin servers which can produce over 1 kW each, power density keeps increasing

Some numbers from the past year...

- $>$ 70 procurement requests via IT (with hardware, software and license counseling), in many cases consisting of several orders
- $>$ 450 reinstalled machines (laptops for masterclasses, upgraded servers etc.)
- $>$ 540 managed Linux systems in total
- $>$ 40 managed Windows systems in total
- $>$ 50 non-trivial hardware issues (includes 23 broken hard disks in servers)
- newly created tickets (most contains dozens of back-and-forth mails):

1779	from October 2022 to October 2023
2206	from October 2023 to October 2024

(includes around 420 automatic, but actionable issues per year, excludes merged tickets, excludes most 'do-you-have-a-minute?' customers)

Funded Projects

Particles, Universe, NuClei and Hadrons for the NFDI

Activities in Bonn:

- JupyterHub frontend for federated Compute infrastructure ('Single Point of Entry')
- Including resources in Bonn in the Compute infrastructure
- Federated storage for 'small' experiments

$\Rightarrow$ half-time for project, handed in midterm report a few weeks ago

FIDIUM

- Federated Digital Infrastructures for Research on Universe and Matter
- Entered new funding phase, preparing for new project call beginning of next year
- Will focus efforts in Bonn on sustainability and checkpointing

Where to begin?

Pop Quiz

I just started my thesis in a research group and inherited some code from another student. Where to begin?

Ideas

Where to begin?

Pop Quiz

I just started my thesis in a research group and inherited some code from another student. Where to begin?

Ideas

- Have a look at that person's notes / documentation
- Try out their 'run' command and see if I can reproduce their results
- ...

Where to begin?

Pop Quiz

I just started my thesis in a research group and inherited some code from another student. Where to begin?

Answer?

$\Rightarrow$ **Yes, but...**

Where to begin?

Pop Quiz

I just started my thesis in a research group and inherited some code from another student. Where to begin?

Real answer

- All of the above are important things to do
- But: Do I understand what is happening in the code and how the workflow works?
- Start smaller, i.e.
  - Try to run the code locally on a small data subset
  - Run the code in an interactive job and try to manually follow the workflow steps
  - Run a single batch job and monitor the resource consumption
  - Run larger sets of batch jobs and monitor the resource consumption (you may observe outliers depending on the data being processed)

Where to begin?

First steps

- Check our documentation
- Have a look at our HTCondor tutorial with examples for you try out
- More in-depth descriptions of available software
  - There's also welcome pages specifically for new ATLAS / Belle II users
- Some pitfalls to be aware of, e.g. Kerberos TGT

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the ‘dry run’ options

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the 'dry run' options

```
$ condor_submit rocky8.jdl -dry-run /dev/stdout
Dry-Run job(s)
ClusterId=1
RequestCpus=1
...
.
1 job(s) dry-run to cluster 1.
```

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the 'dry run' options

```
$ condor_submit rocky8.jdl -dry-run /dev/stdout
Dry-Run job(s)

ERROR: Executable file /home/.../hello_world1.sh does not exist
```

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the 'dry run' options
- My job does not start ⇒ Do my resource requirements exceed the available resources? Or do I just have to wait? Take a look at our last talk about HTCondor

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the 'dry run' options
- My job does not start ⇒ Do my resource requirements exceed the available resources? Or do I just have to wait? Take a look at our last talk about HTCondor

```
$ condor_q -better-analyze 300429.0
         Slots
Step    Matched  Condition
-----  --------  ---------
...
[5]        2759  TARGET.Memory >= RequestMemory
[7]           0  TARGET.Cpus >= RequestCpus
...
```

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the ‘dry run’ options
- My job does not start ⇒ Do my resource requirements exceed the available resources? Or do I just have to wait? Take a look at our last talk about HTCondor
- My job gets put on Hold ⇒ look at the LastHoldReason

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the 'dry run' options
- My job does not start ⇒ Do my resource requirements exceed the available resources? Or do I just have to wait? Take a look at our last talk about HTCondor
- My job gets put on Hold ⇒ look at the LastHoldReason

```
$ condor_q -long xyz.0 | grep LastHoldReason
LastHoldReason = "Error from
 ↪  slot1_114@wn045.baf.physik.uni-bonn.de: STARTER at SOME_IP
 ↪  failed to send file(s) to <SOME_IP>: error reading from
 ↪  /pool/condor/dir_717394/my_output: (errno 2) No such file or
 ↪  directory; SHADOW failed to receive file(s) from <SOME_IP>"
```

How to debug?

