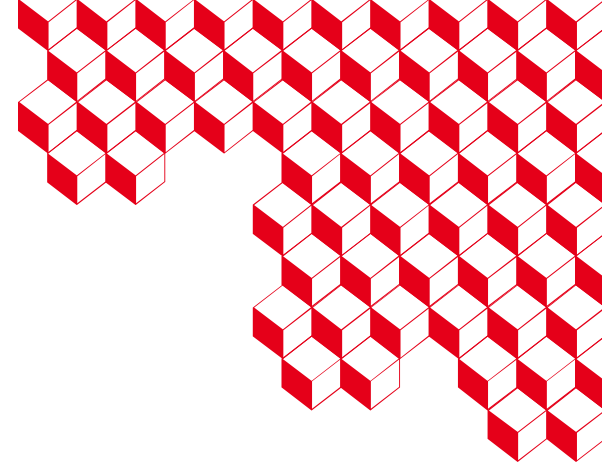




RobinHood v4 Progress Report

LAD'25 – 30th of September, 2025

Gauthier EVRAERD, gauthier.evraerd@cea.fr

Yoann VALERI, yoann.valeri@cea.fr

Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

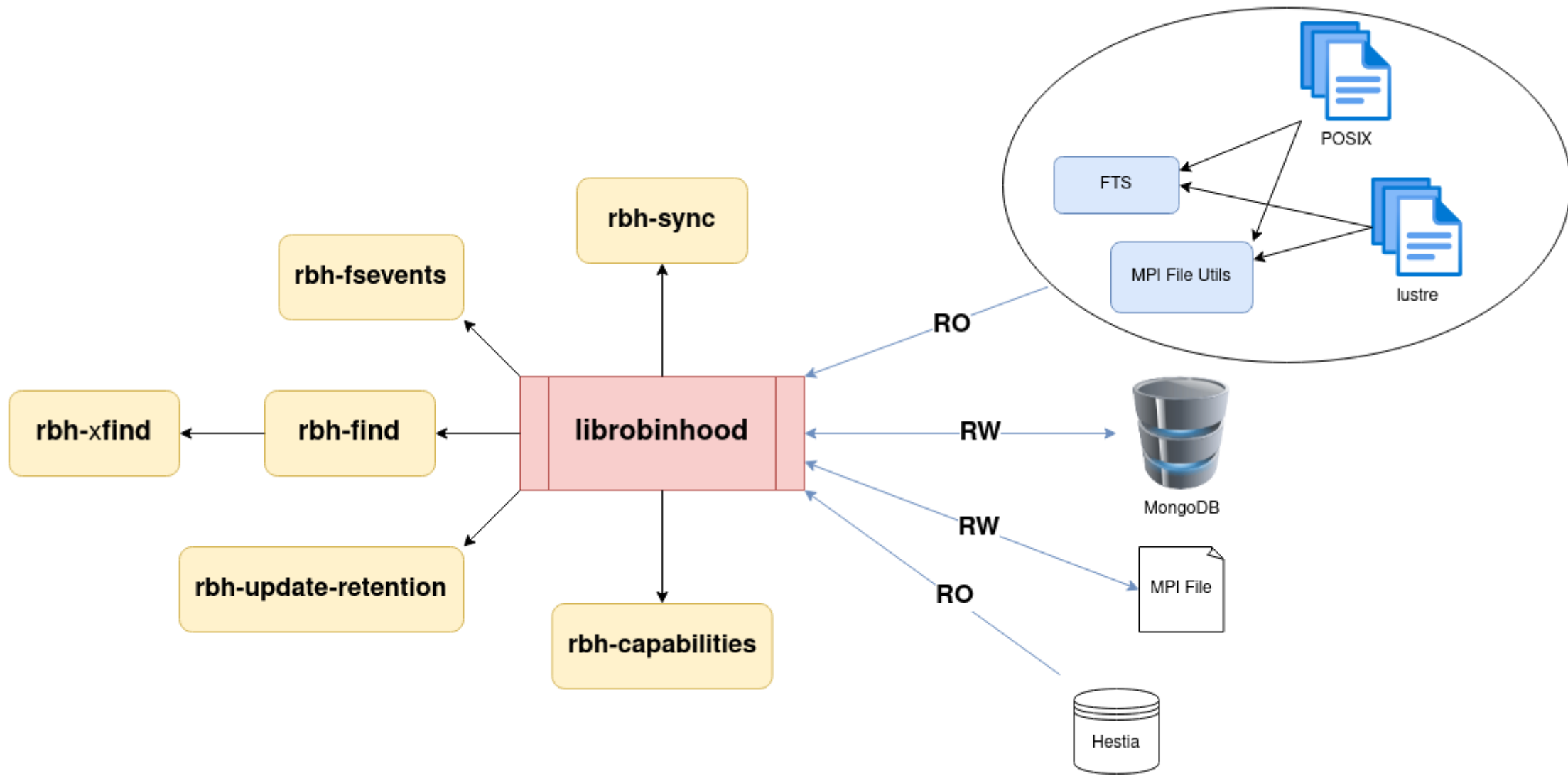
What is RobinHood?