Submit workflows

- My JDL file is not working ⇒ try out the ‘dry run’ options
- My job does not start ⇒ Do my resource requirements exceed the available resources? Or do I just have to wait? Take a look at our last talk about HTCondor
- My job gets put on Hold ⇒ look at the LastHoldReason

How to debug?

Analysis code - Some general pointers

- Inspect log files
- Check resource usage of jobs

```
$ condor_q -constraint 'JobStatus == 2' -af:hj ResidentSetSize_RAW RequestMemory DiskUsage_RAW RequestDisk RequestCPUs
↪  CPUsUsage
 ID         ResidentSetSize_RAW RequestMemory DiskUsage_RAW RequestDisk RequestCPUs CPUsUsage
SOME_ID.0   432360              8192          31678         8388608     2           0.05062805099033083
```

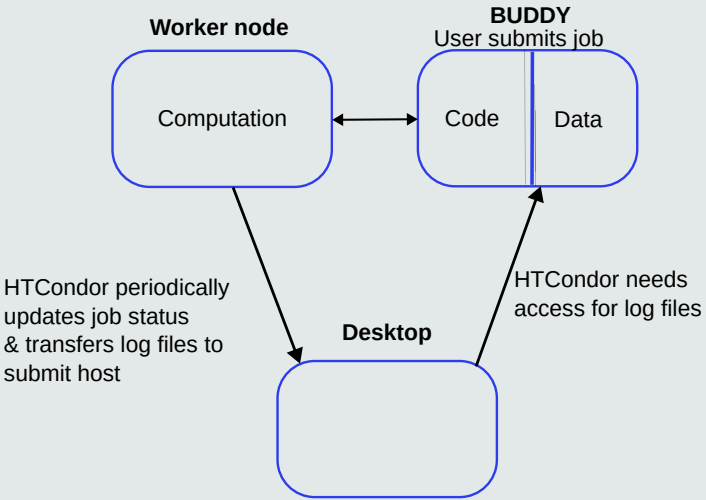
How to debug?

Analysis code - Some general pointers

- Inspect log files
- Check resource usage of jobs
- Generally:
  - Try out the workflow in an interactive job with the same commands you put into the job script
  - Slowly scale up production jobs (start with a few)
  - Understand the scaling of your code, more cores is not always better

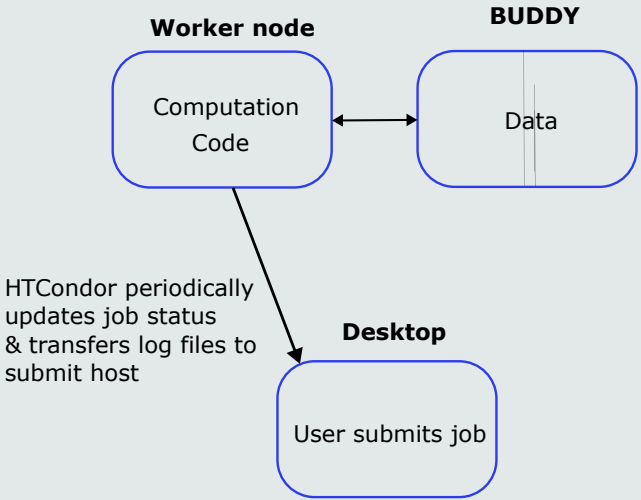
How to get your Code & Data into your jobs?

Why it matters



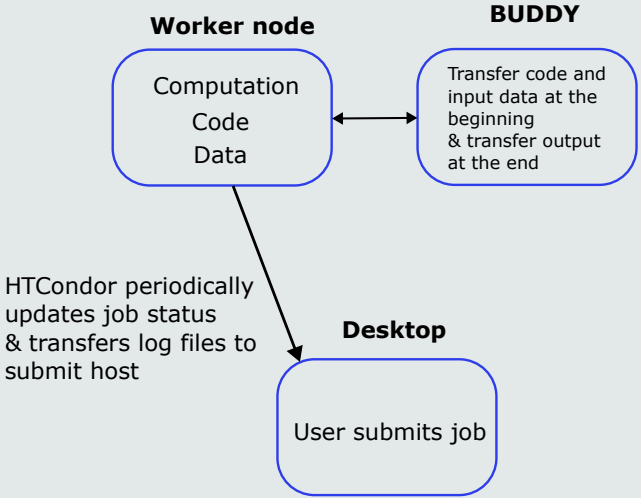
How to get your Code & Data into your jobs?