Provide efficient and easy to use tools to replicate and query any storage system's metadata

One of the main open source Policy Engine for Lustre filesystems over the world

Focus on RobinHood v4 (as of last year)





1 ■ Tools and backends

What's new? (1)

■ librobinhood:

- New S3 plugin
- Alias → shorten frequently used command line components

```
alias:  
small_log: "-type f -size -1G -iname '*.log' -or -iname '_log'"
```

■ rbh-find:

- New filters: UID/user/nouser, GID/group/nogroup, empty, mdt-index, mdt-count, ...
- Generic filters and actions
- Removed overloads of rbh-find

```
[root@vm0 lustre]# touch a  
[root@vm0 lustre]# touch b  
[root@vm0 lustre]# lfs hsm_archive b  
[root@vm0 lustre]# rbh-sync rbh:lustre:. rbh:mongo:lad  
[root@vm0 lustre]# rbh-find rbh:mongo:lad -hsm-state archived  
/b
```

What's new? – rbh-report

■ rbh-report:

- New tool
- Handles grouping on specific fields, output accumulators and formats

■ Filtering

```
[root@vm0 tmp]# mkdir -p dir1/dir2
[root@vm0 tmp]# truncate -s 30 file1
[root@vm0 tmp]# truncate -s 70 dir1/file2
[root@vm0 tmp]# rbh-sync rbh:posix:. rbh:mongo:lad
[root@vm0 tmp]# rbh-find rbh:mongo:lad -printf "path: %p, size: %s\n"
path: /, size: 31
path: /file1, size: 30
path: /dir1, size: 31
path: /dir1/file2, size: 70
path: /dir1/dir2, size: 6
[root@vm0 tmp]# rbh-report rbh:mongo:lad --group-by "statx.type" --output "sum(statx.size),count()"
statx.type || sum(statx.size) | count()
-----
directory ||          68 |      3
file      ||         100 |      2
[root@vm0 tmp]# rbh-report rbh:mongo:lad --group-by "statx.type" --output "sum(statx.size),count()" -not -path "/"
statx.type || sum(statx.size) | count()
-----
directory ||          37 |      2
file      ||         100 |      2
```

What's new? (2)

- **rbh-capabilities** → **rbh-info**:
 - Show the number of entries, size of the storage system, source system

```
[root@vm0 tmp]# rbh-info mongo
Capabilities of mongo:
- filter: rbh-find [source]
- synchronisation: rbh-sync [source]
- update: rbh-sync [target]
- branch: rbh-sync [source for partial processing]
[root@vm0 tmp]# rbh-info rbh:mongo:lad --count
5
[root@vm0 tmp]# rbh-info rbh:mongo:lad --size
2.27 KB
[root@vm0 tmp]# rbh-info rbh:mongo:lad --backend-source
posix
```

What's new? (3)

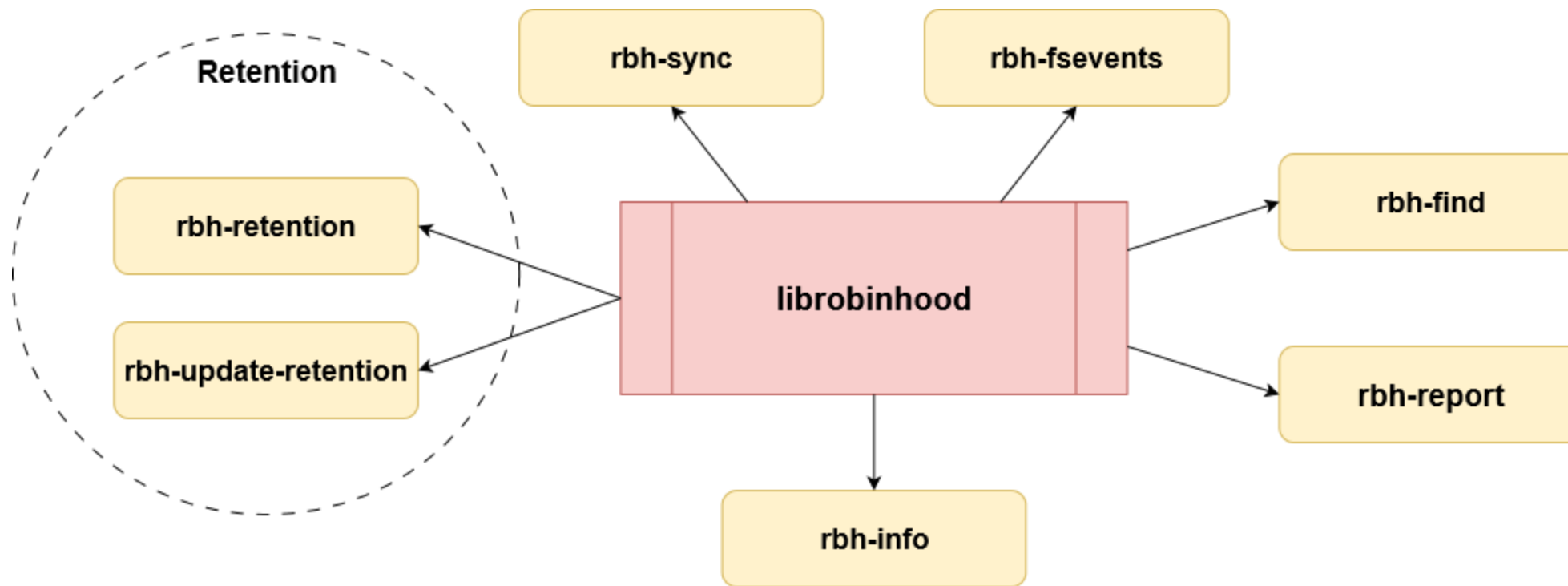
- **rbh-retention:**
 - New tool for regular users
 - Set retention attribute on their entries
 - `date(1)`-like formats

```
[root@vm0 tmp]# rbh-retention file1
'/home/valeriy/tmp/file1' has no retention date set
[root@vm0 tmp]# rbh-retention file1 "3 days" --relative
[root@vm0 tmp]# rbh-retention file1
'/home/valeriy/tmp/file1' is set to expire 3 day(s) after its last modification time,
so '/home/valeriy/tmp/file1' expires on 'Mon Sep 29 07:20:54 UTC 2025'
```