How to mitigate these issues → Link to example workflow



How to get your Code & Data into your jobs?

How to mitigate these issues → Link to example workflow



File systems: A short introduction

- File systems (in general) are just an abstraction layer to store data
- Common expectations:
  - There are file names and directories
  - There is file metadata (e. g. creation time)
  - There is a permission / access control model
  - The file system prevents data corruption by design (e. g. allows locking of files)
  - File systems are backed up
- There are file systems fulfilling none of these, and for classic data analysis, most of that is not really needed.
  $\Rightarrow$ For example, Amazon S3 is a 'flat file system' which only has names and some authentication, but nothing more.
- However, most applications (and users) expect such POSIX-like requirements (**P**ortable **O**perating **S**ystem **I**nterface).

File systems: A short introduction

File systems you may encounter...

- Home directories (NFS / CephFS)
- A local file system holding the operating system
- BAF: CephFS directories (BUDDY: **B**AF **U**ser **D**ata **D**rector**Y**)
- S3: Backup system
- CernVM-FS (CVMFS) for software and container distribution
- ownCloud / Sciebo (WebDAV)
- Potentially, some virtual file systems (e. g. SSHFS, XRootD-FS,...)
- At CERN and DESY: NFS, dCache, AFS, EOS,...
- Maybe an even different file system on your laptop, a USB pendrive, an external hard drive...

Ceph: History

- 2004: First line of code written (summer internship of a computer science student: Sage Weil)
- 2005: fully functional prototype, named 'Ceph'
  name is an abbreviation of 'cephalopod'
- 2006: First public presentations
- 2007: PhD thesis of Sage Weil, now working on Ceph full-time and founding a team
- 2010: Ceph client within the Linux kernel
- 2012: First major stable release, company 'Inktank Storage' founded for commercial support / services
- 2014: Company bought by RedHat
- 2015: Ceph Community Advisory Board founded, includes e. g. CERN and many commercial entities
- 2018: Linux Foundation launches the Ceph Foundation

Ceph: History in Bonn

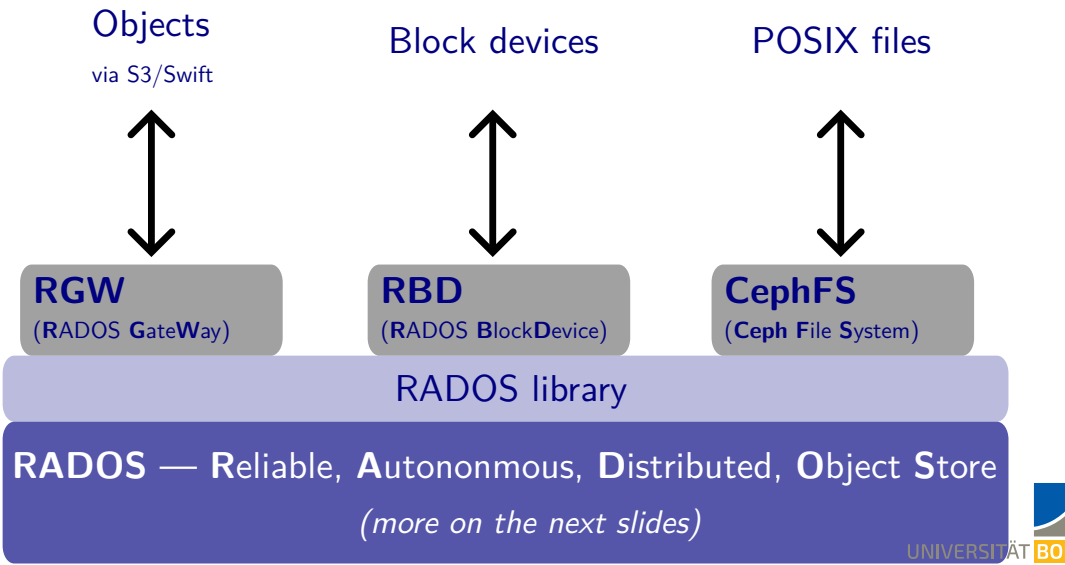
- 2013: First experiments for backup purposes
- 2017-2018: Production setup as storage for the BAF computing cluster
- mid 2018: Production setup as backend for virtual machines
- Autumn 2019: Production setup for Backup system, re-using $>$ 10 years old file servers