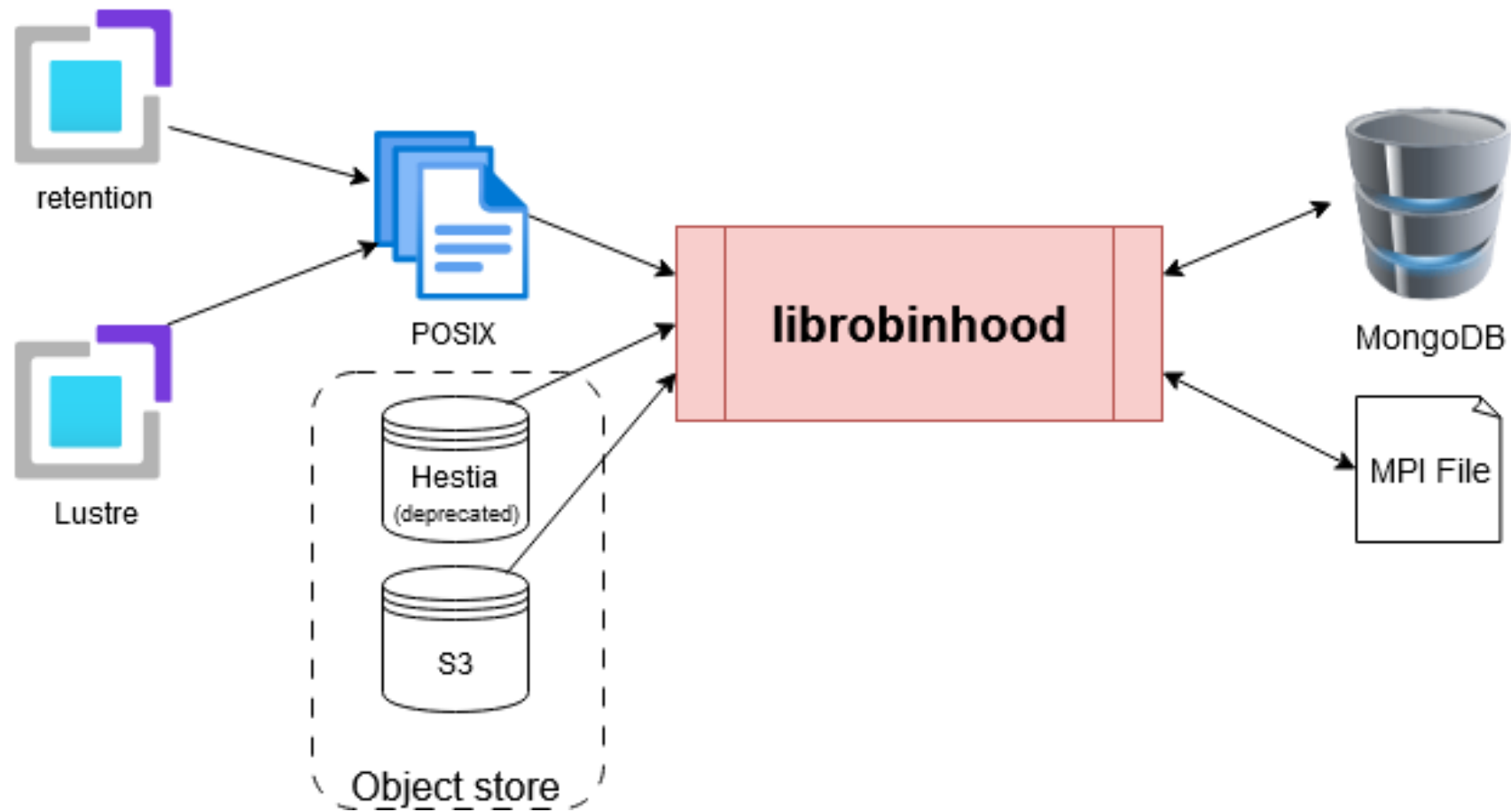
Ongoing work

- Parallelization of enrichment and update in **rbh-fsevents**
- **rbh-policy** → Policy engine of RobinHood v4
 - Configuration files written in Python
 - Syntax and concepts (rules, classes, policies, actions) close to RobinHood v3
- **rbh-undelete** → Recover a removed file still archived in HSM

Current RobinHood v4 - Tools



Current RobinHood v4 – Plugins/extensions



Project status

- RobinHood v4 version 1.3
- Tested with
 - RHEL 8.8 and 9.4
 - Lustre 2.12 and 2.15
 - MongoDB 7 and 8
- Containerfile available for testing
- Deployed on the TERA supercomputer for a small subset of users
 - Tailored for the retention feature
- Deployment on our R&D supercomputer before the end of year



2 ■ Benchmarks

Methodology



Materials	RobinHood	4 nodes 2 CPU Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz → 12 logical cores 64 GB of RAM
	MongoDB	1 node 2 CPU AMD EPYC 7773X 64-Core Processor 2.2 GHz → 128 logical cores 264 GB of RAM 1 SSD NVME 64 TB
	Network	ROCE between MDS and Robinhood Ethernet between Robinhood and MongoDB
Method	Clean the cache on all nodes (Robinhood/MongoDB) before each execution	
Datasets	SCRATCH of a Lustre filesystem for rbh-find and rbh-report (185M inodes → 52 GB MongoDB) Changelogs of a Lustre filesystem for rbh-fsevents	

Metrics



rbh-fsevents	rbh-find	rbh-report
Number of changelogs processed per second (changelogs / s)	Time of execution in seconds	Time of execution in seconds

Improvements done

RobinHood

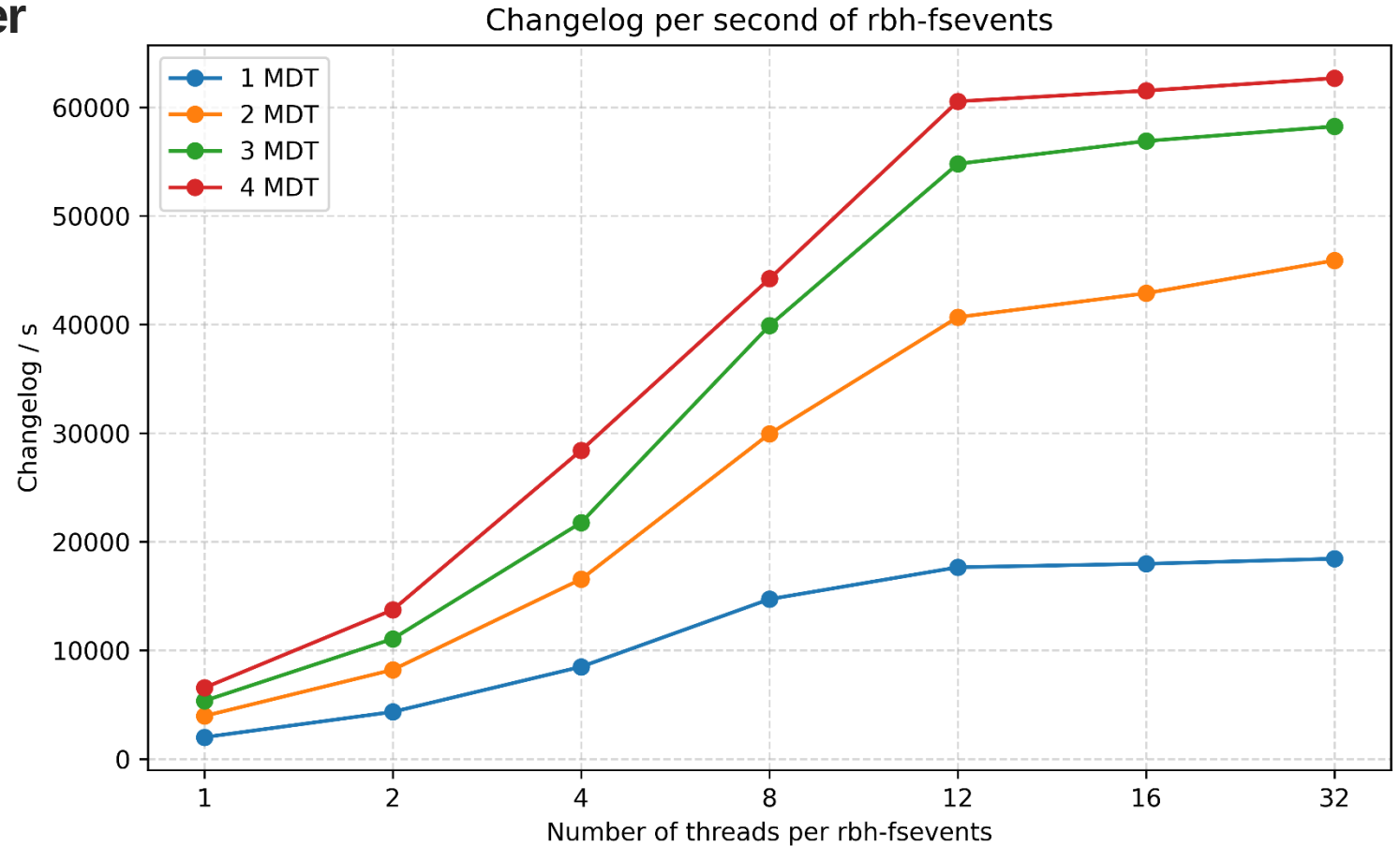
- rbh-find
 - use the projection to reduce the size of data to retrieve → faster read on the database
- rbh-fsevents
 - Parallelization of enrichment and update (producer/consumer)
 - Prevent multiple opens and stats → faster read on the filesystem