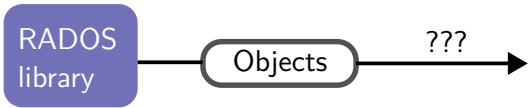
Different usage models

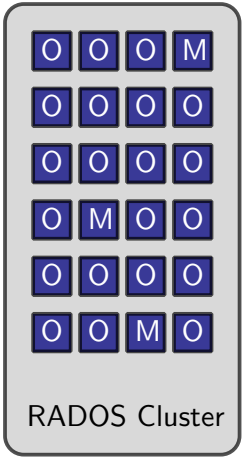
- BAF cluster: 'Classical' file system (POSIX) with directories, files, locking etc.
- VMs: 'Block devices', i. e. virtual 'disks' accessed in blocks
  with extra features such as 'snapshotting', live mirroring and more
- Backup system: Object storage / upload and download of objects via Web protocols

How does Ceph accommodate these use cases, and why choose Ceph for all of them?

Ceph: A layered system







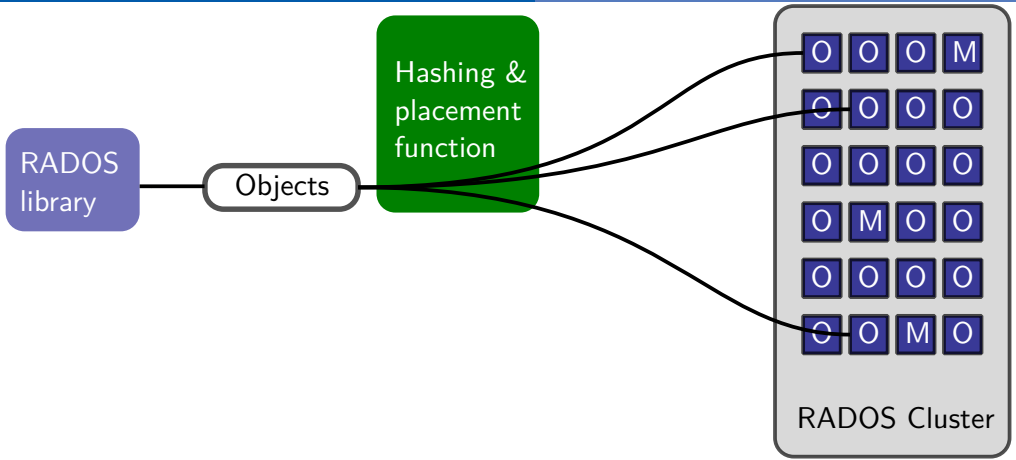
RADOS: **R**eliable **A**utonomic **D**istributed **O**bject **S**tore

O: OSD (**O**bject **S**torage **D**aemon), holds data

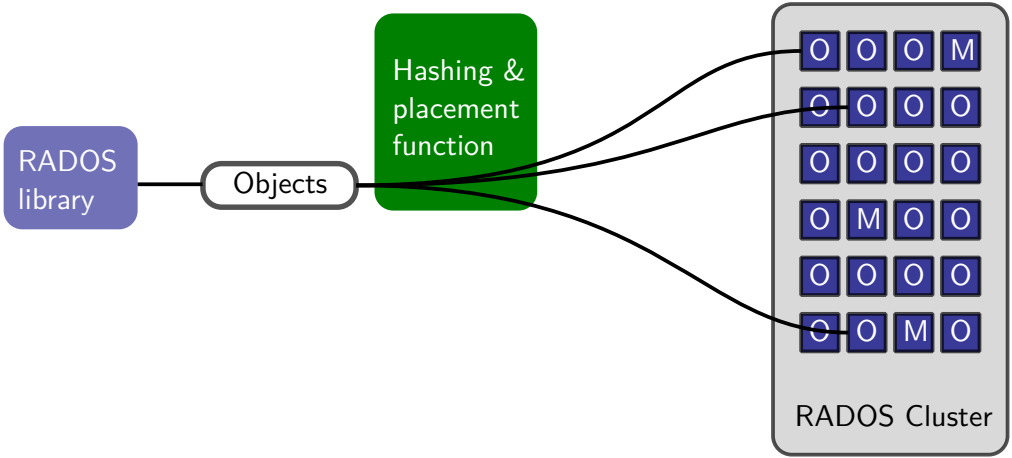
M: MON (**MON**itor daemon), keeps track of status

How to distribute those objects in a stable manner?