MongoDB

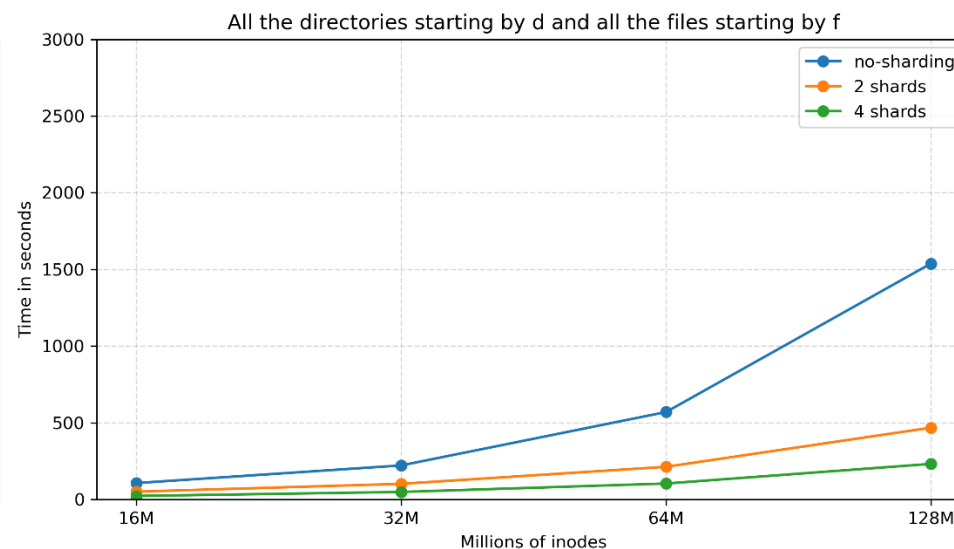
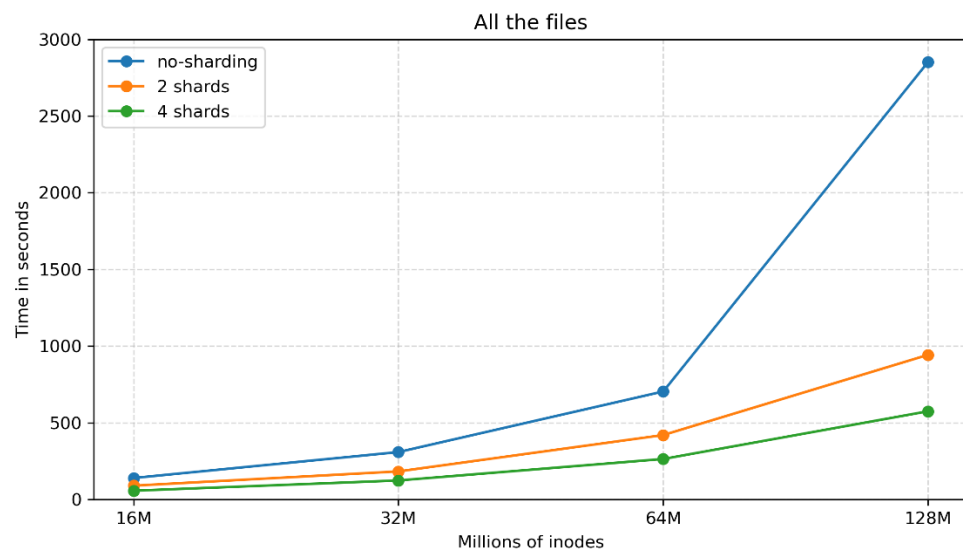
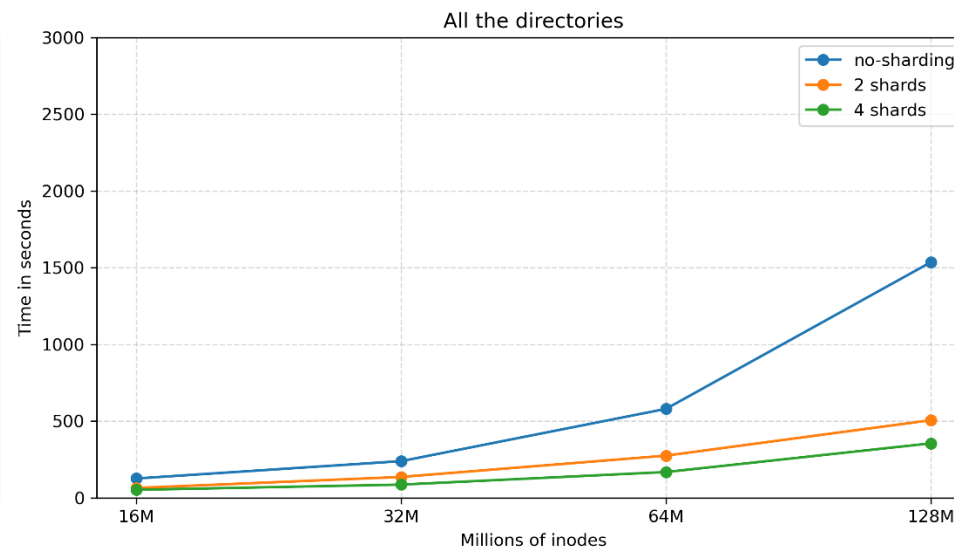
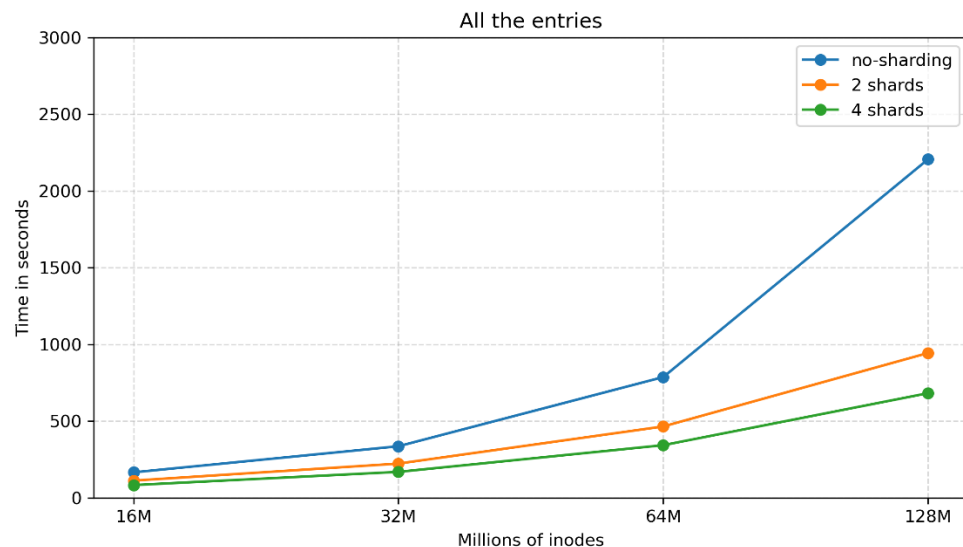
- Sharding on same node and distant nodes → faster read/write on the database

Higher is better !