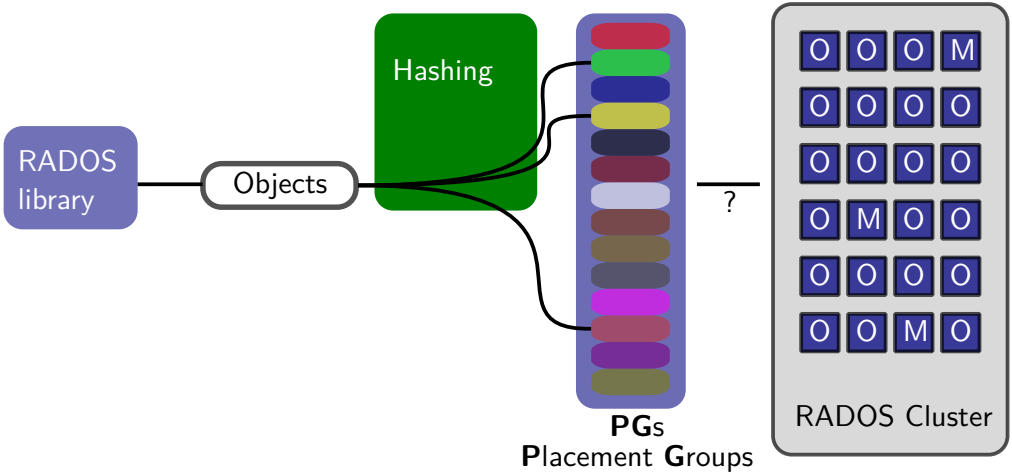
- Could keep a map / table (database)
  ⇒ Could get out of date, would ne needed for each client, slow!



- A function to control placement?
  Hashing means ‘mapping data of arbitrary size to fixed-size values’

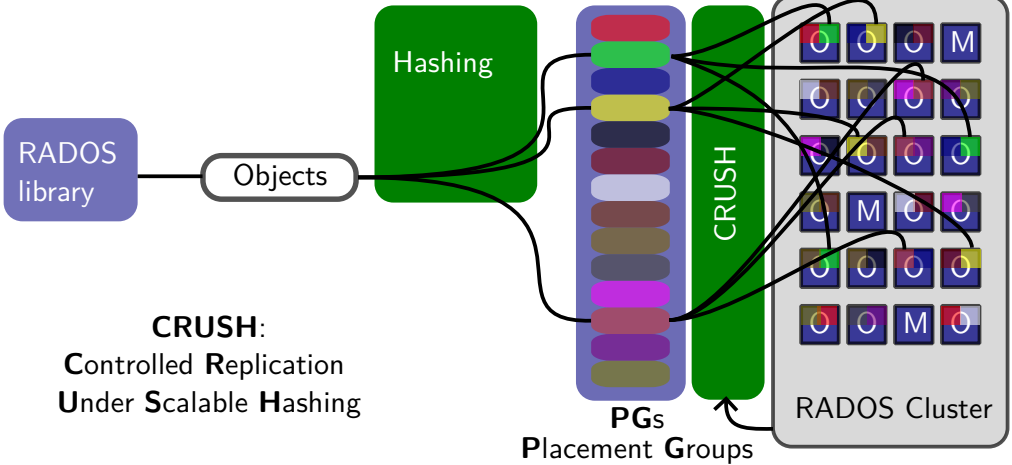


Problems: What if a disk is added / removed / fails? How to distribute copies ('rack-awareness')? ⇒ The Computer Scientist approach:
Solve a problem by adding another layer of indirection!



Two-step placement:

- Map to placement groups by hash.
- Then place these ⇒ Placement rules?



Second-stage placement:

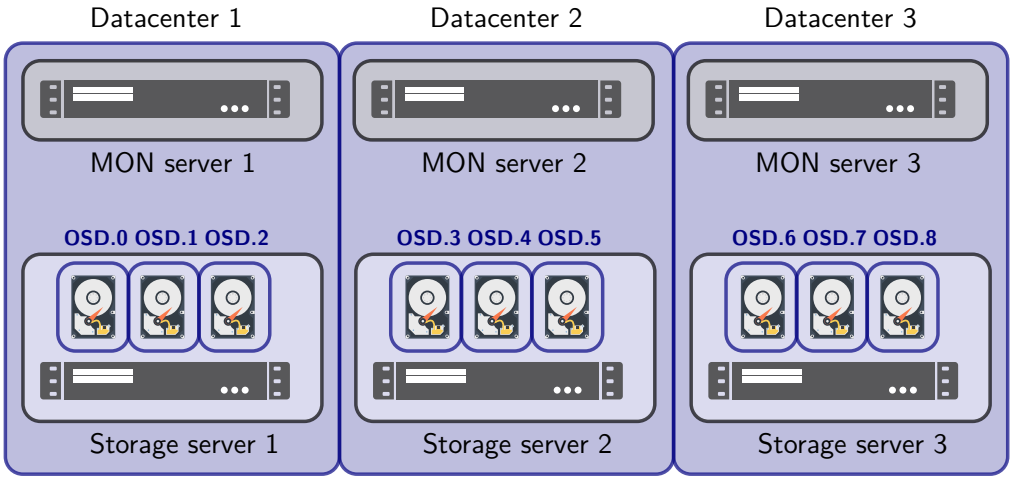
- Known choosing algorithm ('CRUSH' map) used by clients.
- Takes location constraints, cluster state etc. into account.

Ceph Hardware

Paradigms

- Storage is managed by OSDs (**O**bject **S**torage **D**aemons)
- One OSD typically handles exactly one physical block device (HDD, SSD, non-volatile memory,...)
- Storage device used directly by Ceph nowadays (no file system in between)
- Behind the scenes: Raw blocks on disk store objects, index and versions kept in a database
- Can use multiple devices (fast & slow device) for better performance
- Additional 'MON' daemons which keep track of overall status and history (can be run on the same machines)
- For CephFS, additional 'MDS' daemon(s) needed (for file system metadata, locking etc., actual data and metadata kept on Ceph)

Ceph Hardware



Distribution of hardware across rooms

HRZ machine room	HISKP	PI ⇒ future: ROT	FTD

BAF Cluster: Compute Nodes

(location HISKP coming soon)

BAF Cluster: Storage

Virtualization infrastructure

about 125 VMs, hypervisors and storage redundant, Ceph Block devices, 3 copies

Backup system

redundant, Ceph RGWs / S3, 3 copies

Ceph Hardware: If things go wrong...

Silent failures?

- 'Scrubbing' (full read back) just about once every week
- Checksums for all objects kept and checked before returning any data

Hardware can fail...

- Location constraints used (e. g. 'keep one copy in each datacenter')
- OSDs fail? Placement groups are 'remapped' (following same location constraints)
- By default: Writing only allowed if the system would tolerate another failure
- Recovery happens automatically within the RADOS cluster, clients only need to adhere to the map they download from the MONs
- Quorum logic for MON servers (if $< 50\,\%$ are in quorum, cluster blocks client I/O)

$\Rightarrow$ Self-healing, failures do not cause a downtime nor reach the user
(unless too many simultaneous failures)

Ceph: A Software Defined Storage System

Replication

- Data can be replicated (multiple copies)
- Data can also be erasure-coded ("shards", e. g. split each object into 4 shards and calculate 2 extra shards via Reed Solomon Coding)
- Clients don't need to care: They always talk to the 'primary OSD' for a placement group, OSD replicates behind the scenes

Versioning

- Copy-on-Write, i. e. a copy is made if modifications need to be done, no 'in-place' changes.
- Keeping track of versions and references to them when 'snapshots' are made.

Ceph: Usage

S3 / RGW: Using Ceph as object store, storing and retrieving via HTTP(S))

- Done e. g. by Duplicati, Restic etc. for backups
- Tools 'pack' chunks of data and keep an index, storing several versions of files

RBD: Block devices for virtual machines as 'bunch of objects' with a map

- Can be snapshotted (many versions kept in our virtualization infrastructure)
- Differential backup to Backup cluster, incremental backup to 'classic' storage

CephFS: FS with POSIX semantics (Portable Operating System Interface)

- Locking of files (exclusive read/write access)
- In-place modification of files
- File metadata (directories, create / access / mod. times, permissions, . . .)

⇒ Solved by adding an MDS (**M**eta**D**ata **S**ervice), stores data on Ceph itself, can be operated with high availability and scaled out

CephFS on the Desktop Machines?