- We launch one rbh-fsevents per MDT and node
- Observations
 - Good scalability in number of MDT
 - Very good scalability when increasing the threads
- Limitations
 - MDS nodes and network



All shards are on the same node
Lesser is better !



rbh-find



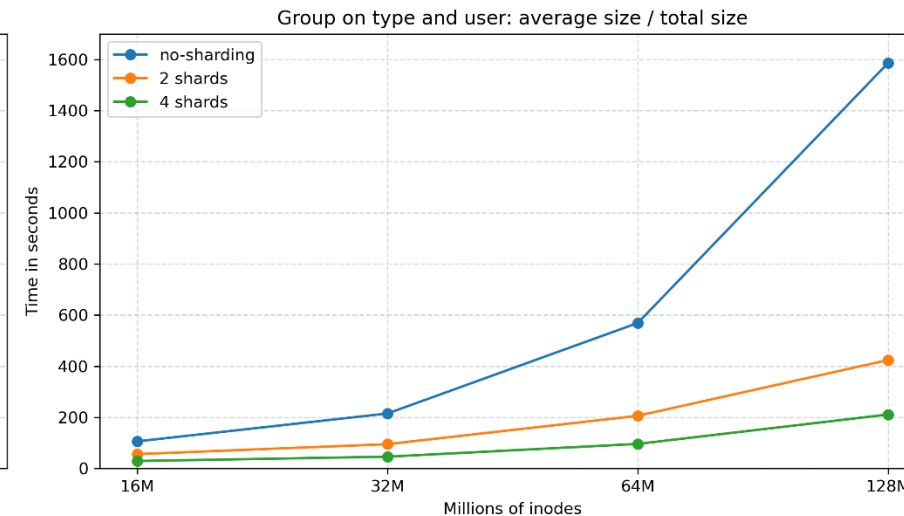
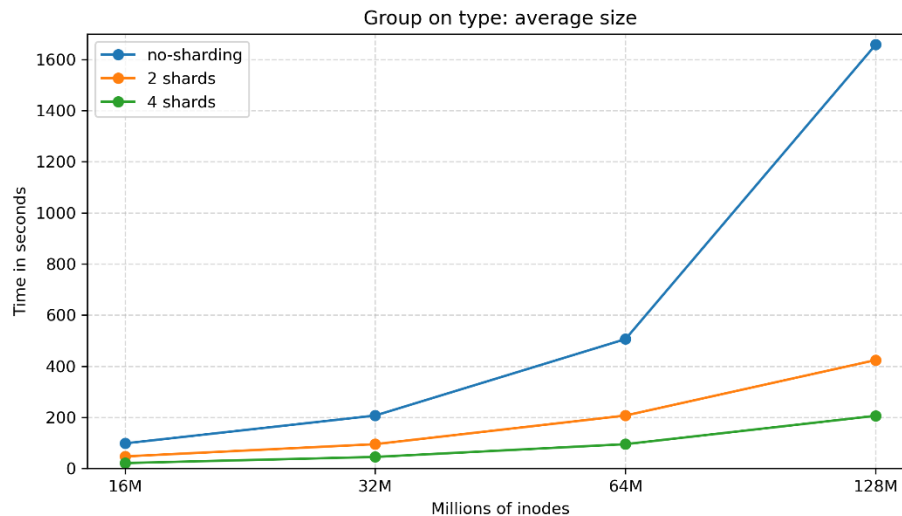
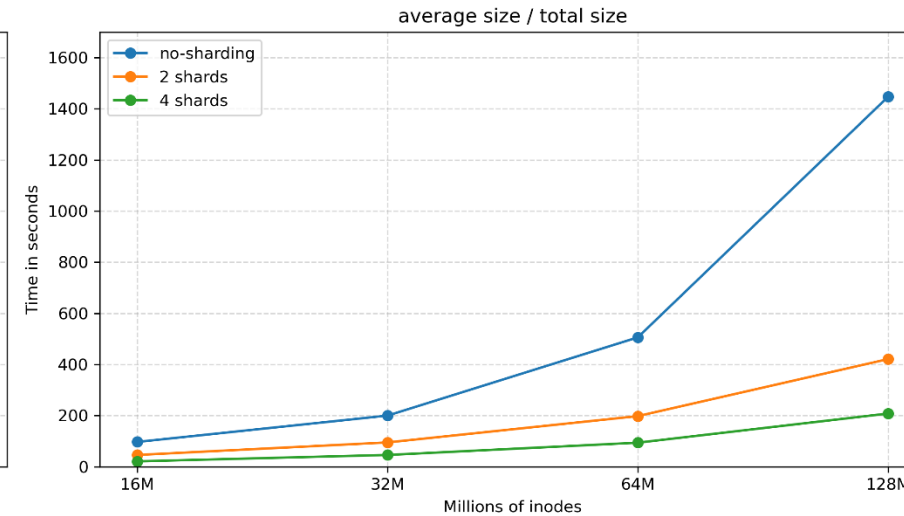
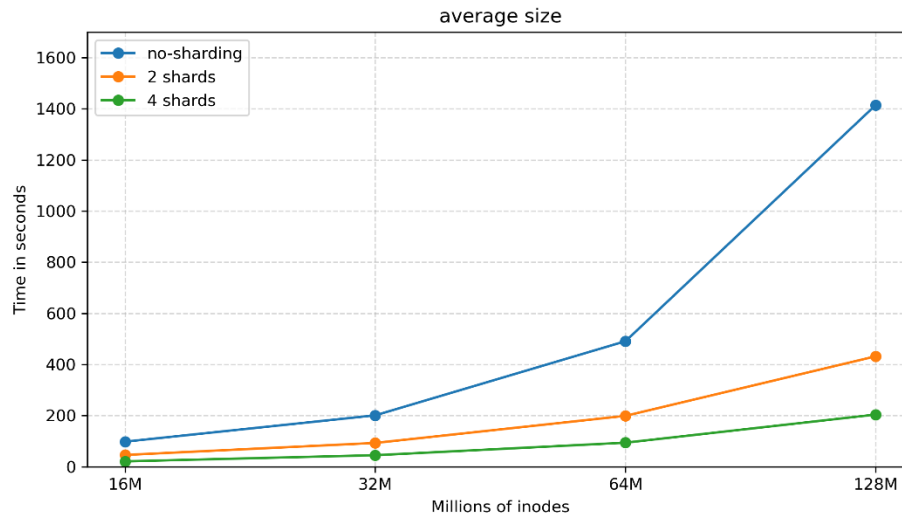
■ Observations