Authentication and Security?

- CephFS itself has system-based authentication, i. e. the client has one key, can access the file system, then user permissions apply
- This approach can not be used for machines which can be 'carried away' or are operates by different administrators!

File system Authentication

- Kerberos allows users to fetch a 'Ticket Granting Ticket' (TGT), then authenticate to various (trusted) services (can be done with Uni ID!)
- Can use that to authenticate via SSH, to web pages, to file systems,...

$\Rightarrow$ Short-lived Kerberos TGT of the user required to access the file system!

$\Rightarrow$ How to do this with CephFS?

CephFS on the Desktop Machines

Network file systems with Kerberos support?

- AFS (fading out since years, used in Bonn with BAF1 with some. . . issues)
- NFS v4 and newer (very widespread especially in the Linux world)
- SMB (widespread in the Microsoft world — Kerberos used there, too!)

Solutions for NFS

- Kernel NFS server: Long history and quite stable, but does not work with all file systems
- NFS Ganesha server: Runs in userspace, active development, direct integration with CephFS

⇒ Used both for `/cephfs` and home directories on desktops

File systems: How are clients synchronized?

Locking

- Exclusive access to files requires a kind of 'lease', i. e. a 'lock' is communicated to a central metadata service
- These are called CAPs (from 'capabilites') in Ceph and can be quite granular
- Another client needs exclusive access, or locks are held very long: Should be returned / are recalled

Client-side caching

- Usage of `cachefilesd`, intercepts all file access and keeps local cache
- Cache based on file sizes and access frequency
- Cache eviction based on LRU (**L**east-**R**ecently-**U**sed) principle
- Server can notify client that cached file is out of date
- Client can check metadata of cached file against server to ensure it is 'fresh'

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I work remotely on a desktop, then close the lid of my laptop and leave for the weekend?

Ideas

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I work remotely on a desktop, then close the lid of my laptop and leave for the weekend?

Ideas

- Everything keeps running
- Kerberos TGT expires and file access is lost...
  - ...and everything crashes?
  - ...and everything freezes?

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I work remotely on a desktop, then close the lid of my laptop and leave for the weekend?

Answer?

$\Rightarrow$ **It depends...**

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I work remotely on a desktop, then close the lid of my laptop and leave for the weekend?

Real answer

- Kerberos TGT will be prolonged automatically for up to 7 days after the last manual login with password
- If it expires (you are gone for a longer while, forget about the session...)
  - Applications don't crash (POSIX), they get 'permission denied' when trying to access any file
  - Applications with structured data (e. g. databases) may frantically retry to save...
    ⇒ This includes e. g. Firefox, Thunderbird, Code Editors...
  - Everything continues as normal if you log in again...
  - If you don't, things keep hammering the file servers.

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I edit the same file on two machines in parallel?

Ideas

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I edit the same file on two machines in parallel?

Ideas

- Last save wins?
- Editor blocks until the other one is closed
- File gets corrupted

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I edit the same file on two machines in parallel?

Answer?

$\Rightarrow$ **It depends...**

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if I edit the same file on two machines in parallel?

Real answer

- Most editors do atomic updates, e. g. write to a separate file, then swap out the original file
- But: Can get stuck trying to lock the file when opening it exclusively
- 'Fights' about getting the lock may start, hammering the servers

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if my desktop hangs and I turn it off hard (power button)?

Ideas

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if my desktop hangs and I turn it off hard (power button)?

Ideas

- It will boot up fine again and work faster
- The file system on the hard drive may get corrupted

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if my desktop hangs and I turn it off hard (power button)?

Answer?

$\Rightarrow$ **It depends...**

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if my desktop hangs and I turn it off hard (power button)?

Real answer

- The file system on the hard driver may indeed get corrupted
- File system checks (hard disk) will slow down boot and after machine has booted, CVMFS cache is checked (slowdown for several minutes possible)
- Machine may be blocked from getting locks back for 5 min, as the server does not know what happened to the original shared file system client
- If there was a hang and restart of the NFS server, mounting and login may fail, another reboot or long waiting time may be needed

CephFS on the Desktop Machines — when things go wrong...

Pop Quiz

What happens if my desktop hangs and I turn it off hard (power button)?