- Increasing the number of shards speeds up rbh-find
- Also increases scalability with the number of inodes in the database
- The number of entries to retrieve affects execution time as much as the complexity of the query

■ Limitations

- Performance of the database nodes (disk/SSD, RAM)
- Network between rbh-find and MongoDB

Same observations and limitations as for rbh-find



RBH3 vs RBH4



Number of inodes: 185 millions

With the same configuration
as the database in production

With 4 shards on the same node

rbh-report	RBH3	RBH4
-g cea -split-users-groups	Immediate	~ 2 minutes 30 seconds
-g cea -split-users-groups -p /dir1	~ 7 hours	~ 3 minutes

rbh-find	RBH3	RBH4
/dir1	~ 6 hours	~ 14 minute
-group cea -size +1G -type f	~ 5 minutes 30 seconds	~ 1 minute 20 seconds



3. Conclusion

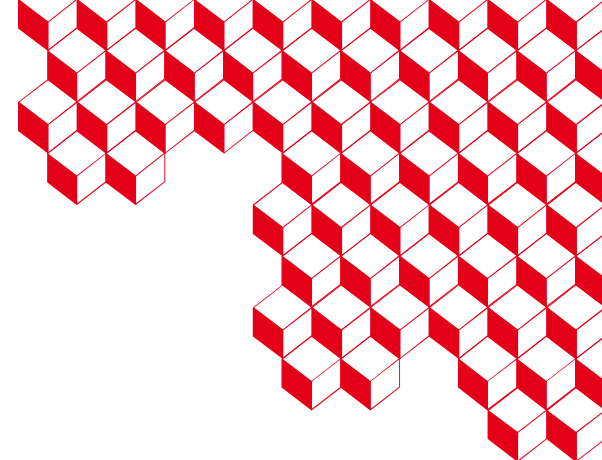
What's next?

- **rbh-fsevents** as a daemon
- Project ID support
- **rbh-gc**
- Metrology
- Phobos¹ backend for cold storage management
- SELinux backend

¹ <https://github.com/phobos-storage/phobos>

Want to help with the development ?

- The tool suite is on Github: <https://github.com/robinhood-suite/robinhood4>
- Reviews done on Gerrithub → Soon to be a Gitlab forge
- Some patches in review:
 - <https://review.gerrithub.io/c/robinhood-suite/robinhood4/+1219423> - **rbh-undelete**
 - <https://review.gerrithub.io/c/robinhood-suite/robinhood4/+1221157> - **rbh-fsevents** parallelization
 - <https://review.gerrithub.io/c/robinhood-suite/robinhood4/+1217782> - **rbh-policy**
- Feel free to install the suite and test it out for yourself, any feedback in appreciated!



Thanks for your attention

RobinHood v4 progress report

Commissariat à l'énergie atomique et aux énergies alternatives – www.cea.fr

Gauthier Evraerd, Yoann Valeri
Software stack developers at CEA
gauthier.evraerd@cea.fr
yoann.valeri@cea.fr
thomas.leibovici@cea.fr

Why a version 4 for RobinHood?



	Version 3	Version 4
Scale up	SQL paradigm <i>MariaDB</i>	NoSQL paradigm <i>MongoDB</i>
Code genericity	Software specialised for Lustre filesystems	Generic tools calling specific backends
Inclusion to Linux repositories	Expert system	Library of features and applications
Code refactoring	Heavy code caused by Lustre behaviour evolution	Clean design to better correspond to current filesystems