Keep in mind the following default intervals

- CephFS client eviction: 5 min
- NFS Ganesha automatic restart: 5 min
- NFS grace period: 90 s (may be lifted earlier with modern NFS if all clients report back)

CephFS on the Desktop Machines — when things go wrong...

Known issues

- Copies in graphical file manager may cause an endless loop (seems to hang, hammers file servers)
  ⇒ Issue known, fixed in Debian 12, only reported by 3 users while seen by many...
- Rarely, cache incoherency observed, i. e. files contain fixed pattern after real content
  ⇒ Mostly if editing from various machines in parallel instead of one machine

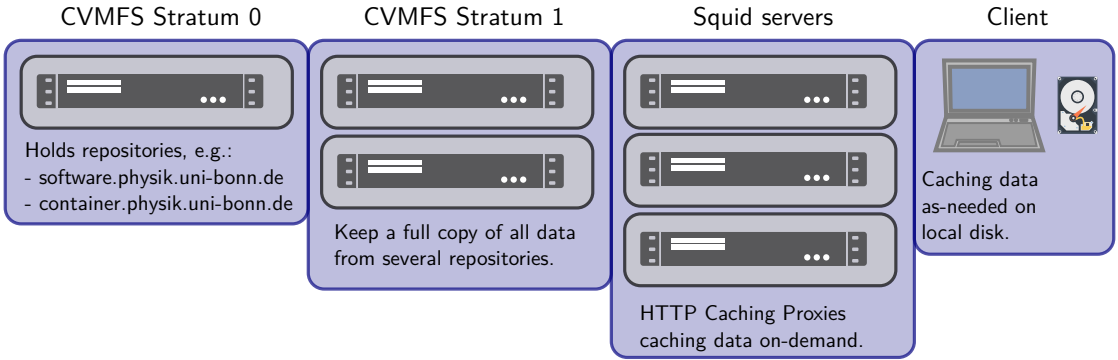
Recommendations

- Work from one machine (e. g. your 'own' desktop), see our documentation on SSH multiplexing, WireGuard etc.
- React on automated logout mails
- Use commandline copying until we've rolled out Debian 12
- Don't pull plugs on desktops / don't turn them off hard, see our desktop reboot documentation if really needed

CernVM-FS: A File system to distribute containers and software

- Optimized for large number of small files (software, containers)
- Read-only, HTTP-based file system
- All data is chunked and hashed, catalogs built for directory subtrees
- Files saved in a content-addressable manner
- Deduplication and versioning by design, compression possible
- Read-only model enables caching in many layers (only catalogs require a short lifetime, usually 5 min)
  (if faster updates are needed, CVMFS supports a notification model)
- Clients only need to cache the data they actually use

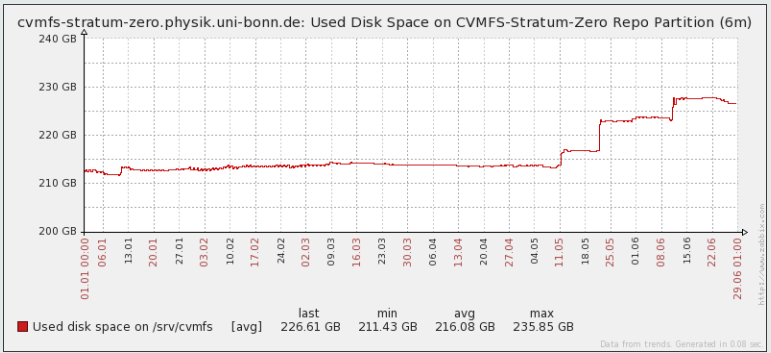
CernVM-FS: A File system to distribute containers and software



CernVM-FS: A File system to distribute containers and software

- Deduplication saves significant space (daily containers, new software versions from time to time)

CVMFS usage over half a year, Containers (daily) & Software



Summary

- New automation concerning institute registration / deregistration rolling out
  Note this is partially to prepare for an upcoming change of the University mail system
- Construction projects are absorbing a significant fraction of time and energy
- Recommendations on how to use the cluster efficiently presented
- File systems:
  - A plethora of file systems are used daily, most 'behind the scenes'
  - Different file systems are ideal for different use cases
  - Some insights into Ceph, how and why it is used were shown
  - Details on how to use file systems efficiently / not breaking them

Recommendation

Courses by the central HPC team: `https://www.hpc.uni-bonn.de`
on Linux, Python, building your own cluster,...

Thank you

for your attention!